Sparse Multivariate Modeling: Priors and Applications

Henao, Ricardo; Winther, Ole

Publication date:
2011

Document Version
Publisher's PDF, also known as Version of record

Sparse Multivariate Modeling: Priors and Applications

Ricardo Henao

Kongens Lyngby 2011
IMM-PHD-2011-253

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-PHD: ISSN 0909-3192

Summary

This thesis presents a collection of statistical models that attempt to take advantage of every piece of prior knowledge available to provide the models with as much structure as possible. The main motivation for introducing these models is interpretability since in practice we want to be able to use them as hypothesis generating tools. All of our models start from a family of structures, for instance factor models, directed acyclic graphs, classifiers, etc. Then we let them be selectively sparse as a way to provide them with structural flexibility and interpretability. Finally, we complement them with different prior assumptions in order to make them appropriate at handling different domain specific situations as time series, non-linearities, batch effects, missing values, etc. In particular, we present a framework for linear Bayesian networks we call sparse identifiable multivariate modeling, a model for peptide-protein/protein-protein interactions called latent protein tree, a framework for sparse Gaussian process classification based on active set selection and a linear multi-category sparse classifier specially targeted to gene expression data. The thesis is organized to provide a general yet self-contained description of every model in terms of generative assumptions, interpretability goals, probabilistic formulation and target applications. Case studies, benchmark results and practical details are also provided as appendices published elsewhere, containing reprints of peer reviewed material.

Resumé

Denne afhandling handler om et sæt af statistiske modeller hvor det er forsøgt at inkludere mest mulig a priori viden med det formål at give modellerne mest mulig struktur. Hovedmotivationen for denne tilgang er at vi ønsker at kunne fortolke modellerne og bruge dem som et hypotesegenerende værktøj. Alle modellerne tager udgangspunkt i en modelfamilie, såsom faktor-modeller, orienterede acykliske grafer, klassifikationsmodeller, etc. Derefter bruger vi en selektiv mekanisme til at sætte mange parametre til nul så modellen er tyndt forbundet (på engelsk *sparse*). Dette komplimenteres slutteligt med a priori information for det specifikke domæne såsom tidsserier, ikke-lineariteter, batch effekter, manglende observationer, etc. Specifikt er fokus i afhandling på følgende: en metode for lineære Bayesianske netværk som vi kalder *sparse identifiable multivariate modeling*, en model for protein-peptid/protein-protein interaktion kaldet *latent protein tree*, en aktiv sæt metode for stor-skala Gaussisk process klassifikation og en lineær tyndt forbundet multi-klasse klassifikations metode specielt velegnet til gen-expressionsdata. Afhandlingen er organiseret således at beskrivelsen af hver model er general og i princippet ikke kræver adgang til baggrundsmateriale. Hver model er forklaret i termer af grundlæggende generative antagelser, fortolkning, probabilistisk formulering og anvendelsesområder. Afhandlingens forskningsartikler, indeholdende casestudier, benchmark resultater og praktiske detaljer, er inkluderet som appendikser.

Preface

This thesis was prepared at DTU informatics, Technical University of Denmark (DTU) and the Bioinformatics Centre (BINF), University of Copenhagen (KU) in partial fulfillment of the requirements for acquiring the degree of Doctor of Philosophy.

The thesis consists of a summary report of the prior distributions, model descriptions, study cases and a collection of five research papers written during the period 2008–2011 and published elsewhere.

Lyngby, February 2011

Ricardo Henao

Papers included

(A) R. Henao and O. Winther. Bayesian Sparse Factor Models and DAGs inference and comparison. In *Advances in Neural Information Processing Systems 22*, pages 746–744, 2009.

(B) R. Henao and O. Winther. PASS-GP: Predictive Active Set Selection for Gaussian processes. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP), 2010*, pages 148–153, 2010.

(C) R. Henao and O. Winther. Sparse Linear Identifiable Multivariate Modeling. To appear, *Journal of Machine Learning Research*, 2011.

(D) R. Henao and O. Winther. Predictive Active Set Selection Methods for Gaussian processes. Submitted to *Neurocomputing*.

(E) R. Henao, J. W. Thompson, M. A. Moseley, G. Ginsburg, L. Carin and J. E. Lucas. Latent Protein Trees. In preparation, *Journal of the American Statistical Association*.

Acknowledgements

First of all I want to express my gratitude to my supervisor Ole Winther. I deeply appreciate the unique opportunity you gave me to come to Denmark and be your student, without your constant support and inspiration it would not be possible for me to be where I am now.

I want to thank all the people at Cognitive Systems Section from DTU Informatics, the Promoter Group at the Bioinformatics Centre and the University Hospital, in particular Carsten Stahlhut, Christian Walder, Ulla Nørhave, Anders Krogh, Albin Sandelin, Bogumil Kaczkowski, Rehannah Borup and Maria Rossing.

At the end of 2009, I shortly visited three labs from universities in the United States. I want to thank Richard Bonneau from New York University, Mike West from Duke University and David Haussler from University of California at Santa Cruz, for let me share my research work with them and for all those useful insights I gathered from very fruitful discussions.

During my studies, I spent almost 6 months at Duke University, visiting the Duke Institute for Genome Sciences & Policy. For this wonderful opportunity, I want to thank Joseph Lucas and all the people of his research group for providing me with a very pleasing environment for doing my research work.

I gratefully acknowledge the support of my sponsors: the DTU Informatics Graduate School ITMAN, the Bioinformatics Centre through Novo Nordisk Foundation grant, the EU Network of Excellence PASCAL 2, Otto Mønsteds Foundation and the Reinholdt W. Jorck and Hustrus Foundation.

To my parents and Diana for their permanent encouragement and patience.

Acronyms

ARD Automatic Relevance Determination. 43

CSLIM Correlated Sparse Linear Identifiable Multivariate Modeling. 8, 24–28, 48, 51

DAG Directed Acyclic Graph. 7, 23–26, 28–30, 46, 51, 52
DP Dirichlet Process. 54

EP Expectation Propagation. 36–39

fPASS-GP Fixed Predictive Active Set Selection for Gaussian Processes. 35–39

GP Gaussian Process. 7, 8, 12, 19, 20, 24, 26, 28, 30, 36, 37, 39, 46, 51, 54
GPC Gaussian Process Classification. 7, 8, 20, 35, 37, 39, 43, 46, 47, 52, 53

HBP Hierarchical Beta Process. 6

IBP Indian Buffet Process. 6
ICA Independent Component Analysis. 29
IVM Informative Vector Machine. 39

KNN k-Nearest Neighbor. 46

LASSO Least Absolute Shrinkage and Selection Operator. 6
LC-MS Liquid Chromatography and Mass Spectrometry. 31, 35
LiNGAM Linear Non-Gaussian Acyclic Model. 29
LOO Leave-One-Out. 7, 37, 40
LPT Latent Protein Tree. 24, 25, 31, 33, 34, 49, 51–54

M-H Metropolis-Hastings. 19, 20, 28, 30, 43
MAP Maximum a-Posteriori. 5, 6, 13, 43
MCMC Markov Chain Monte Carlo. 13, 17, 20, 28, 36
MMHC Max-Min Hill-Climbing. 29
MNIST "Mixed" National Institute of Standards and Technology SD-1,3. 8, 37, 47

OS Order Search. 29

PASS-GP Predictive Active Set Selection for Gaussian Processes. 35–39, 46, 47
PC Prototypical Constraint. 29
PCA Principal Component Analysis. 54

RSVM Reduced complexity SVM. 46
RVM Relevance Vector Machine. 6

SBMC Sparse Bayesian Multi-class Classifier. 36, 54
SC Sparse Candidate. 29
SLIM Sparse Linear Identifiable Multivariate Modeling. 8, 23–28, 30, 51, 52
SMC Sequential Monte Carlo. 21, 33
SNIM Sparse Non-linear Identifiable Multivariate Modeling. 8, 24–28, 46–48, 51
SVM Support Vector Machine. 39, 43, 46, 47

USPS The United States Postal Service. 8, 46, 47, 53

Notation

N	Number of observations in a dataset
d	Number of dimensions in a dataset
x	Scalar
$\mathbf{x}$	Column vector
x_i	The i-th element of $\mathbf{x}$
$\mathbf{x}_{\setminus i}$	Vector $\mathbf{x}$ with the i-th element removed
$\mathbf{X}$	Matrix, $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots]$
$\mathbf{X}^\top$	Transpose of $\mathbf{X}$
$\mathbf{X}^{-1}$	Inverse of $\mathbf{X}$
$\mathbf{P}$	Permutation matrix
$\mathrm{vec}(\mathbf{X})$	vectorized $\mathbf{X}$
$\odot$	Element-wise product
$\delta(x)$	Dirac δ-function
$\Gamma(a)$	Gamma function with parameter $a > 0$
$B(a, b)$	Beta function with parameters $a, b > 0$
$\mathcal{O}(N)$	Has order of N asymptotic complexity
$\mathbf{0}$	Vector of zeros
$\mathbf{1}$	Vector of ones
$\mathbf{I}$	Identity matrix
$\Phi(\cdot)$	Cumulative Gaussian function
$\Theta(\cdot)$	Heaviside step-function
$m_{\boldsymbol{\rho}}(\cdot)$	Mean function with parameters $\boldsymbol{\rho}$
$\mathbf{k}_{\boldsymbol{\upsilon}}(\cdot, \cdot)$	Covariance function with parameters $\boldsymbol{\upsilon}$

Contents

Chapter 1

Introduction

The purpose of modeling is in general terms making sense of particularities of different kinds of natural phenomena, not only biological but physical, social, and even artificially conceived. In this sense, our job is to propose models that help us to understand the way things behave so we can come up with testable hypothesis, emulate them, generalize them or even more ambitiously, to predict their future outcomes. Bayesian data modeling in particular has as core the imposition of structure to the data, meaning that we attempt to reflect what we know about the data and the way it was actually produced as a probabilistic object describing uncertainties about the interactions between every component in the system. From an analytical perspective, we take the model, observe some data we think has a similar generating mechanism and then use Bayesian machinery to *merge* both so we can then derive conclusions about model and data itself. This is in practice an iterative procedure in which repeated observation and careful analysis and intervention lead us to a refined model that better reflects the phenomenon under study.

Interpretability and identifiability are perhaps the most important features of a statistical model if what we really want is to reach a detailed understanding of the task we are dealing with. By interpretability we mean that we can directly relate the structure of the model, their variables and mutual interactions to the underlying mechanism of interest, so analysis of the model in the light of data can be translated into analysis of the actual phenomenon that produced the

data. Identifiability is sometimes thought more as a guarantee than a feature. It is indeed a promise that the set of ambiguities in the model are such that they will not interfere with its interpretability. Although these two flagship features often come at the price of somewhat simplified models, to our view it is better to have a simple identifiable model than an overly complicated one in which the number of possible interpretations of the results is as large as the model itself.

Without any doubt, we can say that any natural phenomenon we can think of can be seen as a potentially infinite complex network of interactions between the components of the system. Although we are interested in understanding everything in its full complexity, it is obviously a titanic task even for intrinsically small systems with apparently just a handful of interactions involved. Luckily for us, no matter how complex a system is, the amount of plausible interactions between its elements must be quite small if we compare it with the set of all imaginable possibilities. This very essential feature of any system we call here interchangeably *sparsity* or *parsimony*, brings hope when dealing with this apparent overwhelming complexity. Beyond this natural sparsity, from a practical perspective we can also find what we can call observed and expected sparsity. By observed sparsity we mean that the number of possible interactions in a model is limited by the portion of the system we observe or at least allowed to observe. This limitation does not only cover the variables we observe but any non-observed ones that might influence the subsystem that is under study. Expected sparsity on the other hand, it is a way of stating that the number of possible interactions in a model is limited by a function or *expectation* of the number of examples, observed and unobserved variables, parameters and the effort needed to infer and derive conclusions from the model with some reasonable degree of confidence.

This thesis is devoted to the kind of modeling where the various views of sparsity mentioned above are explicitly imposed in the model to exploit interpretability as much as possible. In terms of natural sparsity we are particularly interested on inferring structured interactions between sets of variables. Given that we see natural phenomena as networks, we also define our models as networks of variables in which connectivity not only imposes structure but gives purpose to the model. We say purpose because it is such a connectivity what we are finally interested in, thus needs to be inferred from data. In other words, learning connectivity is our hypothesis generating tool. Depending of the field of application and backbone structural features, interpretable models have many names in statistics and machine learning. For instance, in a model where all the variables are unobserved and their interactions are assumed as say undirected or directed, it is called *undirected or directed graph model*. When we have two sets of variables, one of them being observed and the interest resides on the interaction between observed and unobserved variables only, it is called *factor model* or *latent variable model*. If all variables are observed and we require to

infer interactions between them, it is called *clustering*. When a set of variables interact with a set of targets, all of them observed, it is called *multiple regression*. Similar to regression but with the targets being a set of discrete variables encoding groups of categories, it is known as *classification*. Finally, any combination of the previous is also possible and it is surely out there in the literature with some interesting application supporting it.

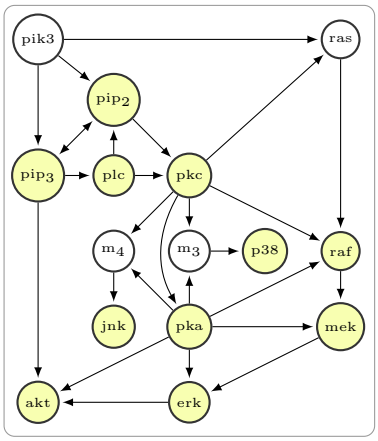


Figure 1.1: Classic protein signaling network (Sachs et al., 2005). The graph represents well accepted directed interactions (edges) between molecules (nodes). Lightly colored nodes denote directly observed proteins. White nodes are not observed but they indicate signaling nodes measured within contextual cellular pathways.

Some particular examples of the structural models described above and described in full later in this thesis include, *protein signaling networks* in which a set of partially observed set of proteins and phospholipids are known to interact in a causal manner. The challenge consists on inferring the set of directed interactions (causal) from quantitative measurements of a set of observable proteins and by acknowledging that there is another set of proteins that are involved in the system but that cannot be measured. Figure 1.1 shows the graphical representation of a classic protein signaling network (Sachs et al., 2005), see Appendix C. The shaded nodes represent observed nodes for which we have measurements and the remaining two (pik3 and ras) are known to be active in the system but cannot be measured. The model to be proposed is then a *directed acyclic graph* with *latent variables* that will attempt to infer the edges of the graph in Figure 1.1, i.e. the causal interactions.

In *proteomics* data analysis, we count with a set of observed isotope groups (peptides) known to be parts of a set of partially labeled proteins. The task consists

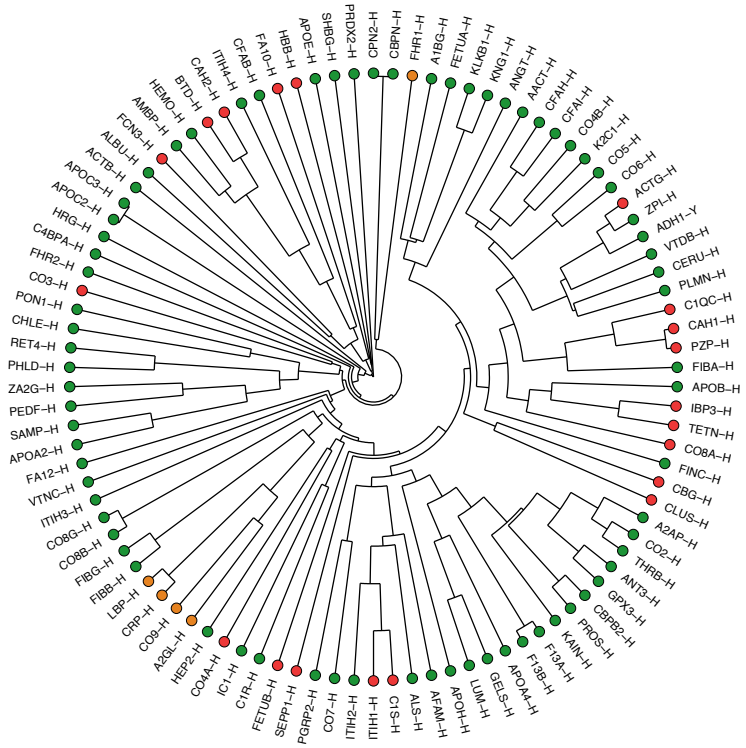


Figure 1.2: Latent protein tree. Each circle represents a human protein and the dendrogram, hierarchical interactions between them. Proteins colored in orange were found to be discriminant at distinguish H1N1/H3N2 from normal patients at different stages of the diseases. Proteins in red where found to match the profile of a different protein, suggesting that they were wrongly annotated in the initial phase of the study.

on finding interactions between isotope groups and proteins, and hopefully also protein-protein interactions. In Figure 1.2 we show a hierarchical representation of a protein-protein interaction structure obtained from a H1N1/H3N2 study, see Appendix E. The model consists basically of a *factor model* for the isotope group to protein assignments and a *hierarchical clustering* model for the protein-protein interactions. The problem comes with other difficulties as some of the isotope groups being unlabeled or wrongly labeled, there is plenty of missing values and outliers, and the technology used to obtain the measurements is know for producing systematic and batch related undesirable effects that need to be counteracted in order to obtain plausible biological interactions.

Most of Non-linear *classification* models are known for being computationally heavy, usually scaling badly with the size of the dataset. Since it is becoming normal to being able to build large datasets, we need models to efficiently handle the problems while still being able to enjoy the benefits of having plenty of data to work with. A well known case of the just described scenario is handwritten digits classification. This task is known for being (i) non-linear, (ii) data is easy to obtain so datasets are typically in the high ten thousands of examples and (iii) the input space is high dimensional but there is plenty of redundancy because digits can be represented as points in a lower dimensional manifold. The latter essentially justify that we should be able to build a classifier using a small portion of the dataset while still being able to successfully deliver state-of-the-art performance.

The landscape of sparsity priors

The great interest in modeling sparse phenomena has led researchers from the statistics and machine learning communities to propose a considerable amount of sparsity inducing prior distributions. The usual setting in sparse modeling is to weight sets of variables such that due to the assumptions made about the nature of the data, most of the weights are zero, i.e. variables with a zero weight do not contribute to the model. To support our assumption, we provide a prior distribution for the weights that places a substantial amount of probability mass around zero. The desired effect can be achieved in two different ways. One is to have an unimodal distribution centered around zero with most of its probability mass in the vicinity of zero, i.e. a heavy-tailed distribution. The alternative is to explicitly have probability mass at zero, i.e. a bimodal distribution that reflects that the weight can be either zero or something else with some probability.

Sparsity inducing unimodal distributions also called *shrinkage priors* are advantageous in several ways. Since the prior distributions are unimodal, absolutely continuous and admit scale mixture of Gaussians representations (Andrews and Mallows, 1974; West, 1987; Branco and Dey, 2001), they result in very simple and efficient sampling based inference procedures. Besides, continuity allows for very fast and scalable optimization inspired methods like Maximum a-Posteriori (MAP), empirical Bayes or variational Bayes (Bernardo and Smith, 1994; Gelman et al., 2004; Bishop, 2006). There are however some drawbacks, first of all the probability mass at zero is never positive, meaning that the posterior distribution of a variable with a shrinkage prior does not lead to a truly sparse solution. Secondly, the non-zero weights are represented by the tails of the distribution therefore their magnitude is usually underestimated. Fortunately, sparse solutions can be achieved by thresholding (Ishwaran and Rao, 2005) or as a result of using point estimates as in MAP (Tibshirani, 1996) or empirical

Bayes (Tipping, 2001) based models. As mentioned before, shrinkage priors admit scale mixture of Gaussians representation, meaning that different priors are obtaining by specifying different mixing densities. For example, an exponential mixing density leads to the Laplace or double exponential prior, very popular due to the seminal Least Absolute Shrinkage and Selection Operator (LASSO) for MAP based regression (Tibshirani, 1996), see also Park and Casella (2008) for its Bayesian counterpart. A Gamma mixing density results in a Student's t distribution that has the advantage of having tunable sparsity, i.e. it can be as sparse as a Cauchy distribution or not sparse at all like a Gaussian distribution. A well known application of this prior is the Relevance Vector Machine (RVM), a model for sparse regression and classification proposed by Tipping (2001). A more extreme approach is to use an improper prior for the variances of a Gaussian distribution to obtain the so called Gaussian-Jeffreys prior, very appealing because results in a parameterless prior. See for instance Figuereido (2003) for a general framework for supervised learning. A more recent alternative called horseshoe prior uses a half Cauchy mixing density. This prior has the particularity of having Cauchy-like tails and an infinitely tall spike at zero for an increased sparsity with less biased non-zero weights. Carvalho et al. (2010) introduces the prior and gives some examples in regression problems. Some other alternatives include the normal-exponential-gamma family (Griffin and Brown, 2005) and the normal-gamma or normal-inverse gamma priors (Caron and Doucet, 2008).

Sparsity priors using bimodal distributions usually called *slab and spike priors* are more correct than shrinkage priors in the sense that they attempt to model zero weights by putting probability mass directly at zero, thus needing two distributions one for zero weights called spike and the other for non-zero weights called slab. The most evident caveat of these priors is their bimodality and discreteness. Both conditions make inference more difficult because of the combinatorial growth of the search space w.r.t. the number of weights. There is a wide range of possibilities when specifying slab and spike priors. The earliest attempt is probably Lempers (1971) and Mitchell and Beauchamp (1988). They use a uniform distribution — the slab and a degenerate distribute at zero — the spike, in the context of regression. West (2003) replaced the slab distribution from uniform to normal and included a prior for the spike/slab trade-off probability for a problem of sparse factor modeling. Lucas et al. (2006) shows that a simple prior distribution for the trade-off probability in slab and spike priors might not be adequate in most cases, so they propose a hierarchical prior focused to weight matrices with column or row-wise sparsity sharing. There is also non-parametric versions of these priors for the case when the weight vector/matrix is in principle infinitely large, see for instance the Indian Buffet Process (IBP) and the Hierarchical Beta Process (HBP) proposed by Ghahramani et al. (2006) and Thibaux and Jordan (2007), respectively. The priors just mentioned have demonstrated to work well even in the ill-posed "large p small n" scenario, however some authors have tried to avoid the discreteness

of the slab and spike priors by replacing the degenerate distribution at zero with a narrow distribution, often Gaussian with small variance (see George and McCulloch, 1993, 1997; Ishwaran and Rao, 2005; Ishwaran and Papana, 2008). Although they tend to have better mixing properties than their discrete counterparts, they do not produce truly sparse solutions thus thresholding is required. See for example the "Zcut" thresholding rule presented by Ishwaran and Rao (2005).

In summary, shrinkage priors are perfect for applications with a large number of variables and limited computational budget are available, and where point estimates are sufficient. Slab and spike priors are better suited for detailed modeling scenarios where weight uncertainty assessment is more desirable, i.e. when we want to directly estimate the distribution of a particular weight of being zero. All in all, shrinkage priors are on the fast-practical side whereas slab and spike priors are more on the detailed-interpretable side.

Contributions

The five papers and one ongoing work included with this thesis are entirely dedicated to sparse modeling, however they still can be separated into two categories. Appendices A, C and E are oriented towards detailed sparse multivariate density modeling whereas the remaining three, namely B, D and F deal with supervised learning. In particular, Appendices B and D have to do with sparsity in Gaussian Process Classification (GPC) and Appendix F with variable selection in multi-category classifiers.

- Appendix A: ***Bayesian Sparse Factor Models and DAGs inference and comparison***. The paper presents a novel approach to learn Directed Acyclic Graphs (DAGs) and factor models within the same framework while also allowing for model comparison between them. It exploits the connection between factor models and DAGs to propose Bayesian hierarchies based on slab and spike priors to promote sparsity, heavy-tailed priors to ensure identifiability and predictive densities to perform the model comparison. The presented approach is demonstrated through extensive experiments on artificial and biological data showing that it outperforms a number of state-of-the-art methods.
- Appendix B: ***PASS-GP: Predictive Active Set Selection for Gaussian processes***. Proposes a new approximation method for Gaussian Process (GP) learning for large data sets that combines inline active set selection with hyperparameter optimization. The predictive distribution of the classifier is used for ranking the data points. It also uses the Leave-One-Out (LOO) (cavity) predictive distribution available in GPCs to make a common ranking

of both active and inactive points, allowing points to be removed again from the active set. The paper also lends both theoretical and empirical support to the active set selection strategy and marginal likelihood optimization on the active set. Experiments on the The United States Postal Service (USPS) and "Mixed" National Institute of Standards and Technology SD-1,3 (MNIST) digit classification databases demonstrate that the method can achieve state-of-the-art results with reasonable time complexity.

- Appendix C: ***Sparse Linear Identifiable Multivariate Modeling***. The paper considers sparse and identifiable linear latent variable (factor) and linear Bayesian network models for parsimonious analysis of multivariate data. It consists of a fully Bayesian hierarchy for sparse models using slab and spike priors, non-Gaussian latent factors and a stochastic search over the ordering of the variables. The framework, which we call Sparse Linear Identifiable Multivariate Modeling (SLIM), is validated and bench-marked on artificial and real biological data sets. In addition, the paper includes the model of Appendix A as special case and proposes two extensions to the basic i.i.d. linear framework: non-linear dependence on observed variables, called Sparse Non-linear Identifiable Multivariate Modeling (SNIM) and allowing for correlations between latent variables, called Correlated Sparse Linear Identifiable Multivariate Modeling (CSLIM), for the temporal and/or spatial data.
- Appendix D: ***Predictive Active Set Selection Methods for Gaussian processes***. The paper extends the work of Appendix A to an active set selection framework for Gaussian Process Classification for cases when the dataset is large enough to render its inference prohibitive. The Framework's backbone consists on a two step alternating procedure of active set update rules and hyperparameter optimization based upon evidence maximization. The active set update rules rely on the ability of the predictive distributions of a GP classifier to estimate the relative contribution of a datapoint when being either included or removed from the model. It specifically introduces two active set rules based on different criteria, the first one that prefers a model with interpretable active set parameters whereas the second puts computational complexity first, resulting in a model with active set parameters that directly control its complexity. Our extensive experiments show that the presented framework can compete with state-of-the-art classification techniques with reasonable time complexity.
- Appendix E: ***Latent Protein Trees***. Unbiased, label-free proteomics is becoming a powerful technique for measuring protein expression in almost any biological sample. The output of these measurements are a collection of features (10's to 100's of thousands, only some of which are identified) and their associated intensities for each sample. Each of the features are each associated with a particular polypeptide having a particular number of Carbon-13 atoms and a particular charge state. Because we know that subsets of features are from the same polypeptide, subsets of polypeptides are from the

same protein, and subsets of proteins are in the same biological pathways, we know that there is a very complex and informative correlational structure inherent in this data. However, attempts to model this data often focus on the identification of single features that are associated with a particular phenotype that is relevant to the experiment. These associations may be computed from hypothesis testing (with correction for multiple testing) or from various regression models. However, to date there have been no published approaches that appropriately model what we know to be multiple different levels of correlation structure. We present a hierarchical Bayesian model which is specifically designed to model the known correlation structure – both at the feature level and at the protein level – in unbiased, label-free proteomics. This model utilizes the partial identification information from peptide sequencing and database lookup as well as the observed correlation structure in the data set in order to appropriately compress features into meta-proteins and to estimate the correlation structure of those identified meta-proteins. We demonstrate the effectiveness of the model in the context of a series of proteomics measurements of serum plasma from a collection of volunteers who were infected with two different strains of viral influenza.

- Appendix F: **Sparse Bayesian multi-category classification**. Classification for the extreme ill-posed case of many more covariates than examples is still an open question despite much recent research within machine learning and statistics. Robust and high performance classifiers for this scenario is a key ingredient in diagnosis based upon gene expression profiling. Current practice often involves using (single gene) univariate tests as a feature extraction step prior to classification. This is in general suboptimal as it can miss important features for example in case when there is co-variation in the inputs. On the other hand, working directly with a high number of non-informative covariates increases the risk for overfitting for standard classifiers. The Appendix fully describes a sparse hierarchical Bayesian multi-category linear classifier especially well-suited for the more covariates than examples scenario arising prominently in classification of gene expression profiles. We show partial results that are at least as good as state-of-the-art linear and non-linear classification methods with and without prior covariate selection.

Each appendix contains the full version of the published work, pointers to the publicly available versions and a links to the complementary websites, where supplementary material and source codes can also be found.

Contributions not included in this thesis

- R. Borup, M. Rossing, R. Henao, Y. Yamamoto, A. Krogdahl, C. Godballe, O. Winther, K. Kiss, L. Christensen, E. Hgdall, F. Bennedbk and F. C.

Nielsen. Molecular Signatures of Thyroid Follicular Neoplasia. *Endocrine-Related Cancer*, 17 (3) 691–708, 2010

- C. Walder, R. Henao, M. Mørup and L. K Hansen. Semi-Supervised Kernel PCA. *ArXiv* (1008.1398, technical report), 2010.
- M. Rossing, R. Borup, R. Henao, O. Winther, J. Vikesaa, O. Niazi, C. Godballe, A. Krogdahl, M. Glud, C. Hjort-Sørensen, K. Kiss, F. N. Bennedbæk, F. C. Nielsen. Down-Regulation of microRNAs Controlling Cell Proliferation in Follicular Thyroid Carcinoma. Submitted to *Journal of Molecular Endocrinology*.

Outline

The remaining contents of this thesis are organized in four chapters and six appendices as it is described next

- Chapter 2 contains a short description the prior distributions used in this thesis, including parameterizations and some useful properties.
- Chapter 3 presents the developed models and methods divided in unsupervised and supervised modeling, together with their respective description, formulation, graphical model or algorithmic description and related work.
- Chapter 4 describes different case studies performed using the models introduced in Chapter 3.
- Chapter 5 presents the conclusions to this thesis and some open questions that might lead to future work.
- Appendices A, B, C, D, E and F contain all the details of the prior distributions, models, inference, practical considerations and applications from the previous chapters in the form of research work already published (3), submitted (1), in preparation (1) or in progress (1).

Chapter 2

Prior Distribution Compendium

Generally speaking, statistical methods constitutes a group of tools focused in making sense of the nature of some process using observed data, while also being able to provide insights about the future behavior of the process as new data is generated. Such an ambitious goal is tackled by specifying a probabilistic model that is assumed to describe the mechanism behind the data generation process. It is then a clear consequence that all the conclusions derived from such a model are conditioned on the model and its parameters.

In Bayesian statistics, probability is used as a measure of conditional uncertainty associated to a particular event likely to occur, given some a-prior information and a set of assumptions. In practice, we are interested in the probability of the event E given a dataset X, the assumptions A about the underlying mechanism producing X and contextual knowledge K if any, i.e. $p(E|X, A, K)$ is a measure of how likely it is for E to occur given a prefixed conditioning set $\{X, A, K\}$. More specifically, starting from a probabilistic model $p(X|\theta)$ that for some value of θ generates the data X, we set a probability distribution $p(\theta|K)$ that describes our prior beliefs about θ before the data is actually observed. It follows from Bayes' theorem that if the model is correct, we can capture all available information about θ in the form of the posterior distribution $p(\theta|X, A, K)$ once the

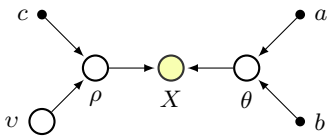


Figure 2.1: Graphical model of hierarchy in equation (2.2). Colored nodes, empty nodes and points represent observed variables, latent variables and hyperparameters, respectively.

data is observed, i.e.

$$p(\theta|X, A, K) = \frac{p(X|\theta)p(\theta|K)}{\int p(X|K)} \ , \tag{2.1}$$

where, the joint probability $p(X, \theta|K)$ in the numerator is the product of two distributions, the prior distribution $p(\theta|K)$ and the data likelihood $p(X|\theta)$, and the denominator is the sum of all possible choices of θ. One of the most appealing aspects of the Bayesian paradigm is that we can incorporate all sorts of hypothesis in the form of priors distributions. More particularly, the model $p(X|\theta)$ is not to be seen simply as a function parameterized by θ but as a multiparameter hierarchical model that aims to describe the underlying structure of X. For instance we can write a hierarchy as follows

$$\begin{aligned} X|\theta, \rho, \upsilon &\sim p(X|\theta, \rho, \upsilon) \ , \\ \theta|a, b &\sim p(\theta|a, b) \ , \\ \rho|\upsilon, c &\sim p(\rho|\upsilon, c) \ , \\ \upsilon &\sim p(\upsilon) \ , \end{aligned} \tag{2.2}$$

where we have defined that the data is generated by joint contribution of three sets of parameters $\{\theta, \rho, \upsilon\}$, θ occurs with probability $p(\theta|a, b)$, ρ depends on υ through $p(\rho|\upsilon, c)$ and $\{a, b, c\}$, called hyperparameters completely specify the behavior of $\{\theta, \rho, \upsilon\}$. Note that the hierarchical model in equation (2.2) with corresponding graphical model in Figure 2.1 implicitly highlights some structural features of the model, for example that θ does not depend on ρ but υ directly influences ρ. The remainder of the structure is encoded through the prior distributions and it can be as complex as we want or can afford to have. To name a few of the priors used in this thesis we could specify ρ as a non-linear function of υ with parameter c (Gaussian Process prior in Section 2.4), θ could be a sparse vector with sparsity rate a and precision b (slab and spike prior in Section 2.2), υ may be a collection of correlated variables assumed to admit a hierarchical structure (coalescent prior in Section 2.5), etc.

Learning in the Bayesian paradigm is conceptually simple, it is nothing but specifying a model $p(X|\theta)$, computing posterior distribution $p(\theta|X, A, K)$ and

then summarize θ, the parameters of interest in some appropriate way. From equation (2.1) it is possible to see that there are essentially three ways to approach the posterior distribution $p(\theta|X, A, K)$. The simplest path is to assume that every parameter setting is equally likely, thus the posterior distribution is simply proportional to the likelihood and the problem is reduced to find the set of parameters for which the data is most likely to occur, i.e. the maximum likelihood solution. When some information about the parameters is available meaning that we count with $p(\theta|K)$, and a point estimate of θ is enough to collect results, form equation (2.1), the posterior density is proportional to the likelihood times the prior and we can try to find the set of parameters for which the data is most likely to occur weighted by our prior beliefs about θ, i.e. the Maximum a-Posteriori solution. Finally, when we are interested in the entire distribution of θ, we compute $p(\theta|X, A, K)$ directly when possible or approximate it using for instance Markov Chain Monte Carlo (MCMC) based methods. This last approach is usually called the fully Bayesian approach.

Sometimes we are also interested in predicting the value of a future observation x generated by the same random process that produced X. Thus we need the so called predictive distribution $p(x|X, A, K)$ that encapsulates the uncertainty about x given $\{X, A, K\}$. Since x is assumed to be generated by the same process as X, thus x is a random sample of $p(x|\theta)$ and $p(x|X, A, K) = \int p(x|\theta)p(\theta|X, A, K)d\theta$. This means that x is an average of the distribution of x conditioned on the unknown value of θ weighted by the posterior distribution of θ given X. The predictive distribution is particularly useful for supervised learning, particularly when we are given a set of variables Z that are assumed to have enough information to predict the value of a set of targets X, thus we want to use a dataset $\{Z, X\}$ to estimate $p(\theta|Z, K)$ so we can then predict the distribution of a new target x, $p(x|z, Z, X, A, K)$, based only on the model and a new observation z.

In other cases we do not have a single hypothesis for the process generating X and we do not want to be forced to choose one of our alternatives, say A_1 and A_2, but let the data itself to decide. This practice called model comparison involves computing marginal likelihoods i.e. $p(X, A, K)$, where we have marginalized out all the parameters of the model. We can thus compare assumptions A_1 and A_2 based upon marginal likelihood ratios $p(X, A_1, K)/p(X, A_2, K)$, often called Bayes factors.

In this chapter we present short descriptions of every prior distribution used in this thesis along with their corresponding probability density functions. Further details of the priors is provided in the appendices or as external references otherwise. Most of the distributions presented here are standard in Bayesian statistics, still we present them here as an attempt to make easier to follow the description of the models presented in the next chapter.

2.1 Standard Priors

Gaussian Distribution

The Gaussian or normal distribution is probably considered the most essential of the probability distributions due to its attachment to the central limit theorem. The distribution is characterized by a symmetric bell shape of width $\sigma^2 > 0$ (variance) and centered at μ (mean). The probability density function can be written as

$$\mathcal{N}(x|\mu, \sigma^2) \;=\; \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \;. \tag{2.3}$$

The Gaussian distribution is also the conjugate to a Gaussian likelihood with known variance. This distribution can be easily turned into multivariate by replacing x and μ by vectors $\mathbf{x}$ and $\boldsymbol{\mu}$, respectively, and the variance σ^2 by a covariance (positive definite) matrix $\boldsymbol{\Sigma}$. See Gelman et al. (2004) for further details and properties.

Exponential Distribution

The exponential distribution is a one-sided continuous probability distribution for non-negative valued variables. It is of particular interest in this work as a mixing density for infinite scale mixture of Gaussian distributions (Andrews and Mallows, 1974). We can write its probability density function as

$$\text{Exponential}(x|\lambda) \;=\; \lambda \exp(-\lambda x) \;, \quad \text{for} \quad x \geq 0 \;, \tag{2.4}$$

where $\lambda > 0$ is its rate parameter. The mean and variance of the exponential distribution are λ^{-1} and λ^{-2}, respectively. See Gelman et al. (2004) for further details and properties.

Gamma Distribution

The gamma distribution is a non-symmetric, two-parameter continuous probability distribution commonly used in Bayesian analysis to describe inverse variances (precisions). This distribution is well known for being conjugate to several likelihood distributions, for instance the Poisson, exponential in equation (2.4), Gaussian with known mean in equation (2.3) and gamma with known shape

in equation (2.5). Its probability density function is expressed in terms of the gamma function and parameterized in terms of a shape $s > 0$ and rate $r > 0$

$$\text{Gamma}(x|s,r) \;=\; \frac{r^s}{\Gamma(s)} x^{s-1} \exp(-rx) \quad \text{for} \quad x \geq 0 \;, \tag{2.5}$$

where $\Gamma(s)$ is the gamma function and $s > 0$. The mean and variance of the gamma distribution are sr^{-1} and sr^{-2}, respectively. See Gelman et al. (2004) for further details and properties.

Bernoulli Distribution

The Bernoulli distribution is a very simple discrete probability distribution that takes value 1 with probability p and 0 with probability $1 - p$. It is particularly useful to represent binary matrices or vectors with independent elements. The probability mass function is then

$$\text{Bernoulli}(x|p) \;=\; p^x (1-p)^{1-k} \;, \quad \text{for} \quad x \in \{0, 1\} \;. \tag{2.6}$$

The mean and variance of this distribution are p and $p(1 - p)$, respectively. When the variable x takes more that two distinct values we can extend the bernoulli distribution as

$$\text{Discrete}(x|\mathbf{p}) \;=\; \prod_{i=k}^{K} p_k^{z_k} \;, \tag{2.7}$$

where $x \in \{1, \ldots, K\}$, $\mathbf{p} = [p_1, \ldots, p_K]$, $\mathbf{z} = [z_1, \ldots, z_K]$ is an auxiliary variable such that $z_k = 1$ if $x = k$ or zero otherwise, and $\sum_k p_k = 1$. See Gelman et al. (2004) for further details and properties.

Beta Distribution

The Beta distribution is defined for variables within the $(0, 1)$ interval and it is parameterized by two positive parameters, a and b. Since the domain of the probability distribution can be seen as a probability, can be used to describe the distribution of an unknown probability quantity. In fact, the beta distribution is the conjugate prior for a Bernoulli likelihood with unknown p, see equation (2.6). The probability density function of this distribution is

$$\text{Beta}(x|a,b) \;=\; \frac{1}{B(a,b)} x^{a-1} (1-x)^{b-1} \;, \quad \text{for} \quad x \in (0, 1) \;, \tag{2.8}$$

where $B(a, b)$ is the beta function with $a, b > 0$. Given that the mean and variance of the beta distribution are $a(a+b)^{-1}$ and $ab(a+b)^{-2}(a+b+1)^{-1}$, we can re-parameterize the beta distribution in equation (2.8) as $\text{Beta}(x, pm, p(1-m))$ so that m is the mean value and p acts as a pseudo count or precision-like parameter that enforces m (see for instance Lucas et al., 2006; Carvalho et al., 2008; Henao and Winther, 2009). Additionally, see Gelman et al. (2004) for further details and properties.

Dirichlet Distribution

The Dirichlet distribution is the multivariate generalization of the beta distribution in equation (2.8). It is parameterized by a vector of K positive values $\boldsymbol{\alpha}$. It is widely used in Bayesian statistics as conjugate prior to the discrete distribution in equation (2.7) and the multinomial distribution. For a $[0, 1]$ valued vector $\mathbf{x}$, its distribution is confined to a simplex of dimensionality $K-1$ and we can write the probability density function as

$$\text{Dirichlet}(\mathbf{x}|\boldsymbol{\alpha}) \;=\; \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\ldots\Gamma(\alpha_K)} \prod_{k=1}^{K} x_k^{\alpha_k - 1} \;,$$

where $\alpha_0 = \sum_k \alpha_k$. When $\alpha = \alpha_1 = \ldots = \alpha_k$, the parameter α is called concentration parameter and when $\alpha = 1$, the Dirichlet distribution is equivalent to a uniform distribution in the $(k-1)$-simplex. See Gelman et al. (2004) for further details and properties.

Laplace Distribution

The Laplace or double exponential distribution is a heavy-tailed continuous probability distribution parameterized by a location parameter μ and a rate $\lambda > 0$. Because of its fat tails is commonly used as prior for sparse variables as it places most of its probability mass close to zero (see Tibshirani, 1996; Park and Casella, 2008). We can write the probability density function as

$$\text{Laplace}(x|\mu, \lambda) \;=\; \frac{\lambda}{2} \exp(-\lambda|x - \mu|) \;.$$

The mean and variance of this distribution are μ and $2\lambda^2$, respectively. The Laplace distribution is not conjugate to any likelihood in close form, however Andrews and Mallows (1974) showed that it can be represented as an infinite scale mixture of Gaussian distributions with exponential mixing density —

equation (2.4), i.e.

$$\text{Laplace}(x|\mu,\lambda) \;=\; \int \mathcal{N}(x,\mu,\sigma^2)\text{Exponential}(\sigma^2|\lambda^2)d\sigma^2 \;. \qquad (2.9)$$

This representation is very convenient for MCMC Gibbs sampling based inference because conjugacy can be fully exploited provided that the conditional posterior of σ^{-2} in equation (2.9) has an inverse Gaussian distribution with parameters $\mu > 0$ and $\lambda > 0$

$$\text{IG}(x|\mu,\lambda) \;=\; \left(\frac{\lambda}{2\pi x^3}\right)^{\frac{1}{2}} \exp\left\{-\frac{\lambda(x-\mu)^2}{2\mu^2 x}\right\} \;, \quad \text{for} \quad x > 0 \;,$$

where μ is the mean and $\mu^3\lambda^{-1}$ is the variance. See Chhikara and Folks (1989) for further details and properties.

Student's *t* Distribution

The Student's t distribution is a symmetric bell-shaped distribution just like the Gaussian distribution, only with heavier tails controlled by a degrees of freedom parameter $\theta > 0$. This power-law distribution has as special cases the Cauchy distribution when $\theta = 1$ and the Gaussian distribution in the limiting case, i.e. $\theta \to \infty$. Its three parameter version also allows for a location parameter μ and a scaled variance σ^2. The probability density function is defined as

$$t(x|\mu,\sigma^2,\theta) \;=\; \frac{\Gamma(\frac{\theta+1}{2})}{\Gamma(\frac{\theta}{2})}\frac{1}{\sqrt{\pi\sigma^2}}\left(1+\frac{(x-\mu)^2}{\sigma^2\theta}\right)^{-\frac{\theta+1}{2}} \;,$$

where μ is only defined for $\theta > 1$ and the variance is $\sigma^2\theta(\theta-2)^2$, for $\theta > 2$. The t distribution has also an infinite scale mixture of Gaussian representation, this time with a gamma mixing density — equation (2.5),

$$t(x|\mu,\sigma^2,\theta) \;=\; \int \mathcal{N}(x|\mu,\upsilon\sigma^2)\text{Gamma}\left(\upsilon^{-1}\middle|\frac{\theta}{2},\frac{\theta}{2}\right)d\upsilon \;, \qquad (2.10)$$

that allows for efficient sampling based inference (see Andrews and Mallows, 1974).

2.2 Slab and Spike Prior

The slab and spike priors are in essence two component mixtures of a continuos part and a δ-function (or a very narrow continuous distribution) at zero

(Lempers, 1971; Mitchell and Beauchamp, 1988; George and McCulloch, 1993; Geweke, 1996; George and McCulloch, 1997; West, 2003). The idea is to represent separately the elements of the continuous variable $\mathbf{x}$ as a non-zero magnitude and as being or not equal to zero. We can write the so called single layer slab and spike hierarchy as

$$\begin{aligned} x_i|r_i,\cdot &\sim (1-r_i)\delta(x_i) + r_i\text{Cont}(x_i|\cdot) \ , \\ r_i|\nu &\sim \text{Bernoulli}(r_i|\nu) \ , \\ \nu|m,p &\sim \text{Beta}(\nu|pm, p(1-m)) \ , \end{aligned} \tag{2.11}$$

where r is a binary indicator of whether $x \neq 0$, $\delta(\cdot)$ is a Dirac δ-function, $\text{Cont}(\cdot)$ is the continuous slab component, $1-\nu$ is the probability that $x = 0$, $\text{Bernoulli}(\cdot)$ and $\text{Beta}(\cdot)$ are the distributions in equations (2.6) and (2.8), respectively.

When $\mathbf{x}$ in equation (2.11) is part of a matrix $\mathbf{X}$ it is not desirable to have a single set of hyperparameters $\{p, m\}$ (pseudo count, mean) for all elements, columns or rows of the matrix. For example, it is very likely that some of the columns of $\mathbf{X}$ turn out to be very sparse but some others not, thus the prior needs to diffuse enough to support both cases, as a result, the conditional $p(r|x,\cdot)$ will be quite spread over the unit interval, rendering interpretation rather difficult. This suggests that setting the hyperparameters to achieve a sensible overall sparsity level might be very complicated in practice. Ideally, we would like to have a model with a high/low sparsity levels with arbitrary certainty about them being or not equal to zero. As pointed out by Lucas et al. (2006) and Carvalho et al. (2008) this undesirable behavior can be avoided by introducing an additional (two layer) slab and spike layer as

$$\begin{aligned} x_i|r_i,\cdot &\sim (1-r_i)\delta(x_i) + r_i\text{Cont}(x_i|\cdot) \ , \\ r_i|\eta_i &\sim \text{Bernoulli}(r_i|\eta_i) \ , \\ \eta_i|q_i,a_m,a_p &\sim (1-q_i)\delta(\eta_i) + q_i\text{Beta}(\eta_i|a_p a_m, a_p(1-a_m)) \ , \\ q_i|\nu &\sim \text{Bernoulli}(q_i|\nu) \ , \\ \nu|b_m,b_p &\sim \text{Beta}(\nu|b_p b_m, b_p(1-b_m)) \ , \end{aligned} \tag{2.12}$$

where we obtained that η_i (the probability of $x_i \neq 0$) is either zero exactly or non-zero from a beta distribution with hyperparameters $\{a_m, p_m\}$. If $\{a_m, p_m\}$ are set to relatively large values we can expect large probabilities for non-zero elements of $\mathbf{X}$ which is precisely the desired behavior. The bottom layer in equation (2.12) still controls the overall sparsity in an indirect way by setting our a-priori expectations about obtaining exact zeros in η_i, thus also in x_i.

When $\mathbf{X}$ or $\mathbf{x}$ happen to be the mean of a Gaussian likelihood we can simply set $\text{Cont}(\cdot)$ to the Gaussian distribution from equation (2.3). This particular two layer slab and spike parameterization has been successfully used as prior

distribution in sparse factor and multiple regression models by Lucas et al. (2006); Carvalho et al. (2008), and for general linear Bayesian networks by Henao and Winther (2009, 2011).

2.3 Order Search Prior

A prior distribution for a permutation matrix $\mathbf{P}$ (doubly stochastic binary matrix) of size $d \times d$ is essentially a distribution for all possible $d!$ permutations of the index ordering vector $[1\ 2\ \ldots\ d]$. In practice we usually do not have any prior knowledge about the distribution of $\mathbf{P}$ thus we have to necessarily adopt a uniform distribution. The problem is that drawing uniform permutations from a uniform distribution in a Metropolis-Hastings (M-H) setting will produce a very low acceptance rate because the search space is combinatory large and from all possible permutations, most of them will have zero support. Our proposal is to draw permutation matrices from $q(\mathbf{P}^\star|\mathbf{P})$, i.e. propose a new permutation $\mathbf{P}^\star$ as a small modification of the current value $\mathbf{P}$. Particularly we use a uniform random transpositions, meaning that we exchange two randomly selected elements of the index vector of d elements. Since we have no a-priori preferred permutation, we may use a M-H acceptance probability $\min(1, \xi_{\to\star})$ with $\xi_{\to\star}$ as a simple ratio of likelihoods in terms of $\mathbf{P}^\star$ and $\mathbf{P}$. This simple approach for sampling permutation matrices has been shown to be useful at least for small dimensions ($d < 100$). See Appendix C for additional details.

2.4 Gaussian Process Prior

A Gaussian Process is a collection of random variables, any finite number of which have a joint Gaussian distribution. For a continuous function $f(x_1, x_2, \ldots)$, the process is completely specified through a mean $m_{\boldsymbol{\rho}}(x_1, x_2, \ldots)$ and a covariance function $k_{\boldsymbol{\upsilon}}(x_i, x_j)$, with parameters $\{\boldsymbol{\rho}, \boldsymbol{\upsilon}\}$. We can thus write

$$y_1, \ldots, y_N, \ldots \sim \mathrm{GP}(y_1, \ldots, y_N, \ldots | m_{\boldsymbol{\rho}}(x_1, x_2, \ldots), k_{\boldsymbol{\upsilon}}(\cdot, \cdot)) \ , \tag{2.13}$$

where the random variable $\mathbf{y} = y_1, \ldots, y_N, \ldots$ represents the value of the function $f(\mathbf{x})$ at location $\mathbf{x}$. Conceptually, the GP is as a prior distribution over continuous functions, in practice however due to the definition a GP, it is a multivariate distribution defined for the values of $f(\mathbf{x})$, $m_{\boldsymbol{\rho}}(\mathbf{x})$ and $k_{\boldsymbol{\upsilon}}(\mathbf{x}, \mathbf{x}')$ at $\mathbf{x}$. The prior distribution in equation (2.13) is very flexible mainly because we can choose the mean and covariance functions depending on the application or available side information. For instance, the covariance function could depend

on time for time series modeling, or spatial information or some other covariates of interest. For finite number of locations $\mathbf{x}$, this prior is conjugate to multivariate Gaussian likelihoods with known variance, making them well suited for applications like regression and classification. For a more complete panorama of Gaussian Process in machine leaning see Rasmussen and Williams (2006).

Assuming that selecting the mean and covariance functions is not an issue, it remains to solve the problem of choosing their parameters $\{\boldsymbol{\rho}, \boldsymbol{\upsilon}\}$. In this thesis we follow two different paths depending on the task to be solved. For multivariate density modeling we opt for fully Bayesian analysis with MCMC based inference, thus we place a hierarchical prior on $\boldsymbol{\upsilon} = [\upsilon_1, \ldots, \upsilon_j, \ldots]$ as follows

$$\upsilon_j | u_s, \kappa \;\sim\; \text{Gamma}(\upsilon_j | u_s, \kappa) \;, \quad \kappa | k_s, k_r \;\sim\; \text{Gamma}(\kappa | k_s, k_r) \;, \tag{2.14}$$

where the random variable κ is shared by all the parameters in $\boldsymbol{\upsilon}$ and $\{u_s, k_s, k_r\}$ is the set of hyperparameters. Given that the conditional distribution of $\boldsymbol{\upsilon}$ is not of any standard closed form, M-H updates are used. See Appendix C for the details about inference and Gelman et al. (2004) for the generalities about the Metropolis-Hastings algorithm.

In supervised modeling, we are in general more interested in making predictions than understanding or finding underlying structures in the data at hand. This is the reason why we consider empirical Bayes as a viable alternative to a fully Bayesian treatment. In empirical Bayes, the hyperparameters of a hierarchical model are set to the most likely ones instead of being marginalized out (Casella, 1985; Bernardo and Smith, 1994; Gelman et al., 2004). It is also known in statistics as maximum marginal likelihood, maximum likelihood type II (Berger, 1985), generalized marginal likelihood (Whaba, 1975) and evidence approximation (MacKay, 1992; Bishop, 2006) in machine learning. In GPC specifically, for a dataset with N observations, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ and a vector of labels $\mathbf{y}$ we compute the marginal likelihood as

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\rho}, \boldsymbol{\upsilon}) \;=\; \int p(\mathbf{y}|\mathbf{f}, \mathbf{X}) p(\mathbf{f}|\mathbf{X}, \boldsymbol{\rho}, \boldsymbol{\upsilon}) d\mathbf{f} \;, \tag{2.15}$$

where $\mathbf{f}$ is a vector with the values of $f(x)$ at the locations given by $\mathbf{X}$ and $p(\mathbf{f}|\mathbf{X}, \boldsymbol{\upsilon})$ is a GP prior as in equation (2.13). Empirical Bayes follows by maximizing equation (2.15) w.r.t. the mean and covariance function parameters $\{\boldsymbol{\rho}, \boldsymbol{\upsilon}\}$ using any suitable optimization technique, like gradient descent (see Rasmussen and Williams, 2006).

2.5 Kingman's Coalescent

Kingman's coalescent is a convenient and powerful method for describing the ancestral tree of a set of individuals (Kingman, 1982a). The process as explained by Kingman is referred to as the coalescent because it describes the probability of coalescent events, i.e. the time point in the genealogy where two individuals merge. For k individuals, Kingman's k-coalescent is nothing but a distribution over genealogies of those k individuals considered or equivalently, over binary trees (genealogies) with k leaves. In particular, the k-coalescent is a continuous-time Markov process that starts at time $t = 0$ with all $\{1, \ldots, k\}$ individuals and evolves back in time, merging pairs of elements until only one is left. Every pair of individuals coalesce independently with rate 1, thus the time between events j and $j-1$ is $\Delta_j \sim \exp\left(\frac{k-j+1}{2}\right)$ and the pair to be merged is chosen uniformly from those available at time $i-1$. With probability one, a random sample from the k-coalescent is a binary tree π with a single root at time $t = -\infty$ and the initial k individuals at time $t = 0$. We can write then

$$p(\pi) = \prod_{j=1}^{k-1} \exp\left(-\left(\frac{k-j+1}{2}\Delta_j\right)\right) \ .$$

Among the interesting properties of this prior distribution we have that the marginal distribution over tree structures is uniform and independent of the merging times and it is infinitely exchangeable. See Kingman (1982b) and Kingman (1982a) for further details.

The prior over the tree structure π is not enough in practice to specify a model, we still need a Markov process to evolve over the tree. Here we use a Brownian diffusion with underlying diffusion covariance Φ. This results in a Gaussian transition density with mean $\boldsymbol{\mu}_j$ and covariance $\Delta_j\Phi$, where $\boldsymbol{\mu}_j$ is the state of the j-th node in the tree and Δ_j is the time elapsed between nodes j and $j-1$. See Appendix E for further details about the formulation and how to perform inference in this setting using Sequential Monte Carlo (SMC) with multinomial resampling (Doucet et al., 2001).

Chapter 3

Models

The models presented in this chapter employ sparsity as leading structural feature and complemented by other sources of prior knowledge which are included depending of the application and specifics of the data at hand, e.g. time series, power law behavior, discrete valued variables, non-linearities, missing values, etc. Models are divided into two main categories, namely unsupervised and supervised. The first kind of models is targeting structured multivariate density modeling under various generative assumptions including factor models, DAGs and general Bayesian networks. The second kind is aimed to model the relationship between observations of a dataset and a group of categories in which they can be grouped. This is conceptually done using the assumption that the set of covariates in the data contains enough information to separate a set of groups that comes in the form of a discrete labeling variable, while also being able to generalize, i.e. to predict the label of new observations not used to infer the model parameters.

3.1 Unsupervised Modeling

We present four different but related generative models for unsupervised modeling that employ different prior assumptions to promote particular subjacent substructures in the data of interest. In particular, SLIM assumes the data can

be represented as an identifiable combination of sparse factor models and sparse DAGs, where each of the latent variables or factors is i.i.d. with heavy-tailed distributions. CSLIM departs from SLIM in the sense that the latent variables are no longer i.i.d. but allows for observation-wise correlations using GP priors, making it appropriate for times series or spatial data modeling. SNIM has the same functionality as SLIM, however it assumes that the DAG representation of the random variables in the data allows for non-linear interactions, also implemented through GP priors. Lastly, Latent Protein Tree (LPT) is in essence a specialized factor model in which the latent factors correlate with each other in a hierarchical fashion, making it capable of building a tree structure over the latent variables involved. In this model we renamed the latent variables to latent proteins because the model allows for such an interpretation since it was originally designed for modeling proteomics data.

Apart from the distinguishing features of the models presented in this Section, they all share the same core built upon sparsity. In particular, we assume that a given observed variable can be explained by a small collection of additional variables, meaning that in any case our models can be represented as sparse graphs with variables as nodes and a small set of interactions as edges (see Figure 1.1). Whether the additional variables are also observed or need to be inferred depends on the mechanism used as generative process. Our main concern and most important feature is without a doubt interpretability, i.e. how can we relate inferred random variable relationships to real life structures like transcriptional networks, signaling networks, peptide-protein or protein-protein interactions, etc. To give an example, in proteomics we are given peptide measurements but we are interested in protein activities, thus our job is to be able to obtain protein profiles from observed sets of peptides. We know that a peptide is likely to be explained by a single protein that needs to be selected from a large pool of candidates. The set of all possible peptide-protein interactions conforms a sparse connectivity matrix and we have to estimate it (see Section 3.1.2).

Figure 3.1 shows a brief summary of the four models including generative assumptions and leading features. The remainder of this section is based on three publications presented in Appendices A, C and E.

PUB

3.1.1 Sparse Identifiable Multivariate Modeling

We consider first a model for a fairly general class of linear Bayesian networks by putting together a linear DAG, $\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{z}$, up to some restrictions in $\mathbf{B}$, and a standard factor model, $\mathbf{x} = \mathbf{C}\mathbf{z} + \boldsymbol{\epsilon}$. Our goal is to explain each one of d observed variables $\mathbf{x}$ as a linear combination of the remaining ones, a set of

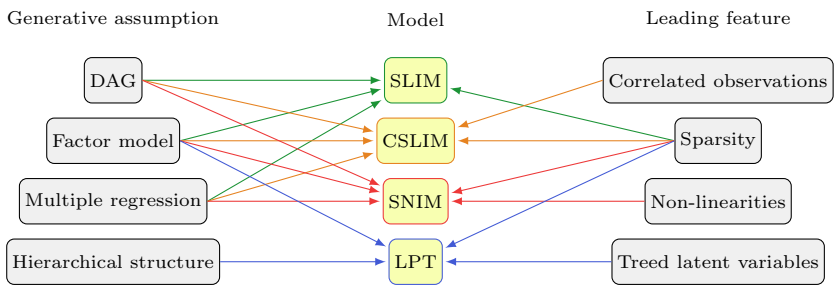


Figure 3.1: Unsupervised model's summary. Bayesian networks include as special cases: factor models, DAGs, multiple regression and their combinations as special cases.

$d+m$ independent latent variables $\mathbf{z}$ and additive noise $\boldsymbol{\epsilon}$. We have then

$$\mathbf{x} = (\mathbf{R} \odot \mathbf{B})\mathbf{x} + (\mathbf{Q} \odot \mathbf{C})\mathbf{z} + \boldsymbol{\epsilon} \ , \tag{3.1}$$

where $\odot$ is the element-wise product and we can further identify the following elements:

- $\mathbf{z}$ is partitioned into two subsets, $\mathbf{z}_D$ is a set of d driving signals for each observed variable in $\mathbf{x}$ and $\mathbf{z}_L$ is a set of m shared general purpose latent variables. $\mathbf{z}_D$ is used here to describe the intrinsic behavior of the observed variables that cannot regarded as "external" noise.
- $\mathbf{R}$ is a $d \times d$ binary connectivity matrix that encodes whether there is an edge between observed variables, by means of $r_{ij} = 1$ if $x_i \to x_j$. Since every non-zero element in $\mathbf{R}$ is an edge of a DAG, $r_{ii} = 0$ and $r_{ij} = 0$ if $r_{ji} \neq 0$ to avoid self-interactions and bi-directional edges, respectively. This also implies that there is at least one variable ordering leading to a permutation matrix $\mathbf{P}$ such that $\mathbf{P}^\top \mathbf{R} \mathbf{P}$ is strictly lower triangular. We have used that $\mathbf{P}$ is orthonormal then $\mathbf{P}^{-1} = \mathbf{P}^\top$.
- $\mathbf{Q} = [\mathbf{Q}_D \ \mathbf{Q}_L]$ is a $d \times (d+m)$ binary connectivity matrix, this time for the conditional independence relations between observed and latent variables. We assume that each observed variable has a dedicated latent variable, thus the first d columns of $\mathbf{Q}_D$ are the identity. The remaining m columns can be arbitrarily specified, by means of $q_{ij} \neq 0$ if there is an edge between x_i and z_j for $d < j \leq m$.
- $\mathbf{B}$ and $\mathbf{C} = [\mathbf{C}_L \ \mathbf{C}_D]$ are respectively, $d \times d$ and $d \times (d+m)$ weight matrices containing the edge strengths for the Bayesian network. Their elements are constrained to be non-zero only if their corresponding connectivities are also non-zero.

The model in equation (3.1) that we call SLIM has to start with two important special cases, (i) if all elements in $\mathbf{R}$ and $\mathbf{Q}_D$ are zero it becomes a standard factor model and (ii) if $m = 0$ or all elements in $\mathbf{Q}_L$ are zero it is a pure DAG.

Identifiability is a very important ingredient of SLIM, hence we need to establish clearly under which conditions each of its components can be readily identified. Using the results provided by Kagan et al. (1973), it can be shown that the standard factor model and the pure DAG part of equation (3.1) are independently identifiable if the set of driving signals in the DAG and at least $m - 1$ latent variables in the factor model are non-Gaussian distributed (up to scale and permutations of the columns of $\mathbf{B}$ and $\mathbf{C}$). For the general case in equation (3.1), Henao and Winther (2011) showed that the model is identifiable if $\mathbf{z}$ is non-Gaussian and the distributions of the driving signals $\mathbf{z}_D$ and the general purpose latent variables $\mathbf{z}_L$ differ beyond scaling. This means in other words that assuming substantially different prior distributions for these two sets of variables is required to ensure identifiability.

For the case where independence of observed variables cannot be granted, for instance due to presence of (time) correlations or smoothness in the data, the i.i.d. assumption made for the latent variables and driving signals does not apply anymore, fortunately the only modification we need to make in the original model is to allow elements in rows of $\mathbf{Z} = [\mathbf{z}_1 \dots \mathbf{z}_N]$ to correlate. In particular, we assume independent Gaussian Process priors for each latent variable instead of independent univariate distributions, to conform what we have called Correlated Sparse Linear Identifiable Multivariate Modeling (CSLIM).

Furthermore, provided that we know the true ordering of the variables needed to build a DAG, i.e. the permutation matrix $\mathbf{P}$ is known then $\mathbf{B}$ is surely strictly lower triangular. As a result, it is very easy to allow for non-linear interactions in the DAG part from the model in equation (3.1) by rewriting it as

$$\mathbf{P}\mathbf{x} = (\mathbf{R} \odot \mathbf{B})\mathbf{P}\mathbf{y} + (\mathbf{Q} \odot \mathbf{C})\mathbf{z} + \boldsymbol{\epsilon} \ , \tag{3.2}$$

where $\mathbf{y} = [y_1, \dots, y_d]^\top$ and $y_{i1}, \dots, y_{iN}$ has jointly a GP prior. This is a straight forward extension that we call Sparse Non-linear Identifiable Multivariate Modeling (SNIM), which is in spirit similar to Friedman and Nachman (2000); Hoyer et al. (2009); Zhang and Hyvärinen (2009, 2010); Tillman et al. (2009), however instead of treating the inherent multiple regression problem in equation (3.2) and the conditional independence of the observed variables due to the DAG assumption independently, we proceed within our SLIM framework by letting the multiple regressor be sparse, thus the conditional independences are encoded through $\mathbf{R}$.

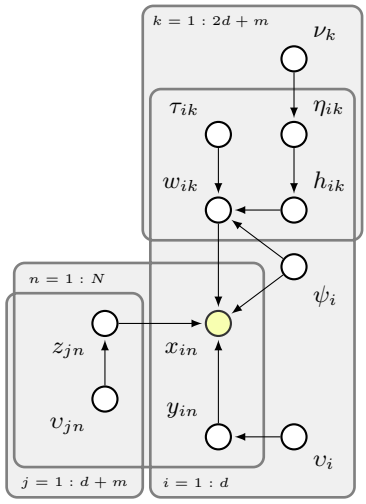


Figure 3.2: Graphical model for SLIM, CSLIM and SNIM.

The three models described above: SLIM, CSLIM and SNIM can be summarized as the graphical model in Figure 3.2 and the following probabilistic hierarchy

$$
\begin{aligned}
\mathbf{x}_n|\mathbf{W},\mathbf{y}_n,\mathbf{z}_n,\mathbf{\Psi} &\sim \mathcal{N}(\mathbf{x}_n|\mathbf{W}[\mathbf{y}_n\ \mathbf{z}_n]^\top,\mathbf{\Psi})\ , \quad \mathbf{W}=[\mathbf{B}\ \mathbf{C}]\ ,\\
\psi_i^{-1}|s_s,s_r &\sim \text{Gamma}(\psi_i^{-1}|s_s,s_r)\ ,\\
w_{ik}|h_{ik},\psi_i,\tau_{ik} &\sim (1-h_{ik})\delta_0(w_{ik}) + h_{ik}\mathcal{N}(w_{ik}|0,\psi_i\tau_{ik})\ ,\\
h_{ik}|\eta_{ik} &\sim \text{Bernoulli}(h_{ik}|\eta_{ik})\ , \quad \mathbf{H}=[\mathbf{R}\ \mathbf{Q}]\ ,\\
\eta_{ik}|\nu_k,\alpha_p,\alpha_m &\sim (1-\nu_k)\delta(\eta_{ik}) + \nu_k\text{Beta}(\eta_{ik}|\alpha_p\alpha_m,\alpha_p(1-\alpha_m))\ ,\\
\nu_k|\beta_m,\beta_p &\sim \text{Beta}(\nu_k|\beta_p\beta_m,\beta_p(1-\beta_m))\\
\tau_{ik}^{-1}|t_s,t_r &\sim \text{Gamma}(\tau_{ik}^{-1}|t_s,t_r)\ ,\\
z_{j1},\ldots,z_{jN}|\upsilon &\sim \begin{cases} \prod_n \mathcal{N}(z_{jn}|0,\upsilon_{jn})\ , & \text{(SLIM)}\\ \text{GP}(z_{j1},\ldots,z_{jN}|k_{\upsilon_j,n}(\cdot))\ , & \text{(CSLIM)}\end{cases}\\
y_{i1},\ldots,y_{iN}|\upsilon &\sim \begin{cases} x_{i1},\ldots,x_{iN}\ , & \text{(SLIM)}\\ \text{GP}(y_{i1},\ldots,y_{iN}|k_{\upsilon_i,x}(\cdot))\ , & \text{(SNIM)}\end{cases}
\end{aligned}
$$

where $\boldsymbol{\epsilon}$ in equation (3.1) has been marginalized out using $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0},\mathbf{\Psi})$, and we have omitted $\mathbf{P}$ and the hyperparameters in the graphical model. Latent variable and driving signal parameters υ can have one of several prior distributions: Exponential$(\upsilon|\lambda^2)$ (Laplace), Gamma$(\upsilon^{-1}|\theta/2,\theta/2)$ (Student's t) or Gamma$(\upsilon|u_s,\kappa)$ (GP), see equations (2.9), (2.10) and (2.14), respectively. We write $k_{\upsilon_j,n}(\cdot)$ and $k_{\upsilon_j,x}(\cdot)$ to make explicit that CSLIM has a covariance function

that depends upon an external indexing variable n that could be for instance time and that SNIM depends directly on the observed data x. The latent variables/driving signals z_{jn} and the mixing/connectivity matrices with elements c_{ij} or b_{ij} are modeled independently. $\mathbf{W}$ has the two layer slab and spike prior from Section 2.2, thus each element in $\mathbf{B}$ and $\mathbf{C}$ have its own slab variance τ_{ij} and probability of being non-zero η_{ij}. Moreover, there is a shared sparsity rate per column ν_k. Variables υ_{jn} are variances if z_{jn} use a scale mixture of Gaussian's representation, or length scales in the GP prior scenario. Since we assume no sparsity for the driving signals, $\eta_{ik} = 1$ for $d + i = k$ and $\eta_{ik} = 0$ for $d + i \neq k$. The permutation matrix $\mathbf{P}$ needed for the DAG representation is provided with the order search prior from Section 2.3. In addition, we can recover the pure DAG by making $m = 0$ and the standard factor model by making instead $\eta_{ik} = 0$ for $k \leq 2d$.

Quantitative model comparison between members of SLIM and its extensions is one of our main concerns because it not only can be used as model selection yardstick but as hypothesis-generating tool. In our case, this is a very difficult task because the marginal likelihood cannot be written as an average over posterior distributions in a simple way. We know that it is still possible using MCMC methods, for example by partitioning of the parameter space and multiple chains or thermodynamic integration (see Chib, 1995; Neal, 2001; Murray, 2007; Friel and Pettitt, 2008), but in general it must be considered as computationally expensive and non-trivial. On the other hand, evaluating the likelihood on a test set $\mathbf{X}^\star$, using predictive densities $p(\mathbf{X}^\star|\mathbf{X}, \mathcal{M})$, where $\mathcal{M}$ is the model to be tested, is simpler from a computational point of view because it can be written in terms of an average over the posterior of the *intensive variables*, $p(\mathbf{W}, \boldsymbol{\epsilon}, \cdot|\mathbf{X})$ and the prior distribution of the *extensive variables* associated with the test points[1], $p(\mathbf{Z}^\star|\cdot)$. This average can be approximated by a combination of standard sampling and exact marginalization using the scale mixture representation of the heavy-tailed distributions presented in Section 2.1.

Inference in SLIM, CSLIM and SNIM is almost entirely done using Gibbs sampling. There are only two exceptions that we address using M-H updates: (i) the permutation matrix $\mathbf{P}$ for the ordering of the variables in DAGs and (ii) the length scales for GP based hierarchies. The cost of running the linear DAGs with latent variables or the factor model in SLIM is roughly the same, i.e. $\mathcal{O}(N_s d^2 N)$ where N_s is the total number of samples including the burn-in period. The memory requirements on the other hand are approximately $\mathcal{O}(N_p d^2)$ if all the samples after the burn-in period N_p are stored. This implies that the inference procedures scale reasonably well if N_s is kept in the lower ten thousands. SNIM is considerably more expensive due to the GP priors, hence the computational

[1]Intensive means not scaling with the sample size. Extensive means scaling with sample size in this case the size of the test sample.

cost rises up to $\mathcal{O}(N_s(d-1)N^3)$. All the remaining details of the models are thoroughly presented in Appendix A and C. PUB

Related Work

We can divide the related work in four categories, namely sparse factor models, linear and non-linear DAG structure learning models and general Bayesian networks. In fully Bayesian sparse factor modeling, two supergroups have been proposed: parametric models with bimodal sparsity promoting priors (West, 2003; Lucas et al., 2006; Carvalho et al., 2008), and non-parametric models where the number of factors is potentially infinite (Knowles and Ghahramani, 2007; Thibaux and Jordan, 2007; Rai and Daume III, 2009). It turns out that most of the parametric sparse factor models can be seen as finite versions of their non-parametric counterparts, for instance West (2003) and Knowles and Ghahramani (2007). The model proposed by West (2003) is, as far as we know, the first attempt to encode sparsity in a factor model explicitly in the form of a prior. The remaining models improve the initial setting by dealing with the optimal number of factors in Knowles and Ghahramani (2007), improved hierarchical specification of the sparsity prior in Lucas et al. (2006); Carvalho et al. (2008); Thibaux and Jordan (2007) and hierarchical structure for the loading matrices in Rai and Daume III (2009). Among the most representative methods for linear DAG learning we have, the Max-Min Hill-Climbing (MMHC) algorithm (Tsamardinos et al., 2006) that first learns the skeleton of the DAG using conditional independence tests in a similar fashion to Prototypical Constraint (PC) algorithms (Spirtes et al., 2001), then the order of the variables is found using a Bayesian-scoring hill-climbing search. The Sparse Candidate (SC) algorithm (Friedman et al., 1999) is in the same spirit but restricts the skeleton to within a predetermined link candidate set of bounded size for each variable. The Order Search (OS) algorithm (Teyssier and Koller, 2005) uses hill-climbing first to find the ordering, and then looks for the skeleton with SC. L_1 regularized Markov Blanket (Schmidt et al., 2007) replaces the skeleton learning from MMHC with a dependency network (Heckerman et al., 2000) written as a set of local conditional distributions represented as regularized linear regressors. The closest model to our linear DAG model is easily the Linear Non-Gaussian Acyclic Model (LiNGAM). Shimizu et al. (2006) provided the important insight that every DAG has a factor model representation, i.e. the connectivity matrix of a DAG gives rise to a triangular mixing matrix in the factor model. They also developed and algorithm based on Independent Component Analysis (ICA) (Hyvärinen et al., 2001) and iterative pruning to learn DAG structures. For General Bayesian networks, Hoyer et al. (2008) extended LiNGAM to allow for latent variables and Silva (2010) introduced a general Gaussian model that also allows for connectivity between latent variables. Finally, the non-linear

DAG models are mostly inspired by Friedman and Nachman (2000) and mainly consist of multiple regression models with independence test based pruning and exhaustive enumeration of variable permutations, see for instance Hoyer et al. (2009) and Zhang and Hyvärinen (2010). When enumeration is not a possibility because the number of variables is large, non-linear extensions to greedy approaches like DAG-search (see "ideal parent" algorithm, Elidan et al., 2007) or PC (see kPC, Tillman et al., 2009) can be used as a computationally affordable alternative.

Bottom line

We proposed a framework for sparse identifiable multivariate modeling in which we can learn models from a rather general class of Bayesian networks and perform quantitative model comparison between them. Model comparison may be used for model selection or serve as a hypothesis-generating tool. We use the likelihood on a test set as a computationally simple quantitative proxy for model comparison and as an alternative to the marginal likelihood. The other key ingredients in the framework are the use of sparsity, identifiable model components and the stochastic search for the correct order of the variables needed by the DAG representation. The slab and spike prior implicitly defines a prior over graph structures and it is thus a computationally attractive alternative to combinatorial structure search since parameter and structure inference can be performed simultaneously. Non-gaussianity of the latent variables is the starting point to guarantee identifiability. The equivalence between factor models and DAG allows to design a stochastic search procedure based on M-H updates to look for permutation matrices that make the connectivity matrix DAG representation strictly lower triangular (in a probabilistic sense). Our two extensions based on GP priors increase the applicability range of SLIM to problems with correlated observations and possible non-linear interactions. Inference is also important because for intractable models like ours, mixing and computational complexity could be a problem. We use scale mixture of Gaussian representations in order to turn parameter inference of non-Gaussian distributions into efficient Gibbs sampling schemes.

3.1.2 Latent Protein Trees

In proteomics data analysis, we are given a dataset containing quantitative expression levels of a number the isotope groups, each of them representing peptides that are stretches of a usually larger proteins of interest. Our main goal here is to estimate protein expression levels from a set of peptide-level

expression measurements. There are several challenges associated to this task: (i) We expect each peptide to be linked to a single protein, the problem is however that given a sample it is not always possible to identify the peptide nor its protein of origin (see Nesvizhskii, 2010, for a comprehensive review on protein identification). (iii) Some peptides could be miss-annotated or post-translationally modified, therefore they may not be representative of the protein expression profile they are associated with. (ii) The analysis is often times done in batches, thus just by looking at the data it is evident that there is a very strong batch effect in the data that needs to be corrected. (iii) There are multiple sources of systematic effects interfering with the protein expression variability we are finally interested in, these effects being likely to arise from technical rather than biological variability. (iv) There are plenty of missing values mainly due to limitations of the Liquid Chromatography and Mass Spectrometry (LC-MS) technology used. Besides these missing values are not evenly distributed in the data making the process of dealing with them even more difficult.

Taking into account all the previous insights, we decided to model the variability of N samples of d isotope groups distributed in N_B batches as a linear combination of four different contributions, batch effect, systematic variability, protein expression and noise as follows

$$\mathbf{x}_n^m = \boldsymbol{\mu}^m + \mathbf{A}\mathbf{z}_n + \mathbf{B}\mathbf{w}_n + \boldsymbol{\epsilon}_n \ , \tag{3.3}$$

where n is the sample index, m is the batch index and we also have the following variables in the right hand side of the equation above:

- $\boldsymbol{\mu}^m$ is the mean batch effect due to batch m.
- $\mathbf{z}_n$ is a vector of N_F latent factors meant to capture systematic variability present in the data.
- $\mathbf{w}_n$ is the expression level of a collection of N_P proteins.
- $\mathbf{A}$ is a $d \times N_F$ weight matrix containing the weights for the systematic effects factors.
- $\mathbf{B}$ is a $d \times N_P$ weight matrix containing the protein expression weights. Since we expect each isotope group to be associated to a single protein, each row of $\mathbf{B}$ contains a single non-zero element. Because of the possibility of miss-annotated peptides, such a non-zero element is not fixed but can be reassigned during inference.
- $\boldsymbol{\epsilon}$ is measurement uncorrelated noise.

The model in equation (3.3) called here LPT can be seen as an augmented factor model with three levels of resolution, the coarse level composed by the batch

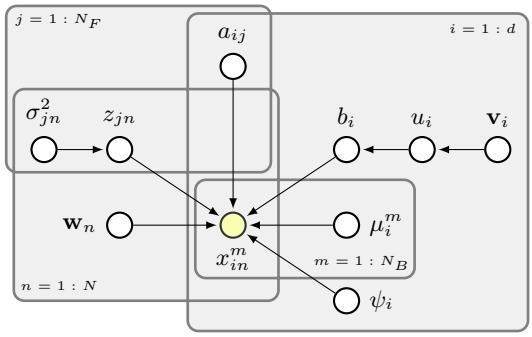


Figure 3.3: Graphical model for LPT.

means and the measurement noise, the middle level by the systematic effects and the detail level entirely dedicated to model the more specific protein expression levels. The name latent protein comes from the fact that actual proteins are not being observed. Although we count with is a set of associations between isotope groups and proteins we can use to infer and label protein expression levels. Such associations that we call in the following *annotation*, are obtained using an implementation of PeptideProphet algorithm (Keller et al., 2002).

The complete hierarchical model for equation (3.3) is shown in Figure 3.3 and can be written as

$$
\begin{aligned}
\mathbf{x}_n^m|\mu_m, \mathbf{A}, \mathbf{z}_n, \mathbf{B}, \mathbf{w}_n, \boldsymbol{\Psi} &\sim \mathcal{N}(\mathbf{x}_n^m|\boldsymbol{\mu}^m + \mathbf{A}\mathbf{z}_n + \mathbf{B}\mathbf{w}_n, \boldsymbol{\Psi}) \ , \\
\psi_i^{-1}|t_s, t_r &\sim \text{Gamma}(\psi_i^{-1}|s_s, s_r) \ , \\
\mu_i^m|t_m, t_p &\sim \mathcal{N}(\mu_i^m|t_m, t_p^{-1}) \ , \\
a_{ij} &\sim \mathcal{N}(a_{ij}|0, 1) \ , \\
z_{jn}|\sigma^2_{jn} &\sim \mathcal{N}(z_{jn}|0, \sigma^2_{jn}) \ , \\
\sigma^2_{jn}|\lambda &\sim \text{Exponential}(\sigma^2_{jn}|\lambda^2) \ , \\
\lambda^2|\ell_s, \ell_r &\sim \text{Gamma}(\lambda^2|\ell_s, \ell_r) \ , \\
b_{i,u_i}|u_i &\sim \mathcal{N}(b_{i,u_i}|0, 1) \ , \\
u_i|\mathbf{v}_i &\sim \text{Discrete}(\mathbf{v}_i) \ , \\
\mathbf{v}_i|\boldsymbol{\rho}_i, \boldsymbol{\kappa} &\sim \text{Dirichlet}(\boldsymbol{\rho}_i + \boldsymbol{\kappa}) \ , \\
\mathbf{w}_1, \ldots, \mathbf{w}_N &\sim \text{Coalescent}(\cdot) \ ,
\end{aligned}
$$

where $\boldsymbol{\epsilon}$ in equation (3.3) has been marginalized out using $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \boldsymbol{\Psi})$, $\boldsymbol{\Psi}$ is a diagonal matrix with elements ψ_i, μ_i^m is an element of $\boldsymbol{\mu}^m$ and we have omitted the hyperparameters in the graphical model. The systematic factors have independent Laplace distributions represented as scale mixtures of Gaussians with

gamma hyperprior on λ^2, see Section 2.1. The non-zero elements of $\mathbf{B}$ specified by the auxiliary variable $\mathbf{u} = [u_1, \ldots, u_d]$ are provided with standardized Gaussian distributions. Each of the elements of $\mathbf{u}$ takes a value between 1 and N_P with probability $\mathbf{v}_i = [v_1, \ldots, v_{N_P}]$. The latter has a Dirichlet prior with hyperparameters set directly from the annotation provided with the data, i.e. κ_k is the total number of isotope groups associated to protein k and ρ_{ik} enforces annotation. We achieve this by letting ρ_{ik} to have a large value if isotope i comes from protein k and zero otherwise. If an isotope group happens to be unannotated, then $\boldsymbol{\rho} = \mathbf{0}$. This setting allows the model to prefer annotated and abundant proteins for which we can most likely obtain plausible profiles. Conversely, it will certainly discourage rare proteins with say only one or two isotope groups associated to it.

The hierarchical model is completed by placing a coalescent prior denoted as Coalescent$(\cdot)$ on the N_P latent proteins conforming $\mathbf{W} = [\mathbf{w}_1 \ \ldots \ \mathbf{w}_N]$, see Section 2.5. With this prior we can both model correlations and have an interpretable representation of isotope groups, latent proteins and their interactions. Figure 3.4 shows the concept for a particular problem with $p = 15$ isotope groups distributed in $N_P = 5$ proteins, where K_k is the subset of isotope groups associated to latent protein k, thus $K_k \subset \{x_1, \ldots, x_d\}$. We can see a hierarchical clustering type of structure in which for instance w_4 and w_5 are more similar than w_3 and w_4, thus more correlated. The *pseudo time* t serves as similarity measure so that more alike proteins merge sooner in time, allowing us to directly quantify their pairwise or group-wise distances. The proposed hierarchy also reflects the fact that isotope groups and latent proteins must lay in different levels and that parent proteins are proxies for the average profile of groups of proteins.

Inference for the coalescent is done using SMC with resampling and the remaining components of the model through standard Gibbs sampling. All the remaining details of the model including relevant conditional posteriors and summaries are presented in Appendix E. PUB

Related work

Latent Protein Trees is clearly related to a number of research works in Bayesian factor models. Here we only mention some work concerning the hierarchical nature of the model, most of them actually more concerned about hierarchical clustering than density modeling. Neal (2003) introduces the so called Dirichlet diffusion tree, a family of prior distributions over multivariate distributions for hierarchical density modeling and clustering. Heller and Ghahramani (2005) proposes a probabilistic method for hierarchical clustering. Although,

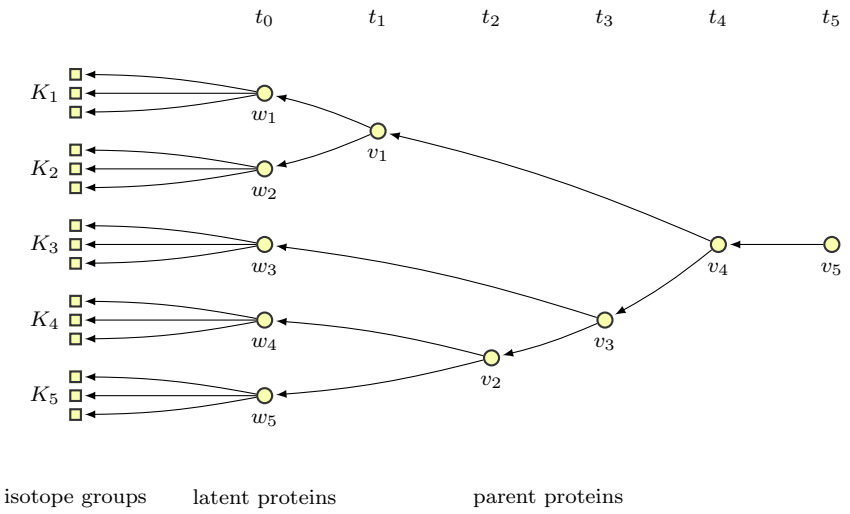


Figure 3.4: Latent protein tree structure. Particular tree with $N_P = 5$ and three isotope groups assigned to each latent protein. The pseudo time variable t denotes the merging points.

the method uses a probabilistic interpretation to build the tree, it does not attempt to model the data density nor places a prior over tree structures. Blundell et al. (2010) extends the work of Heller and Ghahramani (2005) by means of allowing the tree to have an arbitrary branching structure in what they call *rose trees*. Teh et al. (2008) propose an agglomerative hierarchical clustering method based on coalescents. Rai and Daume III (2009) use the method of Teh et al. (2008) to provide the loading matrix of a factor model with a tree structure, however such a structure is only used for visualization and interpretation, meaning that it does not contribute to the density model produced by the factor model. Adams et al. (2010) introduces a model for hierarchical clustering based on nested stick-breaking processes. Its most interesting feature is that observed variables can live at any node of the tree and not only at the leaves as in our LPT model, see Figure 3.4.

There is some work in the literature for proteomics specific data analysis although more oriented towards differential protein expression. For example Daly et al. (2008) describe a mixed effects model for estimating protein level differential expression, however each protein is handled independently discarding any possibility for protein correlation. Karpievitch et al. (2009) presents a statistical model for protein-level abundance that accounts for missing values in the data well because the peptide is not available independent of its abundance or because its abundance is too low to be detected by the machine. Both Daly et al.

(2008) and Karpievitch et al. (2009) assume that the peptide-protein associations are correct and employ maximum-likelihood estimation, failing at properly quantifying the inherent uncertainty of LC-MS based data.

Bottom line

We have described a factor model specifically designed for proteomics data analysis. It successfully handles broad scale variability that is known to come from several sources of technical artifacts such as batch effects and isotope group specific noise, thus enabling us to estimate latent protein profiles that directly model biological signals. Our hierarchical representation of isotope groups, latent proteins and parent proteins provide us with detailed annotation uncertainty assessment, detection of possibly miss annotated isotope groups or post-translational modified proteins and clustering of proteins with similar expression profiles that hopefully reflect biologically related interaction mechanisms. As we will show in Appendix E, some features of our model can be used to define predictive models based either on latent proteins or groups of latent proteins.

3.2 Supervised Modeling

In supervised modeling our interest is to use a set of covariates to classify a set of samples into a set of prefixed categories. Besides, we want to be able to perform predictions for new uncategorized samples, i.e. our model must be able to generalize the classification rule. Sparsity plays a very important role in this setting from two somewhat different perspectives. The first of them is focused towards answering the question of which subset of covariates is most likely to be involved when performing the classification task. This task is usually known in statistics and machine learning as the variable or feature selection problem. The second perspective has to do with sample selection, i.e. what is the most representative subset of the data that can be used during inference to build the classification model without sacrificing too much predictive power. Such a setting is particularly useful when the dataset is large and/or the asymptotical computational complexity of the model is heavily dependent on the number of samples, turning inference prohibitive. In this section, we present two approaches for sample selection in non-linear classifiers that we call Predictive Active Set Selection for Gaussian Processes (PASS-GP) and Fixed Predictive Active Set Selection for Gaussian Processes (fPASS-GP). These methods iteratively build sample subsets (active sets) based on the predictive distributions of a GPC model. The main difference between PASS-GP and fPASS-GP is that

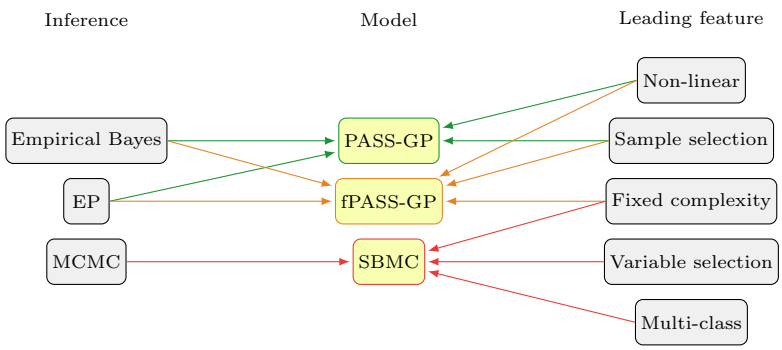


Figure 3.5: Supervised model's summary.

the latter has fixed computational complexity because the size of the active set can be set beforehand. In terms of inference, both approaches use Expectation Propagation (EP) to approximate the posterior of the GP and the marginal likelihood, and empirical Bayes for hyperparameter selection. Finally, we introduce a linear model for variable selection in multiple category classification that we choose to call Sparse Bayesian Multi-class Classifier (SBMC) for simplicity.

We can see the models presented in this section as two extreme cases in classification: (i) when the number of observations is large enough to render the model impractical and (ii) when the number of variables overwhelms the number of observations. The first scenario is common in data mining and machine learning applications, where data collection is relatively inexpensive and the effort required to label the samples is usually in the same order of the data costs. For instance natural images classification/segmentation, in which very large databases can be easily created and curated with the help of internet and cheaply labeled using online tools like LabelMe (`http://labelme.csail.mit.edu/`) or Google's own image labeler (`http://images.google.com/imagelabeler/`). The second scenario covers a vast number of studies in biology, where data collection far more expensive than the labeling due to the technology to be used or just because data is difficult to obtain/measure. In gene expression for example, although large amounts of genes can be measured at the same time, obtaining and preparing samples from patients/individuals is not only expensive but time consuming, thus reducing the chances of building large datasets.

Figure 3.5 shows a brief summary of the three models including their leading features and the inference technique to be used. The remainder of this section is based on two publications presented in Appendices B, D and one work in progress fully described in Appendix F.

PUB

3.2.1 Predictive Active Set Selection Methods for GPC

Gaussian Process Classification is a very attractive non-parametric method for supervised learning since it is conceptually simple, flexible and fully probabilistic. On the downside, its computational complexity scales cubically with the number of samples $\mathcal{O}(N^3)$ and quadratically in the memory requirements $\mathcal{O}(N^2)$. Such an issue is becoming a very critical limitation with the increasing availability of larger and larger datasets. There are basically two ways in which a large dataset can be handled by a GP classifier. One of them is to find a way to select a subset of the data or active set and then build the GP classifier in the standard way, (see Seeger, 2003; Lawrence et al., 2003, 2005; Henao and Winther, 2010). The alternative is to approximate the GPC using a virtual set of samples that need to be inferred, often called pseudo-inputs (see Rasmussen and Williams, 2006; Quiñonero-Candela and Rasmussen, 2005; Naish-Guzman and Holden, 2008). The approaches we present here — PASS-GP and fPASS-GP, consist of an active set selection method in which the selection criterion is based on the (dual representation) weight of the data points. In GPC this is the same as using the predictive distribution, i.e. include data points with small predictive probability. For points in the active set we can compute the LOO (or cavity) predictive probability and use that for deletions. We alternate between active set updates and hyperparameter optimization based upon the marginal likelihood of the active set, i.e. empirical Bayes. It is clear that restricting ourselves to an active set is less elegant than approximating the posterior distribution over a set of pseudo-inputs. However, currently this appears necessary for large data sets like MNIST.

As described in Section 2.4, GPC amounts to compute the marginal $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\upsilon})$, for a dataset $\{\mathbf{X}, \mathbf{y}\}$ and an optimized set of hyperparameters $\boldsymbol{\upsilon}$. From now on we assume without loss of generality that the GP prior $p(\mathbf{f}|\mathbf{X}, \boldsymbol{\upsilon})$ has zero mean function thus $\boldsymbol{\rho}$ as shown in equation (2.13) is no longer required. The likelihood $p(\mathbf{y}|\mathbf{f}, \mathbf{X})$ is not Gaussian because $\mathbf{y}$ is a binary variable, in particular we use a probit link. We can write the posterior distribution as

$$p(\mathbf{f}|\mathbf{X}, \mathbf{y}, \boldsymbol{\upsilon}) = Z^{-1} p(\mathbf{f}|\mathbf{X}, \boldsymbol{\upsilon}) \prod_{n=1}^{N} t(y_n|f_n) \ , \tag{3.4}$$

where $Z = p(\mathbf{y}|\mathbf{X}, \boldsymbol{\upsilon})$ is the marginal likelihood and $t(y_n|f_n) = \Phi(f_n y_n)$ is the cumulative Gaussian function. From equations (2.15) and (3.4) is easy to see that both distributions are intractable, thus we must resort to approximations. We use EP because it has been shown to be the most accurate deterministic approximation available (see Quiñonero-Candela and Rasmussen, 2005; Kuss and Rasmussen, 2005, 2006). The idea behind our active set selection methods is to compute $p(\mathbf{f}_A|\mathbf{X}_A, \mathbf{y}_A, \boldsymbol{\upsilon})$ for an active set A and then approximate $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\upsilon})$.

Algorithm 1: Predictive active set selection based algorithm.

Input : $\{\mathbf{X}, \mathbf{y}\}$, $\boldsymbol{\upsilon}$, N_{init} and N_{maxit}
Output: $q(\mathbf{f}_A|\mathbf{X}_A, \mathbf{y}_A, \boldsymbol{\upsilon})$, $\boldsymbol{\upsilon}_{\text{new}}$ and A
begin
  $A \leftarrow \{1, \ldots, N_{\text{init}}\}$
  for $i = 1$ **to** N_{maxit} **do**
    $\boldsymbol{\upsilon}_{\text{new}} = \text{argmax}_{\boldsymbol{\upsilon}} \log q(\mathbf{y}_A|\mathbf{X}_A, \boldsymbol{\upsilon})$
    Get $q(\mathbf{f}_A|\mathbf{X}_A, \mathbf{y}_A, \boldsymbol{\upsilon})$, $q(y^*|\mathbf{X}_A, \mathbf{y}_A, \mathbf{x}^\star, \boldsymbol{\upsilon})$ and $q_{\backslash n}(y_n|\mathbf{X}_A, \mathbf{y}_{A\backslash n})$
    forall the $\{\mathbf{x}_n, y_n\} \in \{\mathbf{X}_A, \mathbf{y}_A\}$ **do**
      if RemoveRule($q_{\backslash n}(y_n|\mathbf{X}_A, \mathbf{y}_{A\backslash n})$) **then** $A \leftarrow A\backslash\{n\}$
    end
    forall the $\{\mathbf{x}_n^\star, y_n^\star\} \in \{\mathbf{X}_I, \mathbf{y}_I\}$ **do**
      if AdditionRule($q(y^*|\mathbf{X}_A, \mathbf{y}_A, \mathbf{x}^\star, \boldsymbol{\upsilon})$) **then** $A \leftarrow A \cup \{n\}$
    end
  end
end

To build the active set we start from some randomly selected initial active set of size N_{init} and proceed by iteratively adding and removing samples using the predictive distribution

$$q(y^*|\mathbf{X}_A, \mathbf{y}_A, \mathbf{x}^\star, \boldsymbol{\upsilon}) = \int t(y^\star|f^\star) q(f^\star|\mathbf{X}_A, \mathbf{y}_A, \mathbf{x}^\star, \boldsymbol{\upsilon}) df^\star \ ,$$

and the cavity predictive distribution

$$q_{\backslash n}(y_n|\mathbf{X}_A, \mathbf{y}_{A\backslash n}) = \int t(y_n|f_n) q_{\backslash n}(\mathbf{f}_A|\mathbf{X}_A, \mathbf{y}_{A\backslash n}, \boldsymbol{\upsilon}) d\mathbf{f} \ ,$$

where $q_{\backslash n}(\mathbf{f}_A|\mathbf{X}_A, \mathbf{y}_{A\backslash n}, \boldsymbol{\upsilon})$ is the so called cavity distribution, $q(\cdot)$ denotes the approximation obtained by EP and $\{\mathbf{x}_n^\star, y_n^\star\} \in \{\mathbf{X}_I, \mathbf{y}_I\}$, the inactive set. The algorithm proceeds by iterating through three steps: (i) update active set, (ii) recompute the EP approximations and (iii) optimize the hyperparameters, as depicted in Algorithm 1 (N_{maxit} is the maximum number of iterations allowed).

The most significant difference between PASS-GP and fPASS-GP is that in the former the addition and deletion rules for updating the active set are based on a threshold on $q(y^\star|\mathbf{X}_A, \mathbf{y}_A, \mathbf{x}^\star, \boldsymbol{\upsilon})$ and $q_{\backslash n}(y_n|\mathbf{X}_A, \mathbf{y}_{A\backslash n}, \boldsymbol{\upsilon})$, such that the resulting size of the active set cannot be determined before hand. When the computational resources are limited, we might want to set the size of the active set beforehand. In fPASS-GP, instead of thresholding the distributions we add and remove equal amounts of points from the active set so to keep its size constant and hence the overall computational cost. Appendices B and D show all the details about EP based inference, the derived marginal likelihood approximations and the active set update rules for each approach.

Related Work

The most simplistic way to handle the problem of scalability in a GPC is to approximate the covariance matrix of the GP prior using the Nyström method as done by Williams and Seeger (2001), nevertheless as pointed out by Quiñonero-Candela and Rasmussen (2005) such an approximation does not correspond to a well-defined probabilistic model. The Informative Vector Machine (IVM) is perhaps the closest relative to PASS-GP and fPASS-GP. It is a very fast algorithm since its active set updates are as cheap as $\mathcal{O}(1)$, however it is not well suited for integrated model selection since only one point is considered at the time for inclusion in the active set (Lawrence et al., 2003; Seeger, 2003; Lawrence et al., 2005). Besides it only grows the active set letting point deletions out of question, thus leading to unnecessarily large active sets. Naish-Guzman and Holden (2008) introduced the first pseudo-input method for GPC. Its main advantage over active set methods is that the pseudo-inputs and the hyperparameters can be optimized jointly. However, since the number of parameters to be estimated grows with the number and dimension of the pseudo-inputs, this method becomes computationally unfeasible for large datasets and may even suffer from overfitting due to the number of free parameters in the model. See Quiñonero-Candela and Rasmussen (2005) for a complete review on pseudo-input methods, and Keerthi et al. (2006) and Joachims and Yu (2009) for similar approaches designed specifically for Support Vector Machines (SVMs).

Bottom line

We have proposed a framework for active set selection in GPC. The core of our active set update rule is that the predictive distribution of a GP classifier can be used to quantify the relative weight of points in the active set that can be marked for deletion or new points from the inactive set with low predictive probabilities, that make them ideal for inclusion. The algorithmic skeleton of our framework consists on two alternating steps, namely active set updates and hyperparameter optimization. We designed two active set update criteria that target two different practical scenarios. The first we called PASS-GP focuses on interpretability of the parameters of the update rule by thresholding the predictive distributions of GPC. The second acknowledges that in some applications having a fixed computational cost is key, thus fPASS-GP allows to keep the size of the active set fixed so the overall cost and memory requirements can be known beforehand.

3.2.2 Sparse Bayesian Multi-class Classifier

Classification models for the extreme ill-posed case of many more covariates than examples, sometimes referred to as "large p, small n" (West, 2003), has attracted much recent attention in bioinformatics, especially in the context of gene expression profiling (Statnikov et al., 2005). Current practice often involves a combination of supervised and unsupervised single covariate filtering prior to classification. Genes with a deviation of the intensity small or large relative to the mean may be excluded and then univariate t- and F-tests may be used to further reduce the number of input features (Dudoit and van der Laan, 2008). Using univariate supervised techniques as t- and F-tests, may be in general sub-optimal as it can miss important features when the separation of the classes is not aligned with expression of the single genes but rather linear combinations of them. Also, when there is co-variation between the genes then univariate tests give a misleading picture of their significance. The motivation for working with a reduced set is both computational and predictive, i.e. working directly with a high number of non-informative covariates increases the risk for overfitting for most standard classification techniques. Therefore, two step covariate selection/classification procedures with nested validation loops (e.g. using LOO cross-validation) must be used in order to obtain reliable yet unbiased results (Statnikov et al., 2005). The model presented here is a fully Bayesian approach to the problem of multi-category linear classification for the ill-posed setting. The key ingredient we want to capture in the model is sparsity, i.e. that only a small fraction of the genes (probes) can be expected to give discriminatory information.

Suppose we have a set of N independent observations $\mathbf{x}_1, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N$, where $\mathbf{x}_n$ is a vector with d covariates assigned to one of C different classes. The class labels y_n are assumed to have a discrete distributions with parameters $p_{1n}, \ldots, p_{cn}, \ldots, p_{Cn}$. We define stochastic regression functions $f_{cn} = f_c(\mathbf{x}_n)$ for all classes $c = 1, \ldots, C$ and examples $n = 1, \ldots, N$, similar to Albert and Chib (1993). Here we will focus on model on the form $f_c(\mathbf{x}_n) = h_c(\mathbf{x}_n) + \epsilon_{cn}$. The independently distributed additive term ϵ_{cn} encapsulates the link function of the model. For instance, zero mean unit variance Gaussian leads to the probit link, however a more general class of link functions based upon Student's t-distributed errors is also considered. Here we focus on the linear model $h_c(\mathbf{x}_n) = \mathbf{w}_c^\top \mathbf{x}_n$, where $\mathbf{w}_c$ is a weight vector for class c, for which efficient Gibbs sampling based inference can be implemented. Although, in principle this function could be made non-linear by using for examples Gaussian process priors (see Girolami and Rogers, 2006; Liang et al., 2005-09).

In our model, the likelihood function links the regressor with the probabilistic model for the output labeling. The simplest model assigns probability one to

class c when f_{cn} is larger than the other f_{jn} for $j \neq c$, so

$$p_{cn} = p(y_n = c|\mathbf{f}_n) = \prod_{j \neq c} \Theta(f_{cn} - f_{jn}) \ .$$

Here $\mathbf{f}_n = [f_1, \ldots, f_C]^\top$ and $\Theta(\cdot)$ is the Heaviside step-function. From the expression above we can see directly that only differences between regression functions are identifiable in the model. This will in some cases play a role for the interpretation of the inferred model parameters. When $p_{\text{link}}(\epsilon_{cn}|\cdot)$ is Gaussian, i.e. $\epsilon_{cn} \sim \mathcal{N}(\epsilon_{cn}|0,1)$, we may marginalize over f, $\mathbf{f} \sim \mathcal{N}(\mathbf{f}|\mathbf{h}, \mathbf{I})$, to obtain a soft decision boundary model expressed in terms of $\mathbf{h} = [h_1(\mathbf{x}), \ldots, h_C(\mathbf{x})]^\top$ and the parameters of the link distribution as

$$\begin{aligned} p(y = c|\mathbf{h}, \cdot) = &\int \prod_{j \neq c} \Theta(f_c - f_j) \prod_k p_{\text{link}}(f_k|h_k(\mathbf{x}), \cdot) d\mathbf{f} \\ &\int \prod_{j \neq c} \Phi(f_c - h_c(\mathbf{x})) \mathcal{N}(f_c|h_c(\mathbf{x}), 1) df_c \ , \end{aligned} \tag{3.5}$$

where $\Phi(\cdot)$ is the probit link (cumulative Gaussian) function. For the particular case of $C = 2$ this formulation reduces to standard probit regression since $p(y_n = 1|h(\mathbf{x}_n)) = \Phi(h(\mathbf{x})))$ with $h(\mathbf{x}) = (h_1(\mathbf{x}) - h_2(\mathbf{x}))/\sqrt{2}$ and in the linear model $h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ and $\mathbf{w} = (\mathbf{w}_1 - \mathbf{w}_2)/\sqrt{2}$.

We can use the scale mixture of Gaussian representation from Section 2.1 to easily provide ϵ_{cn} with a more general class of distributions, namely the t family, while still being able to sample from the posterior distribution in an efficient way. For instance, the t link contains as special cases, probit when $\sigma^2 = 1$ and $\theta \to \infty$, Cauchy when $\sigma^2 = \theta = 1$, and it is known to be a good approximation of the logit link when $\sigma^2 = 0.401$ and $\theta = 8$ (Albert and Chib, 1993) or similarly $\sigma^2 = 0.413$ and $\theta = 7.581$ obtained in (Chen and Dey, 1998).

The use of sparse models is supported by the assumption that the observed data contains irrelevant covariates, i.e. there is a number of covariates $d' < d$ such that the model with d' (relevant) covariates is at least equally supported by the data as the full model. This is specially true for instance in gene expression classification where the expected number of genes involved in a particular condition (class) is known to be small compared to the size of a commercial microarray. Sparsity is also motivated by the need to control the complexity in the data poor regime $N \ll d$, meaning that we simply have too little data to learn the model in all its complexity but we can learn some useful features of it from limited data. The ideal complexity of a model is thus closely related with the number of observations used to fit its parameters, which in principle means that we cannot expect to use more covariates than observations if we want to prevent overfitting unless we regularize the model in some appropriate

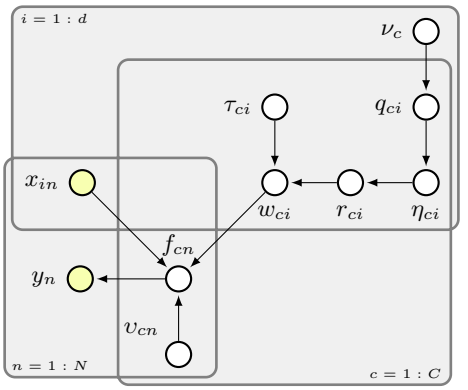


Figure 3.6: Graphical model for SBMC.

way. One possible way of restricting the model is to turn off some parameters of the model, which in the linear case correspond to variable selection. Here we use the two layer slab and spike prior from Section 2.2 on the weight matrix $\mathbf{W} = [\mathbf{w}_1 \ \ldots \ \mathbf{w}_C]^\top$. Now we can take equations (3.5), (2.10) and (2.12) to build the hierarchical model

$$
\begin{aligned}
y_n = c | f_{cn}, \mathbf{w}, \mathbf{x}_n &= \prod_{j \neq c} \Phi(f_{cn} - \mathbf{w}_j^\top \mathbf{x}_n) \\
f_{cn} | \mathbf{w}_c, \mathbf{x}_n, \upsilon_{cn}, \sigma^2 &\sim \mathcal{N}(f_{cn} | \mathbf{w}_c^\top \mathbf{x}_n, \upsilon_{cn} \sigma^2) \ , \\
\upsilon_{cn}^{-1} | \theta &\sim \text{Gamma}\left(\upsilon_{cn} \middle| \tfrac{\theta}{2}, \tfrac{\theta}{2}\right) \ , \\
w_{ci} | r_{ci}, \tau_{ci} &\sim (1 - r_{ci}) \delta(w_{ci}) + r_{ci} \mathcal{N}(w_{ci} | 0, 1) \ , \\
r_{ci} | \eta_{ci} &\sim \text{Bernoulli}(r_{ci} | \eta_{ci}) \ , \\
\eta_{ci} | q_{ci}, \alpha_p, \alpha_m &\sim (1 - q_{ci}) \delta(\eta_{ci}) + q_{ci} \text{Beta}(\eta_{ci} | \alpha_p \alpha_m, \alpha_p (1 - \alpha_m)) \ , \\
q_{ci} | \nu_c &\sim \text{Bernoulli}(q_{ci} | \nu_c) \ , \\
\nu_c | \beta_p, \beta_v &\sim \text{Beta}(\nu_c | \beta_p \beta_m, \beta_p (1 - \beta_m)) \ ,
\end{aligned}
$$

where we can identify the probit link after marginalizing ϵ_{cn}, the scale mixture of Gaussians for the Student's t distribution and the two layer slab and spike hierarchy for the weight vectors. The graphical model shown in Figure 3.6 has two observed nodes $\mathbf{X}$ and $\mathbf{y}$ corresponding to the covariates and class labels respectively. The latent variables $\mathbf{F}$ are connected to the variances υ_{cn} of the t link and the hierarchical prior for $\mathbf{W}$. The slab and spike prior consist on a binary variable r_{ci} indicating whether w_{ci} is non-zero with mean probability η_{ci} and a shared sparsity parameter ν_c with indicator variable q_{ci}. The non-zero elements in $\mathbf{W}$ are independently Gaussian distributed with unit variance.

To make predictions we need to compute a predictive distribution of the form

$$p(y^\star = c|\mathbf{x}^\star, \mathbf{y}, \mathbf{X}, \cdot) = \int p(y^\star = c|f^\star)p(f^\star|\mathbf{W}, \upsilon^\star, \mathbf{x}^\star)p(\mathbf{W}, \mathbf{U}|\mathbf{y}, \mathbf{X})df^\star d\upsilon^\star d\mathbf{W} d\mathbf{U} \ , \tag{3.6}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ and $\mathbf{y} = [y_1, \ldots, y_N]$ are respectively observations and labels used during inference, $\mathbf{x}^\star$ is test observation and $y^\star$ its predicted class. The distributions involved in equation (3.6) are, the posterior of the weight matrix $\mathbf{W}$, the posterior of $f^\star$ in equation (2.10), the variances of the link function $\mathbf{U} = [\upsilon_1, \ldots, \upsilon_N]$ and $\upsilon^\star$. It turns out that even when the integral above is intractable, is not difficult to sample from the posterior distribution.

See Appendix F for more details about inference, hyperparameter settings and posterior summaries and some insights on how to improve the model in terms of identifiability and interpretation. PUB

Related Work

The most relevant related work to the model presented above include sparse multinomial logistic regression using Laplace distributions to achieve sparsity as in Krishnapuram et al. (2005) and Cawley et al. (2007). Since the only way to get sparsity using Laplace distributions without using thresholding is to perform MAP inference, it is hard to asses the uncertainty of the selected variables or ranking them using a criterion other than the magnitude associated to the their weights. Girolami and Rogers (2006) proposed a multi-category GPC in which variable selection is possible through Automatic Relevance Determination (ARD) covariance functions and thresholding. Claeskens et al. (2008) introduce a new information criterion for variable selection using SVMs and also offers a review of existing methods in the same spirit. These approaches are also point estimates as MAP so they have the same issues associated with uncertainty. In addition, similar to the other parameters in the SVM, the level of sparsity must be selected by cross-validation, making the computational cost an issue and the interpretation more difficult. Here we are limiting ourselves to linear models because inference is much easier in this case provided that standard Gibbs sampling can be used. The straightforward extension to non-linearity can be done using GPCs (sparse but binary, classifier Liang et al., 2005-09) or Dirichlet process mixtures (multi-category but not sparse, Shahbaba and Neal, 2009). However, the need for M-H sampling of the kernel parameters (with slab and spike prior) makes inference much more complicated in this case. Related fully Bayesian approaches for the binary classification scenario include Bae and Mallick (2004) using Laplace priors, and Lee et al. (2003), Zhou et al.

(2004a) and Hernandez-Lobato et al. (2010) use single layer slab and spike priors as defined by George and McCulloch (1993, 1997). For multi-category classification, we are aware of two approaches also using single layer spike and slab priors (Zhou et al., 2004b; Sha et al., 2004). The main difference between the latter two and our approach is the more elaborated sparsity prior and that we compute proper predictive distributions to obtain fully probabilistic outputs.

Bottom line

We proposed a fully Bayesian approach to the problem of multi-category classification for the ill-posed setting. The key ingredient we want to capture in the model is sparsity, i.e. that only a small fraction of the genes (probes) can be expected to give discriminatory information. We explicitly model this by having a prior over the weights that put a large point mass at zero, in the form of a slab and spike prior. In this way we consider a large number of potential hypotheses about the data by averaging. This procedure is not only computationally attractive but also avoids the need for multiple comparison correction arising in frequentist approaches to model selection. It also produce robust and interpretable results. We limit ourselves to linear models because inference is much easier as standard Gibbs sampling can be used.

Chapter 4

Applications

The applications presented in this chapter are a collection of case studies using real world datasets from a variety of fields in which we illustrate the capabilities of the models described in the previous chapter. Specifically, we consider flow cytometry, images, gene expression and label-free proteomics data. Here we are only attempting to describe the dataset, the model to be used and some highlights of the obtained results. We also point to the relevant appendix in which extensive experiments, analysis and discussions are included.

4.1 Protein Signaling Network

This case study demonstrates a typical application of sparse multivariate modeling in a realistic biological large N, small d setting. The dataset introduced by Sachs et al. (2005) consists of flow cytometry measurements of 11 phosphorylated proteins and phospholipids (raf, erk, p38, jnk, akt, mek, pka, pkc, pip_2, pip_3, plc). Each observation is a vector of quantitative amounts measured from single cells. Data was generated from a series of stimulatory cues and inhibitory interventions. Hence the data is composed of three kinds of perturbations: general activators, specific activators and specific inhibitors. In the experiments we only use the 1755 observations corresponding to general stimulatory conditions.

It is clear that using the whole dataset, i.e. using specific perturbations, will produce a richer model, however handling interventional data is out of the scope of this work mainly because handling that kind of data with a factor model is not well understood yet. Thus our order search prior to find the permutation matrix $\mathbf{P}$ for DAGs is not appropriate. Focused only on the observational data, we want to test all the possibilities of our multivariate models presented in section 3.1.1 in this dataset, namely, standard factor models, pure DAG, DAGs with latent variables, non-linear DAGs (SNIM) and quantitative model comparison using test likelihoods.

This dataset was initially analyzed using only factor models and DAGs and the results presented in Appendix A show that the factor model is better at explaining the data. This is reasonable since with observational data our DAG model is merely able to capture a substructure of the network widely accepted as the ground truth, as shown in Sachs et al. (2005, Figure 2 and Table 3). With the results obtained using the additional models mentioned above and presented in Appendix C we showed that not only the DAG with latent variables is better at explaining the data but the latent variables found resemble those also depicted in the ground truth. In addition, when comparing with other well known methods for DAGs modeling and our own non-linear DAG, we found that none of them produce results as good as our pure DAG model or our DAG augmented with latent variables.

4.2 USPS Dataset

The USPS digits database contains 9289 grayscale images of size 16×16 pixels, scaled and translated to fall within the range from -1 to 1. Here we are using the traditional data splitting, i.e. 7291 observations for training and the remaining 2007 for testing. For each binary one-against-rest digit classifier (10 in total) we use vectorized images as input and the same GPC model setup consisting on a squared exponential covariance matrix plus an additive noise component (jitter). This dataset is known to be challenging because each binary classification task is very unbalanced and the input data lies highly non-linear yet low dimensional manifold. Results shown in Appendix B and D show that our predictive active set GP classifiers are significantly better than other greedy methods available in the literature like online GP (Csató, 2002), the IVM (Lawrence et al., 2003) and the Reduced complexity SVM (RSVM) (Keerthi et al., 2006). Our classifiers turned out to be also slightly better than a full GP classifier with hyperparameter optimization and comparable with other state-of-the-art classifiers like SVM and k-Nearest Neighbor (KNN) clasifier. When comparing our active set selection methods against each other we found that PASS-GP provides a better

trade-off between accuracy, active set size and computational time.

4.3 MNIST Dataset

The MNIST digits database is similar to the USPS database however it constitutes a much harder task since it is approximately seven times larger, more specifically has 60000 and 10000 as training and testing observations respectively. Besides each image being now of size 28×28 pixels implies that the problem is in 784 dimensions as opposite to the smaller 256 dimensions of the USPS database. This dataset has been extensively studied in the machine learning community (visit `http://yann.lecun.com/exdb/mnist/`) thus the results obtained by a handful of different methods are very competitive. It even counts with an estimation of the human error rate (0.2%), which is in practice the figure to outperform. The most important result we obtained with this dataset is that as far as we know the experiments shown in Appendix B and D are the first GPC based results on MNIST using the whole database. Even if our accuracies are slightly lower than those obtained by state-of-the-art techniques such as SVMs, it is worthwhile to note than these methods use a number of support vectors or basis functions approximately twice as large as the active set sizes reported by our PASS-GP. We also considered the task known as *incorporating invariances* in the data to improve the overall classification results (DeCoste and Schölkopf, 2002). It consists on augmenting the dataset with slightly modified versions of each digit in the dataset. In total nine 1-pixel translations (up, down, left, right and diagonals) were included leading to an expanded dataset nine times larger than its original version. The results obtained with PASS-GP were as low as 0.86%, which is comparable to the 0.68% obtained by SVM.

4.4 Cause-effect Pairs

The cause-effect pairs database initially built for the NIPS 2008 causality competition (Mooij and Janzing, 2010), it contains observations of 51 pairs of real world variables obtained from different publicly available sources. The task in essentially simple, it consists on establishing which of the two variables is the cause an which is the effect. The task turns out to be very hard because these datasets are known for having very complex non-linear interactions and noise levels, rendering standard linear causality tests inappropriate. Nevertheless, it is perfectly suited to benchmark non-parametric/non-linear alternatives. The results obtained using SNIM along with other recently proposed state-of-the-art

techniques (Hoyer et al., 2009; Zhang and Hyvärinen, 2009; Daniusis et al., 2010) are shown in Appendix C and suggest that SNIM is comparable in performance with the most successful techniques tried.

4.5 E. Coli Dataset

The dataset introduced by Kao et al. (2004) consists of temporal gene expression profiles of *E. coli* samples during its transition from glucose to acetate measured using DNA microarrays. Samples from 100 genes were taken at 5, 10, 15, 30, 60 minutes and every hour until 6 hours after transition. The general goal is to reconstruct the unknown transcription factor activities from the expression data and some prior knowledge. The prior knowledge consists of taking the set of transcription factors (ArcA, CRP, CysB, FadR, FruR, GatR, IcIR, LeuO, Lrp, NarL, PhoB, PurB, RpoE, RpoS, TrpR and TyrR) controlling the observed genes and the (up-to-date) connectivity between genes and transcription factors from RegulonDB (Gama-Castro et al., 2008). It is well-known that the information in RegulonDB is still incomplete and hard to obtain for organisms different than *E. coli*. Our goal here is thus to obtain similar transcription factor activities to those found by Kao et al. (2004) without using the information from RegulonDB, but simply taking into account that the data at hand is a time series. In Appendix C we obtained results for two versions of our time seres model — CSLIM, one of them using the connectivity information found in RegulonDB and the other that runs unrestricted. The results suggest that there is no evidence of model preferences from the data point of view, meaning that data does not prefer the model with richer prior information.

4.6 H1N1/H3N2 Data

We start from proteomics data obtained from 43 patients part of the DARPA H1N1/H3N2 plasma project (Zaas et al., 2009). Each observation correspond to a sample from an individual at one of four different reference time points ($t = 0, 0.2, 0.8, 1$) assigned to one of two status groups, namely symptomatic or asymptomatic. H3N2 contains 76 samples (19 patients) obtained in two batches, the first of them containing all observations (42) from time points $t = 0$ and $t = 1$. The remaining 96 samples (24 patients) correspond to the H1N1 study. Alignment and annotation of the three batches available (H1N1, H3N2_1 and H3N2_1) was done using a combination of Mascot (`http://www.matrixscience.com/`), the PeptideProphet algorithm (Keller et al., 2002) and the statistical alignment model described in the supplementary material at the end of Appendix E.

From all available isotope groups, 13845 were found both in H1N1 and H3N2 data but only 4670 were provided with annotation. From the set of 4670 annotated isotope groups, only 1697 share the same annotation according to Mascot-PeptideProphet analysis. The remaining 2973 isotope groups consist of annotations transferred from H1N1 to H3N2 or vice versa, using the alignment model. The set of annotations itself, include 239 proteins from which 106 are assigned to more than a single isotope group. The data has relatively low amount of missing values. The inconvenient is that the 2% of missing values in the dataset are very unevenly distributed, in particular, $H3N2_1$ has 10.3% of them, $H3N2_2$ 0.7% and H1N1 up to 2.5%. We removed one sample because it had more than 30% missing values in the set of annotated isotope groups.

From experiments with LPT, described in full in Appendix E, we found that we can successfully subtract the evident systematic and batch effects from the estimated latent protein expression profiles. Annotation-wise, our model kept the original annotation of approximately 50% of the isotope groups. This is an indication that some of the isotope groups are miss annotated and some of the proteins post-translational modified, as we may expect from this kind of data. We also found that 3% of the isotope groups are not stable in terms of protein assignments thus can be regarded as highly noisy or of poor quality, thus should not be used to derive hypothesis. At a latent protein level, we encountered that 72% of them are representative of their annotation label, thus graded as *identified*. Finally, we found that a group of 5 proteins are discriminative of the symptomatic/asymptomatic status and that samples coming from H3N2 study are easier to classifier, particularly at latter stages of the disease.

Chapter 5

Conclusion

In this thesis we have presented a selection of statistical models for unsupervised and supervised modeling that explicitly impose sparsity as a way to achieve interpretability and/or computational affordability. We showed that sparsity in conjunction with other prior distributions can lead to very powerful tools for analysis of data under very structured yet realistic generative assumptions.

This thesis was written as compact and self-contained as possible. This is why we included in Chapter 2 a short description of all the priors used in our models. Our goal was to make easier to follow the hierarchies presented in Chapter 3 and the way they were parameterized. Of particular interest are the slab and spike priors to promote sparsity, the order search priors to infer the permutations needed to build DAG representations, Gaussian Process priors for nonparametric modeling of time series and non-linearities and the coalescent for hierarchical representations of sets of correlated variables. Chapter 3 presents four models divided into two categories, namely supervised and unsupervised modeling. (i) Sparse identifiable multivariate modeling is targeted to general linear Bayesian networks with Sparse Linear Identifiable Multivariate Modeling and extended to non-linear DAGs and correlated data through Sparse Non-linear Identifiable Multivariate Modeling and Correlated Sparse Linear Identifiable Multivariate Modeling, respectively. (ii) Latent Protein Tree is a specialized factor model for proteomics data analysis. It considers systematic and batch effects subtraction, isotope group-protein annotation assessment and a hierarchical representation

of the correlation structure of the data through a tree representation of latent proteins, parent proteins and isotope groups. (iii) Predictive active set selection brings affordability to GPC by using predictive distributions while keeping hyperparameter optimization as an important ingredient of the model. (iv) Finally, sparse Bayesian multi-category classification tackles the problem of variable selection in the more covariates than examples scenario. Each of the models was introduced and provided with motivation, hierarchical model, related work and its own conclusions. Chapter 4 presents a number of case studies used to illustrate the capabilities of our models through a series of experimental results presented and discussed in full in their respective appendices.

Open questions

Next we provide a list of unanswered questions that complement the work presented in this thesis and that we hope we can handle in a near future.

Interventional data. SLIM cannot handle experimental (interventional) data, and consequently around 80% of the data from the Sachs et al. (2005) study is not used. It is well-established how to learn with interventions in DAGs (see Sachs et al., 2005). The problem remains of how to formulate effective inference with interventional data for factor models.

Scalable order search priors. We do not have yet an ordering search procedure for the non-linear version of SLIM. This is a necessary step to take since exhaustive enumeration of all possible orderings is not an option beyond say 10 variables. The main problem is that the non-linear DAG has no equivalent factor model representation so we cannot directly use the permutation candidates as we do it with SLIM. However, as long as the non-linearities are weak, one might in principle use the permutation candidates found in a factor model, i.e. the linear effects will determine the correct ordering of the variables.

Unlabeled isotope groups and proteins. Including non annotated isotope groups in our current model for LPT has essentially two difficulties. The first is that if we run our model using all the data, we will accept that annotated and annotated isotope groups belong to the same set of proteins. This is unrealistic because such a set of proteins is taken entirely from the annotation. The second has to do with latent protein identification. Lets assume we can let the model decide upon the number of latent proteins, this will allow the model to accommodate proteins beyond the initial set created from the annotation. After inference, we will have to label latent proteins according to the concentration of its components, however this can be done only with annotated isotope groups for which we have protein assignments. Preliminary results using beta processes

suggest that the model becomes harder to interpret because we tent to end up with latent proteins in which the concentration of annotated proteins is too low or too heterogeneous to be able to label them.

Dropped isotope group annotations. Currently, annotation of proteomics data is done mainly with the PeptideProphet algorithm (Keller et al., 2002). The output of the algorithm is a set of associations between measured variables and known peptides and proteins obtained from public databases. Each association has a corresponding score, thus what we call annotated data is the set of isotope groups for which the score is above some very conservatively selected threshold. A possible extension for LPT, consists on treating all isotope groups as potentially annotated and use the scores out of identification algorithm in some suitable way. For instance, we could have weighted associations in which an isotope group is assigned to certain protein with probability proportional to the scores available.

Marginal likelihood approximations for inference. The not so satisfying feature of active set based approximations to GPC, is that we are ignoring some of the training data. Although some of our findings on the USPS data set actually suggest that this can be beneficial for performance, it is of interest to modify our framework to a version where the inactive set is used in a cost efficient way. The representer theorem for the mean prediction and the approximations for marginal likelihood discussed in Appendices B and D might give inspiration for such extensions.

Reducing ambiguities in sparse classification. It is well known that in multi-category classification there is not a unique way to represent classification boundaries. This means that several weight matrices can produce the same classifier, so the model is unfortunately unidentifiable. As far as we know, there is nothing we can do to turn a linear multi-category classifier into a fully identifiable model, however we can at least try to remove as many redundancies as possible, mainly to improve mixing and to ease interpretation. For instance, we can share sparsity indicators across columns of $\mathbf{W}$, meaning that we have to remove category specific sparsities, i.e. $r_{1i}, \ldots, r_{Ci} = r_i$. This will not only make inference faster and reduce the number of discrete variables to be inferred, but it will make interpretation easier in the sense that we will not have to think about category specific variables (genes). The latter is a nice feature to have in a model because we can for example relate genes to particular conditions/diseases, however we cannot be entirely sure if we are observing a truly biological interaction or a byproduct of the ambiguities of the model.

Embedded clustering for sparse classification. It is very common in microarray data that some groups of genes tend to have similar expression profiles. This kind of covariation is very important because it may indicate groups of

genes conforming functional pathways. In sparse classification this phenomenon can be seen from two different angles. (i) two highly correlated variables must be kept in the model if they as separate entities help to the classification problem. (ii) If the two variables are discriminant but highly correlated, there is not a reason to keep both of them in the model since we will have to increase the complexity of the model without need. In theory, SBMC must be able to handle the two angles successfully by indicating that the probability for the two variables of being in the model is 0.5, i.e. there are two modes, one of them using one of the variables and the other with the alternative. The problem is that in practice with thousands of variables and limited computational resources, such situations are hard to detect. Thus it is very likely we end up with a model using one of the variables and discarding the other one. One way to avoid this undesirable effect is to group variables with similar profiles. For this purpose, we can borrow the ideas from our LPT model and perform classification on a set of *latent gene* variables instead of the original input space. This has the benefit of reducing the complexity of the sparse classification model while grouping variables with similar gene expression profiles, thus making interpretation more easy to handle.

Generative non-linear manifold models. Principal Component Analysis (PCA) is perhaps the most widely used method for exploratory data analysis and visualization. It can be viewed as a linear generative model where a small number of independent normally distributed latent variables explain as much as possible the covariation in the data. Real world data is often well-described by a low dimensional manifold provided by linear factor models such as PCA, however it is usually not normally distributed. Thus, PCA is a good method for finding a latent representation but a poor generative model. It could be interesting to investigate non-linear manifold methods that use a factor model with flexible latent variable distributions. For example we can consider popular non-parametric priors like Dirichlet Process (DP) or Gaussian Process priors as principled ways of allowing for non-linearities.

Appendix A

Bayesian Sparse Factor Models and DAGs Inference and Comparison

---

Appears in
Neural Information processing Systems 22

Available from NIPS at
`http://books.nips.cc/papers/files/nips22/NIPS2009_0643.pdf`

Complementary website at
`http://cogsys.imm.dtu.dk/slim`

Bayesian Sparse Factor Models and DAGs Inference and Comparison

Ricardo Henao
DTU Informatics
Technical University of Denmark
2800 Lyngby, Denmark
Bioinformatics Centre
University of Copenhagen
2200 Copenhagen, Denmark
rhenao@binf.ku.dk

Ole Winther
DTU Informatics
Technical University of Denmark
2800 Lyngby, Denmark
Bioinformatics Centre
University of Copenhagen
2200 Copenhagen, Denmark
owi@imm.dtu.dk

Abstract

In this paper we present a novel approach to learn directed acyclic graphs (DAGs) and factor models within the same framework while also allowing for model comparison between them. For this purpose, we exploit the connection between factor models and DAGs to propose Bayesian hierarchies based on spike and slab priors to promote sparsity, heavy-tailed priors to ensure identifiability and predictive densities to perform the model comparison. We require identifiability to be able to produce variable orderings leading to valid DAGs and sparsity to learn the structures. The effectiveness of our approach is demonstrated through extensive experiments on artificial and biological data showing that our approach outperform a number of state of the art methods.

1 Introduction

Sparse factor models have proven to be a very versatile tool for detailed modeling and interpretation of multivariate data, for example in the context of gene expression data analysis [1, 2]. A sparse factor model encodes the prior knowledge that the latent factors only affect a limited number of the observed variables. An alternative way of modeling the data is through linear regression between the measured quantities. This multiple regression model is a well-defined multivariate probabilistic model if the connectivity (non-zero weights) defines a directed acyclic graph (DAG). What usually is done in practice is to consider either factor or DAG models. Modeling the data with both types of models at the same time and then perform model comparison should provide additional insight as these models are complementary and often closely related. Unfortunately, existing off-the-shelf models are specified in such a way that makes direct comparison difficult. A more principled idea that can phrased in Bayesian terms is for example to find an equivalence between both models, then represent them using a common/comparable hierarchy, and finally use a marginal likelihood or a predictive density to select one of them. Although a formal connection between factor models and DAGs has been already established in [3], this paper makes important extensions such as explicitly modeling sparsity, stochastic search over the order of the variables and model comparison.

Is well known that learning the structure of graphical models, in particular DAGs is a very difficult task because it turns out to be a combinatorial optimization problem known to be NP-hard [4]. A commonly used approach for structure learning is to split the problem into two stages using the fact that the space of variable orderings is far more smaller than the space of all possible structures, e.g. by first attempting to learn a suitable permutation of the variables and then the skeleton of the structure given the already found ordering or viceversa. Most of the work so far for continuous data assumes linearity and Gaussian variables hence they can only recover the DAG structure up

to Markov equivalence [5, 6, 7, 8], which means that some subset of links can be reversed without changing the likelihood [9]. To break the Markov equivalence usually experimental (interventional) data in addition to the observational (non-interventional) data is required [10]. In order to obtain identifiability from purely observational data, strong assumptions have to to be made [11, 3, 12]. In this work we follow the line of [3] by starting from a linear factor model and ensure identifiability by using non-normal heavy-tailed latent variables. As a byproduct we find a set of candidate orderings compatible with a linear DAG, i.e. a mixing matrix which is "close to" triangular. Finally, we may perform model comparison between the factor and DAG models inferred with fixed orderings taken from the candidate set.

The rest of the paper is organized as follows. Sections 2 to 5 we motivate and describe the different ingredients in our method, in Section 6 we discuss existing work, in Section 7 experiments on both artificial and real data are presented, and Section 8 concludes with a discussion and perspectives for future work.

2 From DAGs to factor models

We will assume that an ordered d-dimensional data vector $\mathbf{Px}$ can be represented as a directed acyclic graph with only observed nodes, where $\mathbf{P}$ is the usually unknown true permutation matrix. We will focus entirely on linear models such that the value of each variable is a linear weight combination of parent nodes plus a driving signal $\mathbf{z}$

$$\mathbf{x} = \mathbf{P}^{-1}\mathbf{BPx} + \mathbf{z} \,, \tag{1}$$

where $\mathbf{B}$ is a strictly lower triangular square matrix. In this setting, each non-zero element of $\mathbf{B}$ corresponds to a link in the DAG. Solving for $\mathbf{x}$ we can rewrite the problem as

$$\mathbf{x} = \mathbf{P}^{-1}\mathbf{APz} = \mathbf{P}^{-1}(\mathbf{I}-\mathbf{B})^{-1}\mathbf{Pz} \,, \tag{2}$$

which corresponds to a noise-free linear factor model with the restriction that $\mathbf{P}^{-1}\mathbf{AP}$ must have a sparsity pattern that can be permuted to a triangular form since $(\mathbf{I}-\mathbf{B})^{-1}$ is triangular. This requirement alone is not enough to ensure identifiability (up to scaling and permutation of columns $\mathbf{P}_f$)[1]. We further have to use prior knowledge about the distribution of the factors $\mathbf{z}$. A necessary condition is that these must be a set of non-Gaussian independent variables [11]. For heavy-tailed data is it often sufficient in practice to use a model with heavier tails than Gaussian [13]. If the requirements for $\mathbf{A}$ and for the distribution of $\mathbf{z}$ are met, we can first estimate $\mathbf{P}^{-1}\mathbf{AP}$ and subsequently find $\mathbf{P}$ searching over the space of all possible orderings. Recently, [3] applied the fastICA algorithm to solve for the inverse mixing matrix $\mathbf{P}^{-1}\mathbf{A}^{-1}\mathbf{P}$. To find a candidate solution for $\mathbf{B}$, $\mathbf{P}$ is set such that $\mathbf{B}$ found from the direct relation equation (1), $\mathbf{B} = \mathbf{I} - \mathbf{A}^{-1}$ (according to magnitude-based criterion) is as close as possible to lower triangular. In the final step the Wald statistic is used for pruning $\mathbf{B}$ and the chi-square test is used for model selection.

In our work we also exploit the relation between the factor models and linear DAGs. We apply a Bayesian approach to learning a sparse factor models and DAGs, and the stochastic search for $\mathbf{P}$ is performed as an integrated part of inference of the sparse factor model. The inference of factor model (including order) and DAG parameters are performed as two separate inferences such that the only input that comes from the first part is a set of candidate orders.

3 From factor models to DAGs

Our first goal is to perform model inference in the families of factor and linear DAG models. We specify the joint distribution or *probability of everything*, e.g. for the factor model, as

$$p(\mathbf{X}, \mathbf{A}, \mathbf{Z}, \mathbf{\Psi}, \mathbf{P}, \cdot) = p(\mathbf{X}|\mathbf{A}, \mathbf{Z}, \mathbf{P}, \cdot)p(\mathbf{A}|\cdot)p(\mathbf{Z}|\cdot)p(\mathbf{\Psi}|\cdot)p(\mathbf{P}|\cdot)p(\cdot) \,,$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, $\mathbf{Z} = [\mathbf{z}_1, \ldots \mathbf{z}_N]$, N is the number of observations and $(\cdot)$ indicates additional parameters in the hierarchical models. The prior over permutation $p(\mathbf{P}|\cdot)$ will always be chosen to be uniform over the $d!$ possible values. The actual sampling based inference for $\mathbf{P}$ is discussed in the next section and the standard Gibbs sampling components are provided in the supplementary material. Model comparison should ideally be performed using the marginal likelihood. This is more difficult to calculate with sampling than obtaining samples from the posterior so we use the predictive densities on a test set as a yardstick.

[1] These ambiguities are not affecting our ability to find correct permutation $\mathbf{P}$ of the rows.

Factor model Instead of using the noise-free factor model of equation (2) we allow for additive noise $\mathbf{x} = \mathbf{P}_{\mathrm{r}}^{-1}\mathbf{A}\mathbf{P}_{\mathrm{c}}\mathbf{z} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is an additional Gaussian noise term with diagonal covariance matrix $\boldsymbol{\Psi}$, i.e. uncorrelated noise, to account for independent measurement noise, $\mathbf{P}_{\mathrm{r}} = \mathbf{P}$ is the permutation matrix for the rows of $\mathbf{A}$ and $\mathbf{P}_{\mathrm{c}} = \mathbf{P}_{\mathrm{f}}\mathbf{P}_{\mathrm{r}}$ another permutation for the columns with $\mathbf{P}_{\mathrm{f}}$ accounting for the permutation freedom of the factors. We will not restrict the mixing matrix $\mathbf{A}$ to be triangular. Instead we infer $\mathbf{P}_{\mathrm{r}}$ and $\mathbf{P}_{\mathrm{c}}$ using a stochastic search based upon closeness to triangular as measured by a masked likelihood, see below. Now we can specify a hierarchy for the Bayesian model as follows

$$\begin{aligned} \mathbf{X}|\mathbf{P}_{\mathrm{r}}, \mathbf{A}, \mathbf{P}_{\mathrm{c}}, \mathbf{Z}, \boldsymbol{\Psi} &\sim \mathcal{N}(\mathbf{X}|\mathbf{P}_{\mathrm{r}}^{-1}\mathbf{A}\mathbf{P}_{\mathrm{c}}\mathbf{Z}, \boldsymbol{\Psi}) \,, \qquad \mathbf{Z} \sim \pi(\mathbf{Z}|\cdot) \,, \\ \psi_i^{-1}|s_s, s_r &\sim \mathrm{Gamma}(\psi_i^{-1}|s_s, s_r) \,, \qquad \mathbf{A} \sim \rho(\mathbf{A}|\cdot) \,, \end{aligned} \tag{3}$$

where ψ_i are elements of $\boldsymbol{\Psi}$. For convenience, to exploit conjugate exponential families we are placing a gamma prior on the precision of $\boldsymbol{\epsilon}$ with shape s_s and rate s_r. Given that the data is standardized, the selection of hyperparameters for ψ_i is not very critical as long as both "signal and noise" are supported. The prior should favor small values of ψ_i as well as providing support for $\psi_i = 1$ such that certain variables can be explained solely by noise (we set $s_s = 2$ and $s_r = 0.05$ in the experiments).

For the factors we use a heavy-tailed prior $\pi(\mathbf{Z}|\cdot)$ in the form of a Laplace distribution parameterized for convenience as a scale mixture of Gaussians [14]

$$z_{jn}|\mu, \lambda \sim \mathrm{Laplace}(z_{jn}|\mu, \lambda) = \int_0^\infty \mathcal{N}(z_{jn}|\mu, \upsilon)\mathrm{Exponential}(\upsilon_{jn}|\lambda^2)d\upsilon_{jn} \,, \tag{4}$$

$$\lambda^2|\ell_s, \ell_r \sim \mathrm{Gamma}(\lambda^2|\ell_s, \ell_r) \,, \tag{5}$$

where z_{jn} is an element of $\mathbf{Z}$, λ is the rate and υ has an exponential distribution acting as mixing density. Furthermore, we place a gamma distribution on λ^2 to get conditionals for υ and λ^2 in standard conjugate families. We let the components of $\mathbf{Z}$ have on average unit variance. This is achieved by setting $\ell_s/\ell_r = 2$ (we set $\ell_s = 4$ and $\ell_r = 2$). Alternatively one may use a t distribution—again as scale mixture of Gaussians—which can to interpolate between very heavy-tailed (power law) and very light tails, i.e. becoming Gaussian when degrees of freedom approaches infinity. However such flexibility comes at the price of being more difficult to select its hyperparameters, because the model could become unidentified for some settings.

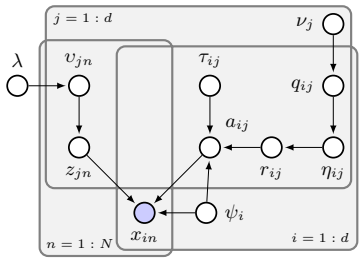


Figure 1: Graphical model for Bayesian hierarchy in equation (3).

The prior $\rho(\mathbf{A}|\cdot)$ for the mixing matrix should be biased towards sparsity because we want to infer something close to a triangular matrix. Here we adopt a two-layer discrete spike and slab prior for the elements a_{ij} of $\mathbf{A}$ similar to the one in [2]. The first layer in the prior control the sparsity of each element a_{ij} individually, whereas the second layer impose a per-factor sparsity level to allow elements within the same factor to share information. The hierarchy can be written as

$$\begin{aligned} a_{ij}|r_{ij}, \psi_i, \tau_{ij} &\sim (1 - r_{ij})\delta(a_{ij}) + r_{ij}\mathcal{N}(a_{ij}|0, \psi_i\tau_{ij}) \,, \\ \tau_{ij}^{-1}|t_s, t_r &\sim \mathrm{Gamma}(\tau_{ij}^{-1}|t_s, t_r) \,, \\ r_{ij}|\eta_{ij} &\sim \mathrm{Bernoulli}(r_{ij}|\eta_{ij}) \,, \\ \eta_{ij}|q_{ij}, \alpha_p, \alpha_m &\sim (1 - q_{ij})\delta(\eta_{ij}) + q_{ij}\mathrm{Beta}(\eta_{ij}|\alpha_p\alpha_m, \alpha_p(1 - \alpha_m)) \,, \\ q_{ij}|\nu_j &\sim \mathrm{Bernoulli}(q_{ij}|\nu_j) \,, \\ \nu_j|\beta_m, \beta_p &\sim \mathrm{Beta}(\nu_j|\beta_p\beta_m, \beta_p(1 - \beta_m)) \,, \end{aligned} \tag{6}$$

where $\delta(\cdot)$ is a Dirac δ-function. The prior above specify a point mass mixture over a_{ij} with mask r_{ij}. The expected probability of a_{ij} to be non-zero is η_{ij} and is controlled through a beta hyperprior with mean α_m and precision α_p. Besides, each factor has a common sparsity rate ν_j that let the elements η_{ij} to be exactly zero with probability $1 - \nu_j$ through a beta distribution with mean β_m and

precision β_p, turning the distribution of η_{ij} bimodal over the unit interval. The magnitude of non-zero elements in $\mathbf{A}$ is specified through the slab distribution depending on τ_{ij}. The parameters for τ_{ij} should be specified in the same fashion as ψ_i but putting more probability mass around $a_{ij} = 1$, for instance $t_s = 4$ and $t_r = 10$. Note that we scale the variances with ψ_i since it makes the model easier to specify and tend to have better mixing properties [15]. The masking matrix r_{ij} with parameters η_{ij} should be somewhat diffuse while favoring relatively large masking probabilities, e.g. $\alpha_p = 10$ and $\alpha_m = 0.9$. Additionally, q_j and should favor very small values with low variance, this is for example $\beta_p = 1000$ and $\beta_m = 0.005$. The graphical model for the entire hierarchy in (3) omitting parameters is shown in Figure 1.

DAG We make the following Bayesian specification of linear DAG model of equation (1) as

$$\mathbf{X}|\mathbf{P}_{\mathrm{r}}, \mathbf{B}, \mathbf{X}, \cdot \sim \pi(\mathbf{X} - \mathbf{P}_{\mathrm{r}}^{-1}\mathbf{B}|\cdot) , \quad \mathbf{B} \sim \rho(\mathbf{B}|\cdot) , \tag{7}$$

where π and ρ are given by equations (4) and (6). The Bayesian specification for the DAG has a similar graphical model to the one in Figure 1 but without noise variances $\mathbf{\Psi}$. The factor model needs only shared variance parameter λ for the Laplace distributed z_{jn} because a change of scale in $\mathbf{A}$ is equivalent to change of variance in z_{jn}. The DAG on the other hand, needs individual variance parameters because it has no scaling freedom. Given that we know that $\mathbf{B}$ is strictly lower triangular, it should be in general less sparse than $\mathbf{A}$, thus we use a different setting for the sparsity prior, i.e. $\beta_p = 100$ and $\beta_m = 0.01$.

4 Sampling based inference

For given permutation $\mathbf{P}$, Gibbs sampling can be used for inference of the remaining parameters. Details of Gibbs sampler is given in the supplementary material and we will focus on the non-standard inference corresponding to the sampling over permutations. There are basically two approaches to find $\mathbf{P}$, one is perform the inference for parameters and $\mathbf{P}$ jointly with $\mathbf{B}$ restricted to be triangular. The other is to let the factor model be unrestricted and search for $\mathbf{P}$ according to a criterion that does not affect parameter inference. Here we prefer the latter for two reasons. First, joint combinatorial and parameter inference in this model will probably have poor mixing with slow convergence. Second, we are also interested in comparing the factor model against the DAG for cases when we cannot really assume that the data is well approximated by a DAG. In our approach the proposal $\mathbf{P}^\star$ corresponds to picking two of the elements in the order vector by random and exchanging them. Other approaches such as restricting to pick two adjacent elements have been suggested as well [16, 7]. For the linear DAG model we are not performing joint inference of $\mathbf{P}$ and the model parameters. Rather we use a set of $\mathbf{P}$s found for the factor model to be good candidates for the DAG.

The stochastic search for $\mathbf{P} = \mathbf{P}_{\mathrm{c}}$ goes as follows: we make inference for the unrestricted factor model, propose $\mathbf{P}_{\mathrm{r}}^\star$ and $\mathbf{P}_{\mathrm{c}}^\star$ independently according $q(\mathbf{P}_{\mathrm{r}}^\star|\mathbf{P}_{\mathrm{r}})q(\mathbf{P}_{\mathrm{c}}^\star|\mathbf{P}_{\mathrm{c}})$ which is the uniform two variable random exchange. With this proposal and the flat prior over $\mathbf{P}$, we use a Metropolis-Hastings acceptance probability simply as the ratio of likelihoods with $\mathbf{A}$ *masked to have zeros above its diagonal* (through masking matrix $\mathbf{M}$)

$$\xi_{\to\star} = \frac{\mathcal{N}(\mathbf{X}|(\mathbf{P}_{\mathrm{r}}^\star)^{-1}(\mathbf{M} \odot \mathbf{P}_{\mathrm{r}}^\star\mathbf{A}(\mathbf{P}_{\mathrm{c}}^\star)^{-1})\mathbf{P}_{\mathrm{c}}^\star, \mathbf{\Psi})}{\mathcal{N}(\mathbf{X}|\mathbf{P}_{\mathrm{r}}^{-1}(\mathbf{M} \odot \mathbf{P}_{\mathrm{r}}\mathbf{A}\mathbf{P}_{\mathrm{c}}^{-1})\mathbf{P}_{\mathrm{c}}, \mathbf{\Psi})} ,$$

The procedure can be seen as a simple approach for generating hypotheses about good, close to triangular $\mathbf{A}$, orderings in a model where the spike and slab prior provides bias towards sparsity.

To learn DAGs we first perform inference on the factor model specified by the hierarchy in (3) to obtain a set of ordering candidates sorted according to their usage during sampling—after the burn-in period. It is possible that the estimation of $\mathbf{A}$ might contain errors, e.g. a false zero entry on $\mathbf{A}$ allowing several orderings leading to several lower triangular versions of $\mathbf{A}$, only one of those being actually correct. Thus, we propose not only to use the best candidate but a set of top candidates of size $m_{\mathrm{top}} = 10$. Then we perform inference on the DAG model corresponding to the structure search hierarchy in (7), for each one of the permutation candidates being considered, $\mathbf{P}_{\mathrm{r}}^{(1)}, \ldots, \mathbf{P}_{\mathrm{r}}^{(m_{\mathrm{top}})}$. Finally, we select the DAG model among candidates using the predictive distribution for the DAG when a test set is available or just the likelihood if not.

5 Predictive distributions and model comparison

Given that our model produces both DAG and a factor model estimates at the same time, it could be interesting to estimate also whether one option is better than the other given the observed data, for example in exploratory analysis when the DAG assumption is just one reasonable option. In order to perform the model comparison, we use predictive densities $p(\mathbf{X}^\star|\mathbf{X}, \mathcal{M})$ with $\mathcal{M} = \{\mathcal{M}_{\mathrm{FA}}, \mathcal{M}_{\mathrm{DAG}}\}$, instead of marginal likelihoods because the latter is difficult and expensive to compute by sampling, requiring for example thermodynamic integration. With Gibbs sampling, we draw samples from the posterior distributions $p(\mathbf{A}, \mathbf{\Psi}, \lambda|\mathbf{X}, \cdot)$ and $p(\mathbf{B}, \lambda_1, \ldots, \lambda_m|\mathbf{X}, \cdot)$. The average over the extensive variables associated with the test points $p(\mathbf{Z}^\star|\cdot)$ is a bit more complicated because naively drawing samples from $p(\mathbf{Z}^\star|\cdot)$ gives an estimator with high variance—for $\psi_i \ll v_{jn}$. In the following we describe how to do it for each model, omitting the permutation matrices for clarity.

Factor model We can compute the predictive distribution by taking the likelihood in equation (3) and marginalizing $\mathbf{Z}$. Since the integral has no closed form we can approximate it using the Gaussian distribution from the scale mixture representation as

$$p(\mathbf{X}^\star|\mathbf{A}, \mathbf{\Psi}, \cdot) = \int p(\mathbf{X}^\star|\mathbf{A}, \mathbf{Z}, \mathbf{\Psi}) p(\mathbf{Z}|\cdot) d\mathbf{Z} \approx \frac{1}{\mathrm{rep}} \prod_n \sum_r^{\mathrm{rep}} \mathcal{N}(\mathbf{x}_n^\star|\mathbf{0}, \mathbf{A}^\top \mathbf{U}_n \mathbf{A} + \mathbf{\Psi}) \,,$$

where $\mathbf{U}_n = \mathrm{diag}(v_{1n}, \ldots, v_{dn})$, the v_{jn} are sampled from the prior and rep is the number of samples generated to approximate the intractable integral ($\mathrm{rep} = 500$ in the experiments). Then we can average over $p(\mathbf{A}, \mathbf{\Psi}, \lambda|\mathbf{X}, \cdot)$ to obtain $p(\mathbf{X}^\star|\mathbf{X}, \mathcal{M}_{\mathrm{FA}})$.

DAG In this case the predictive distribution is rather easy because the marginal over $\mathbf{Z}$ in equation (4) is just a Laplace distribution with mean $\mathbf{BX}$

$$p(\mathbf{X}^\star|\mathbf{B}, \cdot) = \int p(\mathbf{X}^\star|\mathbf{B}, \mathbf{X}, \mathbf{Z}) p(\mathbf{Z}|\cdot) d\mathbf{Z} = \prod_{i,n} \mathrm{Laplace}(x_{ij}|[\mathbf{BX}]_{in}, \lambda_i) \,,$$

where $[\mathbf{BX}]_{ij}$ is the element indexed by the i-th row and n-th column of $\mathbf{BX}$. In practice we compute the predictive densities for a particular $\mathbf{X}^\star$ during sampling and then select the model based on its ratio. Note that both predictive distributions depend directly on λ—the rate of Laplace distribution, making the estimates highly dependent on its value. This is why it is important to have the hyperprior on λ of equation (5) instead of just fixing its value.

6 Existing work

Among the existing approaches to DAG learning, our work is most closely related to LiNGAM (Linear Non-Gaussian Acyclic Model for causal discovery) [3] with several important differences: Since LiNGAM relies on fastICA to learn the mixing is not inherently sparse, hence a pruning procedure based on Wald statistic and model fit second order information should be applied after obtaining an ordering for the variables. The order search in LiNGAM assumes that there is not estimation errors during fastICA model inference, then a single ordering candidate is produced. LiNGAM produces and select a final model among several candidates, but in contrast to our method such candidates are not different DAGs with different variable orderings but DAGs with different sparsity levels. The factor model inference in LiNGAM, namely fastICA is very efficient however their structure search involves repeated inversions of matrices of sizes $d^2 \times d^2$ which can make it prohibitive for large problems. More explicitly, the computational complexity of LiNGAM is roughly $\mathcal{O}(N_{\mathrm{fit}} d^6)$ where N_{fit} is the number of model fit evaluations. In contrast, the complexity in our case is $\mathcal{O}(N_{\mathrm{ite}} d^2 N)$ where N_{ite} is the total number of samples including burn-in periods for both, factor model and DAG inferences. Finally, our model is more principled in the sense that all the approach is within the same Bayesian framework, as a result it can be extended to for example binary data or time series by selecting some suitable prior distributions.

Much work on Bayesian models for DAG learning already exist. For example, the approach presented in [16] is a Gaussian Bayesian network and therefore suffers from lack of identifiability. Besides, order search is performed directly for the DAG model making necessary the use of longer

sampler runs with a number of computational tricks when the problem is large ($d > 10$), i.e. when exhaustive order enumeration is not an option.

7 Experiments

We consider four sets of experiments in the following. The first two consist on extensive experiments using artificial data, the third addresses the model comparison scenario and the last one uses real data previously published in [17]. In every case we ran 2000 samples after a burn-in period of 4000 iterations and three independent chains for the factor model, and a single chain with 1000 samples and 2000 as burn-in for the DAG[2]. Hyperparameter settings are discussed in Section 3.

LiNGAM suite We evaluate the performance of our model against LiNGAM[3] using the artificial model generator presented in [3]. The generator produces both dense and sparse networks with different degree of sparsity, $\mathbf{Z}$ is generated from a non-Gaussian heavy-tailed distribution, $\mathbf{X}$ is generated using equation (1) and then randomly permuted to hide the correct order, $\mathbf{P}$. For the experiment we have generated 1000 different dataset/models using $d = \{5, 10\}$, $N = \{200, 500, 1000, 2000\}$ and the DAG was selected using the (training set) likelihood in equation (7). Results are summarized in Figure 2 using several performance measures. For the particular case of the area under the ROC curve (AUC), we use the conditional posterior of the masking matrix, i.e. $p(\mathbf{R}|\mathbf{X}, \cdot)$ where $\mathbf{R}$ is a matrix with elements r_{ij}. AUC is an important measure because it quantifies how the model accounts for the uncertainty of presence or absence of links in the DAG. Such uncertainty assessment is not possible in LiNGAM where the probability of having a link is simply zero or one, however the AUC can be still computed.

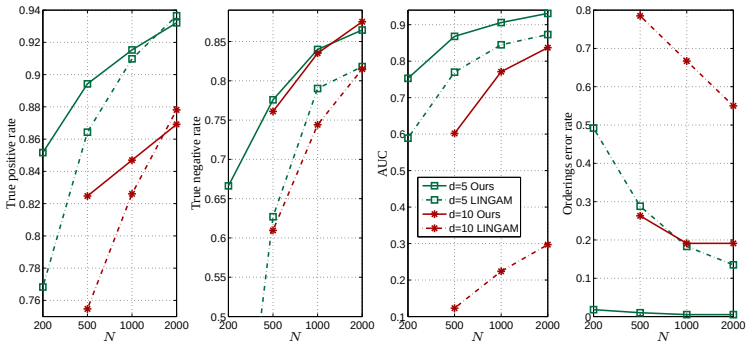
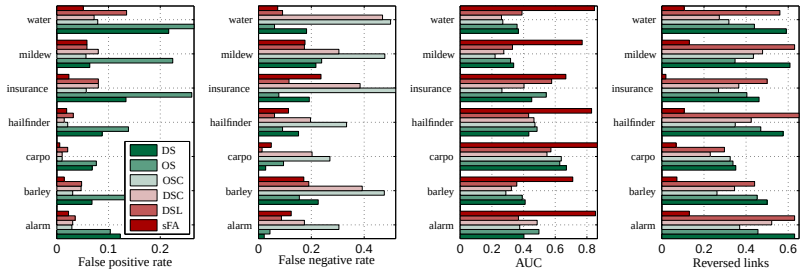


Figure 2: Performance measures for LiNGAM suite. Symbols are: square for 5 variables, star for 10 variables, solid line for sFA and dashed line for LiNGAM. (a) True positive rate. (b) True negative rate. (c) Frequency of AUC being greater than 0.9. (d) Number of estimated correct orderings.

In terms of true negative rates, AUC and ordering error rate, our approach is significantly better than LiNGAM. The true positive rate results in Figure 2(a) show that LiNGAM outperform our approach only for $N = 2000$. However by comparing it to the true positive rate, it seems than LiNGAM prefer more dense models which could be an indication of overfitting. Looking to the ordering errors, our model is clearly superior. It is important to mention that being able to compute a probability for a link in the DAG to be zero, $p(b_{ij} \neq 0|\mathbf{X}, \cdot)$, turns out to be very useful in practice, for example to reject links with high uncertainty or to rank them. To give an idea of running times on a regular two-core 2.5GHz machine, for $d = 10$ and $N = 500$: LiNGAM took in average 10 seconds and our method 170 seconds. However, when doubling the number of variables the times were 730 and 550 seconds for LiNGAM and our method respectively, which is in agreement with our complexity estimates.

[2]Source code available upon request (C with Matlab interface).
[3]Matlab package available at http://www.cs.helsinki.fi/group/neuroinf/lingam/.

Bayesian networks repository Next we want to compare some of the state of the art (Gaussian) approaches to DAG learning on 7 well known structures[4], namely alarm, barley, carpo, hailfinder, insurance, mildew and water ($d = 37, 48, 61, 56, 27, 35, 32$ respectively). A single dataset of size 1000 per structure was generated using a similar procedure to the one used before. Apart from ours (sFA), we considered the following methods[5]: standard DAG search (DS), order-search (OS), sparse candidate pruning then DAG-search (DSC) [6], L1MB then DAG-search (DSL) [8], sparse-candidate pruning then order-search (OSC) [7]. Results are shown in Figure 3, including the number of reversed links found due to ordering errors.



(a) (b) (c) (d)

Figure 3: Performance measures for Bayesian networks repository experiments.

In this case, our approach obtained slightly better results when looking at the false positive rate, Figure 3(a). The true negative rate is comparable to the other methods suggesting that our model in some cases is sparser than the others. AUC estimates are significantly better because we have continuous probabilities for links to be zero (in the other methods we had to use a binary value). From Figure 3(d), the number of reversed links in the other methods is quite high as expected due to lack of identifiability. Our model produced a small amount reversed links because it was not able to find any of the true orderings, but indeed something quite close. This results could be improved by running the sampler for a longer time or by considering more candidates. We also tried to run the other approaches with data generated from Gaussian distributions but the results were approximately equal to those shown in Figure 3. On the other hand, our approach performs similarly but the number of reversed links increases significantly since the model is no longer identified. The most important advantage of the (Gaussian) methods used in this experiment is their speed. In all cases they are considerably faster than sampling based methods. Their speed make them very suitable for large scale problems regardless of their identifiability issues.

Model comparison For this experiment we have generated 1000 different datasets/models with $d = 5$ and $N = \{500, 1000\}$ in a similar way to the first experiment but this time we selected the true model to be a factor model or a DAG uniformly. In order to generate a factor model we basically just need to be sure that $\mathbf{A}$ cannot be permuted to a triangular form. We kept 20% of the data to compute the predictive densities to then select between all estimated DAG candidates and the factor model. We found that for $N = 500$ our approach was able to select true DAGs 91.5% of the times and true factor models 89.2%, corresponding to an overall error of 9.6%, For $N = 1000$ the true DAG and true factor model rates increased to 98.5% and 94.6% respectively. This results demonstrate that our approach is very effective at selecting the true underlying structure in the data between the two proposed hypotheses.

Protein-signaling network The dataset introduced in [17] consists on flow cytometry measurements of 11 phosphorylated proteins and phospholipids (Raf, Erk, p38, Jnk, Akt, Mek, PKA, PKC, PIP_2, PIP_3, PLCγ). Each observation is a vector of quantitative amounts measured from single cells, generated from a series of stimulatory cues and inhibitory interventions. The dataset contains both observational and experimental data. Here we are only using 1755 samples corresponding to

[4] http://compbio.cs.huji.ac.il/Repository/.

[5] Parameters: 10000 iterations, 5 candidates (SC, DSC), max fan-in of 5 (OS, OSC) and *Or* strategy and MDL penalty (DSL).

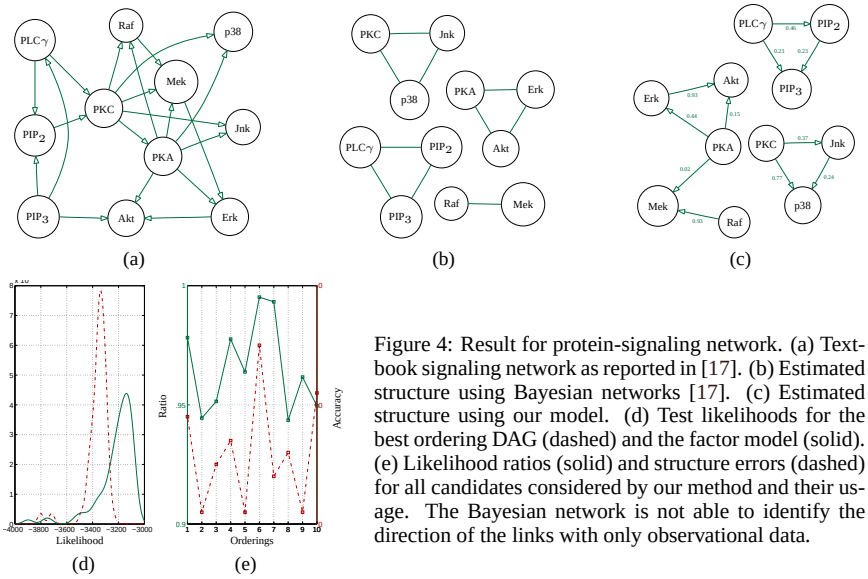


Figure 4: Result for protein-signaling network. (a) Textbook signaling network as reported in [17]. (b) Estimated structure using Bayesian networks [17]. (c) Estimated structure using our model. (d) Test likelihoods for the best ordering DAG (dashed) and the factor model (solid). (e) Likelihood ratios (solid) and structure errors (dashed) for all candidates considered by our method and their usage. The Bayesian network is not able to identify the direction of the links with only observational data.

pure observational data and randomly selected 20% of the data to compute the predictive densities. Using the entire set will produce a richer model, however interventions are out of the scope of this paper. The textbook ground truth and results are presented in figure 4. From the 21 possible links in figure 4(a), the model from [17] was able to find 9, but also one falsely added link. In 4(b), a marginal likelihood equivalent prior is used and they therefore cannot make any inferences about directionality from observational data alone, see Figure 4(b). Our model in Figure 4(c) was able to find 10 true links, one falsely added link and only two reversed links (RL), one of them is $\text{PIP}_2 \rightarrow \text{PIP}_3$ which according to the ground truth is bidirectional and the other one, $\text{PLC}\gamma \rightarrow \text{PIP}_3$ which was also found reversed using experimental data in [17]. Note from figure 4(e) that the predictive density ratios correlate quite well with the structural accuracy. The predictive densities for the best candidate (sixth in Figure 4(e)) is shown in Figure 4(d) and suggests that the factor model is a better option which makes sense considering that estimated DAG in figure 4(c) is a substructure of the ground truth. We also examined the estimated factor model and we found out that three factors could correspond to unmeasured proteins (PI3K, MKK and IP3), see Figure 2 and table 3 in [17]. We also tried the above methods. Results were very similar to our method in terms of true positives (≈ 9) and true negatives (≈ 32), however none of them were able to produce less than 6 reversed links that corresponds to approximately two-thirds of total true positives.

8 Discussion

We have proposed a novel approach to perform inference and model comparison of sparse factor models and DAGs within the same framework. The key ingredients for both Bayesian models are spike and slab priors to promote sparsity, heavy-tailed priors to ensure identifiability and predictive densities to perform the comparison. A set of candidate orderings is produced by the factor model. Subsequently, a linear DAG is learned for each of the candidates. To the authors' knowledge this is the first time that a method for comparing such a closely related linear models is proposed. This setting can be very beneficial in situations where the prior evidence suggests both DAG structure and/or unmeasured variables in the data. For example in the protein signaling network [17], the textbook ground truth suggests both DAG structure and a number of unmeasured proteins. The previous approach [17] only performed structure learning in DAGs but our results suggest that the data is better explained by the factor model. For further exploration of this data set, we obviously need to modify our approach to handle hybrid models, i.e. graphs with directed/undirected links and observed/latent nodes as well as being able to use experimental data. Our Bayesian hierarchical approach is very flexible. We are currently investigating extensions to other source distributions (non-parametric Dirichlet process, temporal Gaussian processes and discrete).

References

[1] M. West. Bayesian factor regression models in the "large p, small n" paradigm. In J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, editors, *Bayesian Statistics 7*, pages 723–732. Oxford University Press, 2003.

[2] J. Lucas, C. Carvalho, Q. Wang, A. Bild, J. R. Nevins, and M. West. *Bayesian Inference for Gene Expression and Proteomics*, chapter Sparse Statistical Modeling in Gene Expression Genomics, pages 155–176. Cambridge University Press, 2006.

[3] S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7:2003–2030, October 2006.

[4] D. M. Chickering. Learning Bayesian networks is NP-complete. In D. Fisher and H.-J. Lenz, editors, *Learning from Data: AI and Statistics*, pages 121–130. Springer-Verlag, 1996.

[5] I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, October 2006.

[6] N. Friedman, I. Nachman, and D. Pe'er. Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm. In K. B. Laskey and H. Prade, editors, *UAI*, pages 206–215, 1999.

[7] M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In *UAI*, pages 548–549, 2005.

[8] M. W. Schmidt, A. Niculescu-Mizil, and K. P. Murphy. Learning graphical model structure using L1-regularization paths. In *AAAI*, pages 1278–1283, 2007.

[9] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, January 1995.

[10] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, March 2000.

[11] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, December 1994.

[12] C. M. Carvalho, J. Chang, J. E. Lucas, J. R. Nevins, Q. Wang, and M. West. High-dimensional sparse factor modeling: Applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484):1438–1456, December 2008.

[13] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley-Interscience, May 2001.

[14] D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society: Series B (Methodology)*, 36(1):99–102, 1974.

[15] T. Park and G. Casella. The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, June 2008.

[16] N. Friedman and D. Koller. Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50(1–2):95–125, January 2003.

[17] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, April 2005.

Appendix B

PASS-GP: Predictive Active Set Selection for Gaussian Processes

Appears in
Machine Learning for Signal Processing (MSLP) 2010

Available from IEEE at
`http://dx.doi.org/10.1109/MLSP.2010.5589264`

Complementary website at
`http://cogsys.imm.dtu.dk/passgp`

PASS-GP: PREDICTIVE ACTIVE SET SELECTION FOR GAUSSIAN PROCESSES

Ricardo Henao and Ole Winther

DTU Informatics, Technical University of Denmark, Denmark
Bioinformatics Centre, University of Copenhagen, Denmark

ABSTRACT

We propose a new approximation method for Gaussian process (GP) learning for large data sets that combines inline active set selection with hyperparameter optimization. The predictive probability of the label is used for ranking the data points. We use the leave-one-out predictive probability available in GPs to make a common ranking for both active and inactive points, allowing points to be removed again from the active set. This is important for keeping the complexity down and at the same time focusing on points close to the decision boundary. We lend both theoretical and empirical support to the active set selection strategy and marginal likelihood optimization on the active set. We make extensive tests on the USPS and MNIST digit classification databases with and without incorporating invariances, demonstrating that we can get state-of-the-art results (e.g. 0.86% error on MNIST) with reasonable time complexity.

1. INTRODUCTION

Gaussian processes is an attractive non-parametric framework for supervised learning. It is conceptually and algorithmically simple, flexible and fully probabilistic. However, it is limited to tasks with no more than a few thousand training observations, because inference scales cubically and the memory requirements quadratically, with respect to the training set size, N. In comparison with support vector machines (SVM), Gaussian processes usually provide results comparable in terms of prediction power with the additional benefit of probabilistic outputs, i.e. error bars for predictions and Bayesian model selection as a consistent framework for automatically setting the model hyperparameters. The cubic complexity has inspired a considerable amount of recent research on sparse approximations. These methods accelerate the training and prediction times to $\mathcal{O}(NM^2)$ and $\mathcal{O}(M^2)$, respectively. Depending on the method, $M < N$ is the rank in a low-rank approximation to the covariance (Gram) matrix, the size of an active set or of a pseudo-input set. See [1, 2, 3] for a recent review, subsequent unifying view and an even more recent extension of these ideas to classification, respectively. In the fully independent training conditional (FITC) approximation the sparse solution is built on a pseudo-input subset of the training data [2, 3]. The main advantage of FITC over active set methods like Informative Vector Machine (IVM) [4] is that the pseudo-inputs and the hyperparameters can be learned jointly by gradient based optimization, also producing highly sparse solutions. However, since the number of parameters to be learnt grows with the number and dimension of the pseudo-inputs, this method becomes computationally unfeasible for large datasets and may even suffer from overfitting due to the number of free parameters in the model.

The approach presented here, Predictive Active Set Selection (PASS-GP) is an active set selection method in which the selection criterion is based on the (dual representation) weight of the data points. In GP classification this is the same as using the predictive distribution (i.e. include data points with small predictive probability). For points in the active set we can compute the leave-one-out (or cavity) predictive probability and use that for deletions. We alternate between active set updates and hyperparameter optimization based upon the marginal likelihood of the active set. Restricting to an active set is less elegant than FITC. However, currently this appears necessary for large data sets like MNIST and we also present support for this approximation being a very good approximation to running the full GP.

This paper is organized as follows: in Section 2, a brief introduction to GP classifiers using expectation propagation is provided, next in Section 3, our active set selection for GP is described; a justification based upon a 'representer theorem' for the predictive mean is given in 4, a marginal likelihood approximation to the full GP is introduced in Section 5, followed by the experimental results in Section 6. Finally, the discussion is given in Section 7.

2. GAUSSIAN PROCESSES FOR CLASSIFICATION

Given a set of input random variables $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$, a Gaussian process is defined as a joint Gaussian distribution over functions in the input points $\mathbf{f} = [f_1, \ldots, f_N]^T$ with mean vector $\mathbf{m}$ (taken to be zero in the following) and covariance matrix $\mathbf{K}$ with elements $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and hyperparameters $\boldsymbol{\theta}$. For classification assuming independently

observed binary ± 1 labels $\mathbf{y} = [y_1, \ldots, y_N]^T$ and a probit likelihood function $t(y_n|f_n) = \Phi(f_n y_n)$, we have the intractable posterior $p(\mathbf{f}|\mathbf{X}, \mathbf{y}) = \frac{1}{Z} p(\mathbf{f}|\mathbf{X}) \prod_{n=1}^{N} t(y_n|f_n)$, where $Z = p(\mathbf{y}|\mathbf{X})$ is the marginal likelihood. To perform averages we must resort to approximations. Here we use Expectation Propagation (EP) because it is the currently the most accurate deterministic approximation, see e.g. [1, 5]. In EP, the likelihood function is locally approximated by an un-normalized Gaussian distribution

$$q(\mathbf{f}|\mathbf{X}, \mathbf{y}) = p(\mathbf{f}|\mathbf{X}) \prod_{n=1}^{N} z_n \tilde{t}(y_n|f_n) = \frac{1}{Z_{\mathrm{EP}}} p(\mathbf{f}|\mathbf{X}) \mathcal{N}(\mathbf{f}|\tilde{\mathbf{m}}, \widetilde{\mathbf{C}}) = \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{C}) \,, \quad (1)$$

where $p(\mathbf{f}|\mathbf{X}, \mathbf{y}) \approx q(\mathbf{f}|\mathbf{X}, \mathbf{y})$, the z_n are the normalization coefficients and $\tilde{t}(y_n|f_n)$ are the site Gaussian approximations. In order to obtain $q(\cdot)$, one starts from $q(\mathbf{f}|\mathbf{X}, \mathbf{y}) = p(\mathbf{f}|\mathbf{X})$ and update the individual $\tilde{t}_n$ approximations sequentially. For this purpose, we delete the site approximation $\tilde{t}_n$ from the current posterior leading to the cavity distribution $q_{\backslash n}(\mathbf{f}|\mathbf{X}, \mathbf{y}_{\backslash n}) = p(\mathbf{f}|\mathbf{X}) \prod_{i \neq n} z_i \tilde{t}(y_i|f_i)$ from which we can obtain a cavity predictive distribution

$$q(y_n|\mathbf{X}, \mathbf{y}_{\backslash n}) = \int t(y_n|f_n) q_{\backslash n}(\mathbf{f}|\mathbf{X}, \mathbf{y}_{\backslash n}) d\mathbf{f} = \Phi\left(\frac{y_n m_{\backslash n}}{\sqrt{1 + v_{\backslash n}}}\right) \,, \quad (2)$$

where $m_{\backslash n} = v_{\backslash n}(C_{nn}^{-1} m_n - \widetilde{C}_{nn}^{-1} \widetilde{m}_n)$ and $v_{\backslash n} = (C_{nn}^{-1} - \widetilde{C}_{nn}^{-1})^{-1}$. We combine the cavity distribution with the exact likelihood $t(y_n|f_n)$, to obtain the so-called tilted distribution $q_n(\mathbf{f}|\mathbf{X}, \mathbf{y}) = z_n t(y_n|f_n) q_{\backslash n}(\mathbf{f}|\mathbf{X}, \mathbf{y}_{\backslash n})$. Since we need to choose the parameters of the site approximations we must minimize some divergence measure. It is well known that when $q(x)$ is Gaussian, minimizing $\mathrm{KL}(p(x)||q(x))$ is equivalent to moment matching between those two distributions including zero-th moments for the normalizing constants. The EP algorithm iterates by updating each site approximation in turn and makes several passes over the training data to achieve convergence.

With the Gaussian approximation to the posterior distribution in eq. (1), it is possible to calculate the predictive distribution of a binary label as

$$q(y^*|\mathbf{X}, \mathbf{y}, \mathbf{x}^*) = \int t(y^*|f^*) q(f^*|\mathbf{X}, \mathbf{y}, \mathbf{x}^*) df^* = \Phi\left(\frac{y^* m^*}{\sqrt{1 + v^*}}\right) \,, \quad (3)$$

where $q(f^*|\mathbf{X}, \mathbf{y}, \mathbf{x}^*)$ is the approximate predictive Gaussian distribution (the marginal of $q(\mathbf{f}, f^*|\mathbf{X}, \mathbf{y}, \mathbf{x}^*)$) with mean $m^* = \mathbf{k}^{*\top}(\mathbf{K} + \widetilde{\mathbf{C}})^{-1} \widetilde{\mathbf{m}}$ and variance $v^* = k^{**} + \mathbf{k}^{*\top}(\mathbf{K} + \widetilde{\mathbf{C}})^{-1} \mathbf{k}^*$. In addition, the approximation to the marginal likelihood $p(\mathbf{y}|\mathbf{X})$ results to be the normalization constant in eq. (1), i.e. $q(\mathbf{y}|\mathbf{X}) = Z_{\mathrm{EP}}$. $\log Z_{\mathrm{EP}}(\boldsymbol{\theta})$ and its derivatives could be used jointly with conjugate gradient updates to perform model selection under the evidence maximization framework. For a detailed presentation of GP including its implementation details, consult [1, 5].

3. PREDICTIVE ACTIVE SET SELECTION

The EP algorithm is performed by iterative updates of each site approximation using the whole dataset $(\mathbf{X}, \mathbf{y})$. In the active set scenario on the other hand, we only want to approximate the posterior distribution in eq. (1) using a small subset, the active set $(\mathbf{X}_A, \mathbf{y}_A)$. Since exploring all possible active sets is obviously intractable, the problem is how to choose an active set that gives a performance as good as possible within the available computing time. The IVM for instance, computes in each iteration the differential entropy score for all points not already part of the active set $(\mathbf{X}_I, \mathbf{y}_I)$ and make updates by including the single point leading to maximum score. Despite this greedy heuristic, IVM has proved to behave quite well in practice, giving the so far best reported GP performance on the USPS and MNIST tasks [4, 6]. We propose a similar iterative approach with two main modifications:

- **Active set inclusion/deletion** based directly upon the data point weight in prediction. The 'representer theorem' for the mean prediction, discussed in Section 4, leads directly to the weight being expressed in terms of (a derivative of) the cavity predictive probability. This means that we can actually use the predictive distribution for a point in the inactive set to predict the weight it would have if it would be included in the active set. For classification we use the (cavity) predictive probability to decide upon deletion and inclusion because it is monotonically related the weight and a readily interpretable quantity.
- **Hyperparameter optimization** must be an integral part of algorithm, because the weights of the examples (and thus the active set) to a large degree depend upon hyperparameter values. We therefore alternate between active set updates and hyperparameter optimizaton using several passes over the data set to allow for convergence.

Next we discuss the details of our PASS-GP algorithm followed by a detailed comparison with IVM. As already stated, we use the predictive distribution as scoring function for inclusion and deletion. This means that we will include in the updated active set, points above some inclusion threshold, $p_{\mathrm{inc}} \in [0, 1]$. Such a threshold has a clear interpretation, for instance, setting $p_{\mathrm{inc}} = 0.5$ will include all misclassified points by the current GP while $p_{\mathrm{inc}} = 0.6$ will include the points near to the decision boundary as well. Ranking every point in the data at each iteration could become prohibitive for large datasets. In order to cover the whole dataset, we

split data into N_{sub} non-overlapping subsets and process each one of them in each iteration, such that each subset is between 100 and 1000 data points.

A closer look at eq. (2) reveals that the cavity distribution can be seen as a leave-one-out estimator [7]; thus it can be used also as scoring function for deletions from the current active set. This, together with the idea that points in the active set with cavity probability near to one (or greater than p_{del}, the deletion threshold) do not contribute significantly to the resulting decision boundary, can be removed from the active set.

To perform hyperparameter selection jointly with active set updates, we start from a fixed randomly selected active set of size N_{init}. It should be large enough to provide a good initial hyperparameter set values. Since we expect only small corrections of the hyperparameter values between iterations, we start the model selection procedure from the values obtained in the previous one.

Differences between PASS-GP and IVM. Since IVM is the closest relative of PASS-GP, we briefly discuss the main differences between the two: (i) The active set and thus the computational complexity is usually fixed beforehand in IVM. PASS-GP works with inclusion and deletion thresholds instead. (ii) IVM does not allow for deletions from the active set which is a clear disadvantage as points often become irrelevant at a later stage, when more points have been included. In PASS-GP we can make an (almost) unbiased common ranking of all training points active as well as inactive, using a quantity that is meaningful and directly related to the weight of the training point in predictions. Using both inclusions/deletions and several passes over the training set makes PASS-GP quite insensitive to the initial choice of active set. (iii) The hyperparameter optimization is a part of the algorithm in PASS-GP working on subsets of data between updates and iterating over the data set in principle until convergence. IVM makes a single inclusion per step and in principle stops when the limit for the active set is reached. (iv) In terms of complexity time per iteration IVM it is faster than PASS-GP, $\mathcal{O}(N \cdot |A|)$ against $\mathcal{O}(|A|^2 \cdot (2 + N/N_{\mathrm{sub}}))$, however storage requirements are considerable lower, $\mathcal{O}(|A|^2)$ compared to $\mathcal{O}(N \cdot |A|)$.

4. REPRESENTER FOR MEAN PREDICTION

The 'representer theorem' for the posterior mean of $\mathbf{f}$ [7], connects the predictive probability and the weight of a data point. Using that $p(\mathbf{f}|\mathbf{X}) = -\mathbf{K}\frac{\partial}{\partial \mathbf{f}} p(\mathbf{f}|\mathbf{X})$, we get the exact relation for the posterior mean $\langle\, \mathbf{f} \,\rangle = \mathbf{K}\boldsymbol{\alpha}$ with the weight of point n being

$$\alpha_n = \frac{1}{p(\mathbf{y}|\mathbf{X})} \int p(\mathbf{f}|\mathbf{X}) \frac{\partial}{\partial f_n} p(\mathbf{y}|\mathbf{f}) d\mathbf{f}$$
$$= \frac{\langle\, p'(y_n|f_n) \,\rangle_{\setminus n}}{\langle\, p(y_n|f_n) \,\rangle_{\setminus n}} = \frac{\partial}{\partial h} \log \langle\, p(y_n|f_n + h) \,\rangle_{\setminus n} \Big|_{h=0} ,$$

where $\langle\, \cdot \,\rangle_{\setminus n}$ denotes an average over a posterior without the n-th data point and $p'(y_n|f_n) = \partial p'(y_n|f_n)/\partial f_n$. The final expression says that the weight is nothing but the log derivative of the cavity predictive probability $\langle\, p(y_n|f_n) \,\rangle_{\setminus n} = p(y_n|\mathbf{X}, \mathbf{y}_{\setminus n})$. For regression, $p(y_n|f_n) = \mathcal{N}(y_n|f_n, \sigma^2)$ and $\alpha_n = \frac{y_n - \langle\, f_n \,\rangle_{\setminus n}}{\sigma^2 + v_n}$ with $v_n = \langle\, f_n^2 \,\rangle_{\setminus n} - \langle\, f_n \,\rangle^2_{\setminus n}$. α_n will therefore be small when the cavity mean has a small deviation from the target relative to the variance. For a new data point $(\mathbf{x}, y)$ we can calculate the weight of this point *exactly*, replacing the cavity average with the full average in the above. We can therefore predict without any rerunning EP, how much weight this new point will have. For classification we can calculate the weight using the current EP approximation. When $z_n = y_n \langle\, f_n \,\rangle_{\setminus n} / \sqrt{1 + v_n}$ is above ≈ 4, the cavity probability eq. (2) approaches one and $\alpha_n \approx y_n \exp(-z_n^2/2)/\sqrt{2\pi(1 + v_n)}$. This fast decay indicates that GP classification in many cases effectively will be sparse even though $\boldsymbol{\alpha}$ strictly does not contain zeros.

In the inclusion/deletion steps we rank data points according to their weights. For classification we can use the predictive probability directly, since it is a monotonic function of the weight. Including a new data point will of course affect the value of all other weights as well leading to a rearrangement of their rank. Including multiple data points will also invalidate the predicted value of the weights (e.g. think of the extreme of two new data points being identical). We therefore have to recalculate the weights by retraining with EP for classification or simply updating the posterior for regression before going to the next step. If we have already an active set covering the decision regions pretty well then this rearrangement step will amount to a minor adjustment and the approximation will work well.

In this work we have only used the representer theorem for active set selection. It is also possible, but not tested here, to use all training points for prediction while only calculating the posterior on the active set. The inactive set weights are then simply set to the predicted values from the active set posterior. To get the full predictive probability one also has to calculate the contribution to the predictive variances which can be obtained by a similar theorem but for the predictive variance [7].

5. MARGINAL LIKELIHOOD APPROXIMATIONS

In this section we decompose the marginal likelihood in the active and inactive set contributions. We will argue that the contribution from the active set will dominate justifying why we can limit ourselves to optimizing the hyperparameters over this set. In the following section we will investigate this assumption empirically. The marginal likelihood can be decomposed via the chain rule as

$$p(\mathbf{y}|\mathbf{X}) = p(\mathbf{y}_I|\mathbf{y}_A, \mathbf{X}_A, \mathbf{X}_I) p(\mathbf{y}_A|\mathbf{X}_A) , \qquad (4)$$

Digit	0	1	2	3	4	5	6	7	8	9
GP+hopt (%)	0.75	0.70	1.49	1.30	1.69	1.59	0.65	0.60	1.40	0.75
Time (s)	891797±135820									
GP+hPASS-GP (%)	0.70	0.70	1.49	1.20	1.69	1.35	0.80	0.60	1.20	0.70
Time (s)	70683±4892									
PASS-GP (%)	0.70	0.67	1.25	1.03	1.51	1.18	0.59	0.61	1.16	0.66
Active set	196	143	294	308	305	336	238	233	349	275
Time (s)	65	56	194	210	224	252	60	81	259	146

Table 1. Results for USPS data. Figures in PASS-GP are averages over 10 repetitions

where we have used the marginalization property of GPs, $p(\mathbf{y}_A|\mathbf{X}) = \int p(\mathbf{y}_A|\mathbf{f}_A)p(\mathbf{f}_A|\mathbf{X}_A)d\mathbf{f}_A = p(\mathbf{y}_A|\mathbf{X}_A) \equiv Z_{\mathrm{EP},A}$. We identify the last term with the marginal likelihood for the active set. The conditional marginal likelihood term can be written as

$$p(\mathbf{y}_I|\mathbf{y}_A, \mathbf{X}_A, \mathbf{X}_I) = \int p(\mathbf{y}_I|\mathbf{f}_I)p(\mathbf{f}_I|\mathbf{X}_I, \mathbf{f}_A)p(\mathbf{f}_A|\mathbf{X}_A, \mathbf{y}_A)\, d\mathbf{f}_A d\mathbf{f}_I \,, \quad (5)$$

where we have used $p(\mathbf{f}|\mathbf{X}) = p(\mathbf{f}_I|\mathbf{X}_I, \mathbf{f}_A)p(\mathbf{f}_A|\mathbf{X}_A)$. We can make an EP approximation here just like in eq. (1) by replacing the posterior $p(\mathbf{f}_A|\mathbf{X}_A, \mathbf{y}_A)$ by the multivariate Gaussian $q(\mathbf{f}_A|\mathbf{X}_A, \mathbf{y}_A) = \mathcal{N}(\mathbf{f}_A|\mathbf{m}_A, \mathbf{C}_{AA})$ where means and variances are found by EP. Marginalizing over $\mathbf{f}_A$ in eq. (5) makes it now tractable

$$q(\mathbf{y}_I|\mathbf{y}_A, \mathbf{X}_A, \mathbf{X}_I) \approx \int p(\mathbf{y}_I|\mathbf{f}_I)\mathcal{N}(\mathbf{f}_I|\mathbf{m}_{I|A}, \mathbf{C}_{II|A})d\mathbf{f}_I \,,$$

with parameters $\mathbf{m}_{I|A} = \mathbf{K}_{IA}(\mathbf{K}_{AA} + \widetilde{\mathbf{C}}_{AA})^{-1}\widetilde{\mathbf{m}}_A$ and $\mathbf{C}_{II|A} = \mathbf{K}_{II} - \mathbf{K}_{IA}(\mathbf{K}_{AA} + \widetilde{\mathbf{C}}_{AA})^{-1}\mathbf{K}_{AI}$. where the tilted 'moments' are defined in Section 2. When the inactive set consists of one example, we get the EP predictive distribution eq. (3), and otherwise we have to solve for a new marginal likelihood. Denoting the marginal likelihood for a set $(\mathbf{X}, \mathbf{y})$ with a non-zero mean GP prior by $Z(\mathbf{y}, \mathbf{m}, \mathbf{K}) = \int p(\mathbf{y}|\mathbf{f})\mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{K})\, d\mathbf{f}$ and its corresponding EP approximation by $Z_{\mathrm{EP}}(\mathbf{y}, \mathbf{m}, \mathbf{K})$ we can write the approximation to the marginal likelihood in eq. (4) as

$$Z_{\mathrm{ACC}} \equiv Z_{\mathrm{EP}}(\mathbf{y}_I, \mathbf{m}_{I|A}, \mathbf{C}_{II|A})Z_{\mathrm{EP}}(\mathbf{y}_A, 0, \mathbf{K}_{AA}) \,. \quad (6)$$

Using this approximate decomposition reduces the complexity of EP from $\mathcal{O}(N^3 N_{\mathrm{pass}})$ to $\mathcal{O}(N^3 + (|I|^3 + |A|^3)N_{\mathrm{pass}})$, unfortunately this is still too costly for large N.

We end this section with a few more qualitative comments that we will follow up upon in the empirical work. Since I contains the well-classified patterns with predictive probability close to one, the marginal likelihood per example will be much smaller for the $I|A$-term than for the A-term. The values of the hyperparameters (length scales, etc.) will to a very large degree be determined by the active set examples lying close to the decision boundary. Finally the product of marginals will be a lower bound to the marginal likelihood: $p(y_I|y_A, \mathbf{X}) > \prod_{i\in I} P(y_i|\mathbf{y}_A, \mathbf{X}_A, \mathbf{x}_i)$ because the easy well separated patterns in I will enforce each other. Hence using this lower bound we can compute a cheap approximation to $q(\mathbf{y}|\mathbf{X})$, denoted by Z_{APP}, which we illustrate in the next section (see Figure 1).

6. EXPERIMENTS

The USPS digits database contains 9289 grayscale images of size 16×16 pixels, scaled and translated to fall within the range from -1 to 1. Here we are using the traditional data splitting, i.e. 7291 observations for training and the remaining 2007 for testing. For each binary one-against-rest classifier we are using the same model setup consisting on a squared exponential covariance matrix plus an additive bias to account for the imbalance of the task, this means that we have three hyperparameters in total, namely signal variance, characteristic length scale and bias. The settings used for the algorithm were $N_{\mathrm{init}} = 300$, $N_{\mathrm{sub}} = 10$, $N_{\mathrm{pass}} = 2$, $p_{inc} = 0.6$ and $p_{del} = 0.99$[1]. Since the initial set and the partitions are chosen at random from the whole training set, each task was repeated 10 times[2]. In table 1 are shown the results by means of average test error, active set size and running time. In addition, we show the results of the GP built using the entire training set with hyperparameter optimization[3] (GP+hopt) and without it (GP+hPASS-GP). In the latter using the hyperparameters obtained with PASS-GP.

The results obtained on USPS show that PASS-GP is performing slightly better than GP+hopt and GP+hPASS-GP. This could be due to numerical instability produced by the size of the problem, by the iterative nature of the EP algorithm and/or not enough iterations for the model selection procedure. However, it could also mean that optimizing on the active achieves a better 'local' fit around the decision boundary region. A priori one cannot expect that one set of hyperparameters are able to describe all regions in input space and that might be what we see here.

[1] All the experiments were run on a dual core AMD Opteron 1GHz processor with 2GB RAM.

[2] More detailed tables including variances can obtained upon request.

[3] For this purpose we used the code provided with [1] limiting the model selection to 20 conjugate gradient iterations.

Digit	0	1	2	3	4	5	6	7	8	9
Error (%)	0.18	0.15	0.38	0.32	0.31	0.30	0.27	0.43	0.45	0.57
Active set	790	782	1545	1776	1354	1584	1028	1376	2026	2102
Time (s)	830	1585	4214	4627	2833	3223	1509	3041	6279	6321

Table 2. Results for MNIST data. Figures are averages over 10 repetitions

Table 1 also shows that PASS-GP has a highly sparse representation and running times several magnitude orders below the full GP. Combining the ten binary classifiers in a one-against-rest setting, PASS-GP obtained $4.51 \pm 0.17\%$ which is significantly better than[4], 5.13% by GP+hopt, 4.78% by GP+hPASS-GP, 5.15% by online GP [8], 4.98% by IVM [6] and comparable with state-of the-art techniques such as SVM, see [9].

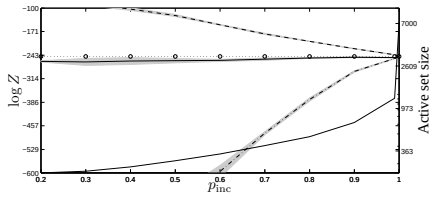


Fig. 1. Marginal log-likelihood approximations as a function of the inclusion threshold p_{inc} for 2s against non-2s. The left axis is for the (log) marginal likelihood approximations: Z_{EP} ($\circ \cdot \circ$), Z_{ACC} (—), Z_{APP} ($- \cdot -$) and $Z_{\mathrm{EP},A}$ (– –). The right axis and the increasing solid line correspond to the active set size, $|A|$. Light shades represent variances across repetitions.

Next, we want to try the approximations proposed in Section 5. For this purpose, Figure 1 shows the behavior of the marginal approximation and the active set size for different values of p_{inc} from 0.2 to 1. From the figure it is clear that PASS-GP is able to approximate the marginal likelihood of the full GP up to some small error using Z_{ACC} and that the same does not hold for Z_{APP}. However, as $p_{\mathrm{inc}} \to 1$ both approximations converge to Z_{EP}. The approximation error made using eq. (6) is small and decreasing with p_{inc}. It is very interesting that even with large values of $p_{\mathrm{inc}} = 0.99$ the size of the active set is still below 10% of the training data and contribution to the log marginal likelihood from the inactive set basically vanishes.

The MNIST digits database has 60000 and 10000 as training and testing examples respectively. Each example is a gray-scale image of 28×28 pixels. The estimated human test error is around 0.2%. The settings used for the algorithm are nearly the same as those for USPS with only two differences. N_{sub} is set 100 since the training set in MNIST is almost ten times larger than USPS and we are not updating the hyperparameter in each iteration but every 10-th, in order to make the training process faster. We also ran our algorithm with hyperparameter updates every single iteration without any noticeable improvement in performance (results not shown). Table 2 shows the test error rates, active set sizes and running times for each binary classifier. The result obtained on the multi-class task was $1.38 \pm 0.06\%$.

From the table can be seen that in every case the size of the active set is less than 5% of the training set and that the running times are not in any case greater than 2 hours which sounds reasonable considering that we are performing model selection. As far as the authors know this are the first GP results on MNIST using the whole database. IVM [4] has with sub-sampled images of size 13×13 been tested to produce a test error rate of $1.54 \pm 0.04\%$. Seeger [6] made additional tests on some digits (5, 8 and 9) on the full size images without any further improvement. On the other hand, PASS-GP is again comparable with state-of-the-art techniques not including preprocessing stages and/or data augmentation, for example SVM is 1.4% and 1.22% using RBF and a 9 degree polynomial kernel, respectively. The reported sizes of support vector sets are approximately two times bigger than our active sets [10]. We have repeated our experiment using the polynomial kernel from above to obtain $1.31 \pm 0.06\%$.

Incorporating Invariances. It has been shown that a good way to improve the overall performance of a classifier is by incorporating prior knowledge in the training procedure by means of externally handling invariances of the data. In [10], it is shown that instead of just dealing with the invariances by augmenting the original dataset—which turns out to be unfeasible in many cases—it is better to augment only the support vector set. We therefore try the same procedure as suggested in [10], i.e. four 1-pixel translations (left, up, right and down directions) on each element of the active set for USPS and eight 1-pixel translations (including diagonals as well) for MNIST, resulting in new training sets of size $5 \times |A|$ and $9 \times |A|$ respectively. For this experiment we have used fairly the same settings of the previous experiments but this time keeping the hyperparameters to those values found on the original training set. We made the important observation that in order to get a performance improvement a large active set was needed. For training on the augmented data we increased p_{inc} from 0.6

[4] Assuming independent errors the standard deviation on the performance is $\sqrt{\epsilon(1-\epsilon)/N_{\mathrm{test}}}$ giving approximately 0.4% for USPS and 0.1% for MNIST

Digit	0	1	2	3	4	5	6	7	8	9
USPS (%)	0.63	0.38	1.01	0.69	0.93	1.16	0.51	0.37	0.59	0.65
Active set	870	442	1251	1316	1654	1425	1242	987	1532	1281
MNIST (%)	0.14	0.14	0.24	0.24	0.29	0.22	0.17	0.35	0.29	0.35
Active set	6505	4372	11401	12988	9776	11960	7360	9872	15194	14790

Table 3. Results after including translation invariances over. Figures are averages over 10 and 5 repetitions in each case

to 0.99 for USPS and 0.9 for MNIST. We speculate that we can get even better performance—at the expense of a substantial increase in complexity—by increasing p_{inc} in the initial run to get a larger initial active set to work with. Results in table 3 shows that in test error rate terms, PASS-GP obtained $3.35 \pm 0.03\%$ for USPS and $0.86 \pm 0.02\%$ for MINST on the multi-class task, being comparable to state-of-the art-techiques, e.g. SVM obtained 3.2% on USPS and 0.68% on MNIST with the same procedure. The difference in performance is probably due to our active set not being big enough. The sizes reported for SVMs [10] are typically twice as large.

7. DISCUSSION

We have proposed a new sparse approximation algorithm for Gaussian processes (GP) called Predictive Active Set Selection GP (PASS-GP). It is an active set procedure where the predictive probability is used to rank data points for inclusion/deletions. We have presented theoretical and practical support that our active set selection strategy is efficient while retaining the benefits of GP: error bars, model selection, prior knowledge integration and state-of-the-art performance. Compared to other approximative methods for GP, PASS-GP should be slower than methods which are more online in nature [8, 4], but faster than FITC approximations [2, 3]. Another appealing feature of PASS-GP is that all of the parameters of the algorithm are provided with a clear interpretation making the setup process straightforward. Additionally, we have noticed in practice that our approximation is quite insensitive to the initial active set selection and also that more than two or three passes through the data do not yield improved performance nor large active set sizes. The code used in this work is based on the Matlab toolbox provided with [1] and will be publicly available.

The not so satisfying feature of active set approximations, is that we are ignoring some of the training data. Although some of our findings on the USPS data set actually suggest that this can be beneficial for performance, it is of interest to make a modified version where the inactive set is used approximately in a cost efficient way. The representer theorem for the mean prediction and the approximations for marginal likelihood discussed in this paper might give inspiration for such methods. In conclusion, efficient methods for GPs are still much in need when the data abundant such ordinal regression for collaborative filtering.

8. REFERENCES

[1] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, Cambridge, MA, 2006.

[2] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, December 2005.

[3] A. Naish-Guzman and S. Holden, "The generalized FITC approximation," in *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., pp. 1057–1064. MIT Press, Cambridge, MA, 2008.

[4] N. D. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse Gaussian process methods: The informative vector machine," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds., pp. 600–616. The MIT Press, Cambridge, MA, 2003.

[5] M. Kuss and C. E. Rasmussen, "Assessing approximate inference for binary Gaussian process classification," *Journal of Machine Learning Research*, vol. 6, pp. 1679–1704, October 2005.

[6] M. Seeger, *Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations*, Ph.D. thesis, University of Edinburg, December 2003.

[7] M. Opper and O. Winther, "Gaussian processes for classification: Mean-field algorithms," *Neural Computation*, vol. 12, no. 11, pp. 2655–2684, 2000.

[8] L. Csató, *Gaussian Processes - Iterative Sparse Approximations*, Ph.D. thesis, Aston University, March 2002.

[9] B. Schölkopf and A. J. Smola, *Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond*, The MIT Press, 2001.

[10] D. DeCoste and B. Schölkopf, "Training invariant support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 161–190, January 2002.

Appendix C

Sparse Linear Identifiable Multivariate Modeling

To appear
Journal of Machine Learning Research

Available from Arxiv at
`http://arxiv.org/abs/1004.5265`

Complementary website at
`http://cogsys.imm.dtu.dk/slim`

Journal of Machine Learning Research xx (2011) 1-45 Submitted 11/09; Revised 11/10; Published xx/11

Sparse Linear Identifiable Multivariate Modeling

Ricardo Henao RHENAO@BINF.KU.DK
Ole Winther OWI@IMM.DTU.DK
DTU Informatics
Richard Petersens Plads, Building 321
Technical University of Denmark
DK-2800 Lyngby, Denmark

Bioinformatics Centre
University of Copenhagen
Ole Maaloes Vej 5
DK-2200 Copenhagen N, Denmark

Editor: Aapo Hyvärinen

Abstract

In this paper we consider sparse and identifiable linear latent variable (factor) and linear Bayesian network models for parsimonious analysis of multivariate data. We propose a computationally efficient method for joint parameter and model inference, and model comparison. It consists of a fully Bayesian hierarchy for sparse models using slab and spike priors (two-component δ-function and continuous mixtures), non-Gaussian latent factors and a stochastic search over the ordering of the variables. The framework, which we call SLIM (Sparse Linear Identifiable Multivariate modeling), is validated and bench-marked on artificial and real biological data sets. SLIM is closest in spirit to LiNGAM (Shimizu et al., 2006), but differs substantially in inference, Bayesian network structure learning and model comparison. Experimentally, SLIM performs equally well or better than LiNGAM with comparable computational complexity. We attribute this mainly to the stochastic search strategy used, and to parsimony (sparsity and identifiability), which is an explicit part of the model. We propose two extensions to the basic i.i.d. linear framework: non-linear dependence on observed variables, called SNIM (Sparse Non-linear Identifiable Multivariate modeling) and allowing for correlations between latent variables, called CSLIM (Correlated SLIM), for the temporal and/or spatial data. The source code and scripts are available from `http://cogsys.imm.dtu.dk/slim/`.

Keywords: Parsimony, sparsity, identifiability, factor models, linear Bayesian networks

1. Introduction

Modeling and interpretation of multivariate data are central themes in machine learning. Linear latent variable models (or factor analysis) and linear directed acyclic graphs (DAGs) are prominent examples of models for continuous multivariate data. In factor analysis, data is modeled as a linear combination of independently distributed factors thus allowing for capture of a rich underlying co-variation structure. In the DAG model, each variable is expressed as regression on a subset of the remaining variables with the constraint that total connectivity is acyclic in order to have a properly defined joint distribution. Parsimonious (interpretable) modeling, using sparse factor loading matrix or restricting the number of

parents of a node in a DAG, are good prior assumptions in many applications. Recently, there has been a great deal of interest in detailed modeling of sparsity in factor models, for example in the context of gene expression data analysis (West, 2003, Lucas et al., 2006, Knowles and Ghahramani, 2007, Thibaux and Jordan, 2007, Carvalho et al., 2008, Rai and Daume III, 2009). Sparsity arises for example in gene regulation because the latent factors represent driving signals for gene regulatory sub-networks and/or transcription factors, each of which only includes/affects a limited number of genes. A parsimonious DAG is particularly attractable from an interpretation point of view but the restriction to only having observed variables in the model may be a limitation because one rarely measures all relevant variables. Furthermore, linear relationships might be unrealistic for example in gene regulation, where it is generally accepted that one cannot replace the driving signal (related to concentration of a transcription factor protein in the cell nucleus) with the measured concentration of corresponding mRNA. Bayesian networks represent a very general class of models, encompassing both observed and latent variables. In many situations it will thus be relevant to learn parsimonious Bayesian networks with both latent variables and a non-linear DAG parts. Although attractive, by being closer to what one may expect in practice, such modeling is complicated by difficult inference (Chickering (1996) showed that DAG structure learning is NP-hard) and by potential non-identifiability. Identifiability means that each setting of the parameters defines a unique distribution of the data. Clearly, if the model is not identifiable in the DAG and latent parameters, this severely limits the interpretability of the learned model.

Shimizu et al. (2006) provided the important insight that every DAG has a factor model representation, i.e. the connectivity matrix of a DAG gives rise to a triangular mixing matrix in the factor model. This provided the motivation for the Linear Non-Gaussian Acyclic Model (LiNGAM) algorithm which solves the identifiable factor model using Independent Component Analysis (ICA, Hyvärinen et al., 2001) followed by iterative permutation of the solutions towards triangular, aiming to find a suitable ordering for the variables. As final step, the resulting DAG is pruned based on different statistics, e.g. Wald, Bonferroni, χ^2 second order model fit tests. Model selection is then performed using some pre-chosen significance level, thus LiNGAM select from models with different sparsity levels and a fixed deterministically found ordering. There is a possible number of extensions to their basic model, for instance Hoyer et al. (2008) extend it to allow for latent variables, for which they use a probabilistic version of ICA to obtain the variable ordering, pruning to make the model sparse and bootstrapping for model selection. Although the model seems to work well in practice, as commented by the authors, it is restricted to very small problems (3 or 4 observed and 1 latent variables). Non-linear DAGs are also a possibility, however finding variable orderings in this case is known to be far more difficult than in the linear case. These methods inspired by Friedman and Nachman (2000), mainly consist of two steps: performing non-linear regression for a set of possible orderings, and then testing for independence to prune the model, see for instance Hoyer et al. (2009) and Zhang and Hyvärinen (2010). For tasks where exhaustive order enumeration is not feasible, greedy approaches like DAG-search (see "ideal parent" algorithm, Elidan et al., 2007) or PC (Prototypical Constraint, see kernel PC, Tillman et al., 2009) can be used as computationally affordable alternatives.

Factor models have been successfully employed as exploratory tools in many multivariate analysis applications. However, interpretability using sparsity is usually not part of

the model, but achieved through post-processing. Examples of this include, bootstrapping, rotating the solutions to maximize sparsity (varimax, procrustes), pruning or thresholding. Another possibility is to impose sparsity in the model through L_1 regularization to obtain a maximum a-posteriori estimate (Jolliffe et al., 2003, Zou et al., 2006). In fully Bayesian sparse factor modeling, two approaches have been proposed: parametric models with bimodal sparsity promoting priors (West, 2003, Lucas et al., 2006, Carvalho et al., 2008, Henao and Winther, 2009), and non-parametric models where the number of factors is potentially infinite (Knowles and Ghahramani, 2007, Thibaux and Jordan, 2007, Rai and Daume III, 2009). It turns out that most of the parametric sparse factor models can be seen as finite versions of their non-parametric counterparts, for instance West (2003) and Knowles and Ghahramani (2007). The model proposed by West (2003) is, as far as the authors know, the first attempt to encode sparsity in a factor model explicitly in the form of a prior. The remaining models improve the initial setting by dealing with the optimal number of factors in Knowles and Ghahramani (2007), improved hierarchical specification of the sparsity prior in Lucas et al. (2006), Carvalho et al. (2008), Thibaux and Jordan (2007), hierarchical structure for the loading matrices in Rai and Daume III (2009) and identifiability without restricting the model in Henao and Winther (2009).

Many algorithms have been proposed to deal with the NP-hard DAG structure learning task. LiNGAM, discussed above, is the first fully identifiable approach for continuous data. All other approaches for continuous data use linearity and (at least implicitly) Gaussianity assumptions so that the model structure learned is only defined up to equivalence classes. Thus in most cases the directionality information about the edges in the graph must be discarded. Linear Gaussian-based models have the added advantage that they are computationally affordable for the many variables case. The structure learning approaches can be roughly divided into stochastic search and score (Cooper and Herskovits, 1992, Heckerman et al., 2000, Friedman and Koller, 2003), constraint-based (with conditional independence tests) (Spirtes et al., 2001) and two stage; like LiNGAM, (Tsamardinos et al., 2006, Friedman et al., 1999, Teyssier and Koller, 2005, Schmidt et al., 2007, Shimizu et al., 2006). In the following, we discuss in more detail previous work in the last category, as it is closest to the work in this paper and can be considered representative of the state-of-the-art. The Max-Min Hill-Climbing algorithm (MMHC, Tsamardinos et al., 2006) first learns the skeleton using conditional independence tests similar to PC algorithms (Spirtes et al., 2001) and then the order of the variables is found using a Bayesian-scoring hill-climbing search. The Sparse Candidate (SC) algorithm (Friedman et al., 1999) is in the same spirit but restricts the skeleton to within a predetermined link candidate set of bounded size for each variable. The Order Search algorithm (Teyssier and Koller, 2005) uses hill-climbing first to find the ordering, and then looks for the skeleton with SC. L_1 regularized Markov Blanket (Schmidt et al., 2007) replaces the skeleton learning from MMHC with a dependency network (Heckerman et al., 2000) written as a set of local conditional distributions represented as regularized linear regressors. Since the source of identifiability in Gaussian DAG models is the direction of the edges in the graph, a still meaningful approach consists of entirely focusing on inferring the skeleton of the graph by keeping the edges undirected as in Dempster (1972), Dawid and Lauritzen (1993), Giudici and Green (1999), Rajaratman et al. (2008).

In this paper we propose a framework called SLIM (Sparse Linear Identifiable Multivariate modeling, see Figure 1) in which we learn models from a rather general class of Bayesian

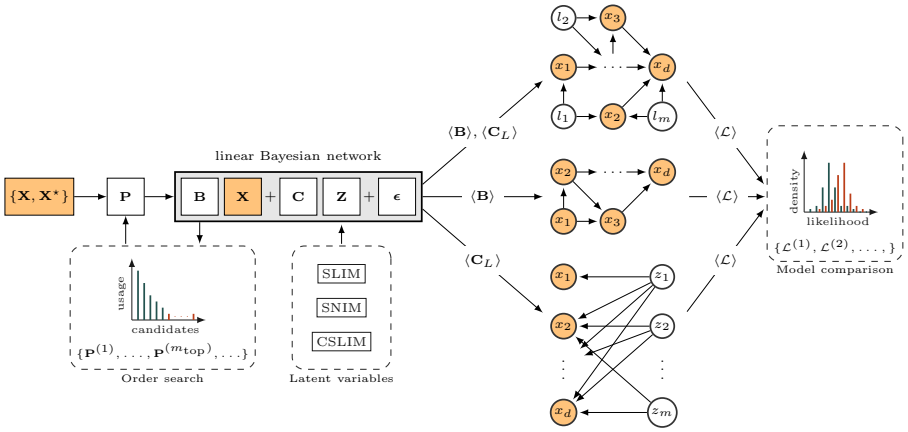


Figure 1: SLIM in a nutshell. Starting from a training-test set partition of data $\{\mathbf{X}, \mathbf{X}^\star\}$, our framework produces factor models $\mathbf{C}$ and DAG candidates $\mathbf{B}$ with and without latent variables $\mathbf{Z}$ that can be compared in terms of how well they fit the data using test likelihoods $\mathcal{L}$. The variable ordering $\mathbf{P}$ needed by the DAG is obtained as a byproduct of a factor model inference. Besides, changing the prior over latent variables $\mathbf{Z}$ produces two variants of SLIM called CSLIM and SNIM.

networks and perform quantitative model comparison between them[1]. Model comparison may be used for model selection or serve as a hypothesis-generating tool. We use the likelihood on a test set as a computationally simple quantitative proxy for model comparison and as an alternative to the marginal likelihood. The other two key ingredients in the framework are the use of sparse and identifiable model components (Carvalho et al., 2008, Kagan et al., 1973, respectively) and the stochastic search for the correct order of the variables needed by the DAG representation. Like LiNGAM, SLIM exploits the close relationship between factor models and DAGs. However, since we are interested in the factor model by itself, we will not constrain the factor loading matrix to have triangular form, but allow for sparse solutions so pruning is not needed. Rather we may ask whether there exists a permutation of the factor-loading matrix agreeing to the DAG assumption (in a probabilistic sense). The slab and spike prior biases towards sparsity so it makes sense to search for a permutation in parallel with factor model inference. We propose to use stochastic updates for the permutation using a Metropolis-Hastings acceptance ratio based on likelihoods with the factor-loading matrix being masked. In practice this approach gives good solutions up to at least fifty dimensions. Given a set of possible variable orderings inferred by this method, we can then learn DAGs using slab and spike priors for their connectivity matrices. The so-called slab and spike prior is a two-component mixture of a continuous distribution and degenerate δ-function point mass at zero. This type of model implicitly defines a prior over

1. A preliminary version of our approach appears in NIPS 2009: Henao and Winther, Bayesian sparse factor models and DAGs inference and comparison.

structures and is thus a computationally attractive alternative to combinatorial structure search since parameter and structure inference are performed simultaneously. A key to effective learning in these intractable models is Markov Chain Monte Carlo (MCMC) sampling schemes that mix well. For non-Gaussian heavy-tailed distributions like the Laplace and t-distributions, Gibbs sampling can be efficiently defined using appropriate infinite scale mixture representations of these distributions (Andrews and Mallows, 1974). We also show that our model is very flexible in the sense that it can be easily extended by only changing the prior distribution of a set of latent variables, for instance to allow for time series data (CSLIM, Correlated SLIM) and non-linearities in the DAG structure (SNIM, Sparse non-Linear Identifiable Multivariate modeling) through Gaussian process priors.

The rest of the paper is organized as follows: Section 2 describes the model and its identifiability properties. Section 3 provides all prior specification including sparsity, latent variables and driving signals, order search and extensions for correlated data (CSLIM) and non-linearities (SNIM). Section 4 elaborates on model comparison. Section 5 and Appendix A provide an overview of the model and practical details on the MCMC-based inference, proposed workflow and computational cost requirements. Section 6 contains the experiments. We show simulations based on artificial data to illustrate all the features of the model proposed. Real biological data experiments illustrate the advantages of considering different variants of Bayesian networks. For all data sets we compare with some of the most relevant existing methods. Section 7 concludes with a discussion, open questions and future directions.

2. Linear Bayesian networks

A Bayesian network is essentially a joint probability distribution defined via a directed acyclic graph, where each node in the graph represents a random variable x. Due to the acyclic property of the graph, its node set $x_1, \ldots, x_d$ can be partitioned into d subsets $\{V_1, V_2, \ldots, V_d\} \equiv \mathcal{V}$, such that if $x_j \to x_i$ then $x_j \in V_i$, i.e. V_i contains all *parents* of x_i. We can then write the joint distribution as a product of conditionals of the form

$$P(x_1, \ldots, x_d) = \prod_{i=1}^{d} P(x_i|V_i) \ ,$$

thus x_i is conditionally independent of $\{x_j|x_i \notin V_j\}$ given V_i for $i \neq j$. This means that $p(x_1, \ldots, x_d)$ can be used to describe the joint probability of any set of variables once $\mathcal{V}$ is given. The problem is that $\mathcal{V}$ is usually unknown and thus needs to be (at least partially) inferred from observed data.

We consider a model for a fairly general class of linear Bayesian networks by putting together a linear DAG, $\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{z}$, and a factor model, $\mathbf{x} = \mathbf{C}\mathbf{z} + \boldsymbol{\epsilon}$. Our goal is to explain each one of d observed variables $\mathbf{x}$ as a linear combination of the remaining ones, a set of $d + m$ independent latent variables $\mathbf{z}$ and additive noise $\boldsymbol{\epsilon}$. We have then

$$\mathbf{x} = (\mathbf{R} \odot \mathbf{B})\mathbf{x} + (\mathbf{Q} \odot \mathbf{C})\mathbf{z} + \boldsymbol{\epsilon} \ , \tag{1}$$

where $\odot$ is the element-wise product and we can further identify the following elements:

- $\mathbf{z}$ is partitioned into two subsets, $\mathbf{z}_D$ is a set of d driving signals for each observed variable in $\mathbf{x}$ and $\mathbf{z}_L$ is a set of m shared general purpose latent variables. $\mathbf{z}_D$ is used here to describe the intrinsic behavior of the observed variables that cannot regarded as "external" noise.

- $\mathbf{R}$ is a $d \times d$ binary connectivity matrix that encodes whether there is an edge between observed variables, by means of $r_{ij} = 1$ if $x_i \to x_j$. Since every non-zero element in $\mathbf{R}$ is an edge of a DAG, $r_{ii} = 0$ and $r_{ij} = 0$ if $r_{ji} \neq 0$ to avoid self-interactions and bi-directional edges, respectively. This also implies that there is at least one permutation matrix $\mathbf{P}$ such that $\mathbf{P}^\top \mathbf{R} \mathbf{P}$ is strictly lower triangular where we have used that $\mathbf{P}$ is orthonormal then $\mathbf{P}^{-1} = \mathbf{P}^\top$.

- $\mathbf{Q} = [\mathbf{Q}_D \; \mathbf{Q}_L]$ is a $d \times (d + m)$ binary connectivity matrix, this time for the conditional independence relations between observed and latent variables. We assume that each observed variable has a dedicated latent variable, thus the first d columns of $\mathbf{Q}_D$ are the identity. The remaining m columns can be arbitrarily specified, by means of $q_{ij} \neq 0$ if there is an edge between x_i and z_j for $d < j \leq m$.

- $\mathbf{B}$ and $\mathbf{C} = [\mathbf{C}_L \; \mathbf{C}_D]$ are respectively, $d \times d$ and $d \times (d+m)$ weight matrices containing the edge strengths for the Bayesian network. Their elements are constrained to be non-zero only if their corresponding connectivities are also non-zero.

The model (1) has two important special cases, (i) if all elements in $\mathbf{R}$ and $\mathbf{Q}_D$ are zero it becomes a standard factor model (FM) and (ii) if $m = 0$ or all elements in $\mathbf{Q}_L$ are zero it is a pure DAG. The model is not a completely general linear Bayesian network because connections to latent variables are absent (see for example Silva, 2010). However, this restriction is mainly introduced to avoid compromising the identifiability of the model. In the following we will only write $\mathbf{Q}$ and $\mathbf{R}$ explicitly when we specify the sparsity modeling.

2.1 Identifiability

We will split the identifiability of the model in equation (1) in three parts addressing first the factor model, second the pure DAG and finally the full model. By identifiability we mean that each different setting of the parameters $\mathbf{B}$ and $\mathbf{C}$ gives a unique distribution of the data. In some cases the model is only unique up to some symmetry of the model. We discuss these symmetries and their effect on model interpretation in the following.

Identifiability in factor models $\mathbf{x} = \mathbf{C}_L \mathbf{z}_L + \boldsymbol{\epsilon}$ can be obtained in a number of ways (see Chapter 10, Kagan et al., 1973). Probably the easiest way is to assume sparsity in $\mathbf{C}_L$ and restrict its number of free parameters, for example by restricting the dimensionality of $\mathbf{z}$, namely m, according to the Ledermann bound $m \leq (2d + 1 - (8d + 1)^{1/2})/2$ (Bekker and ten Berge, 1997). The Ledermann bound guarantees the identification of $\boldsymbol{\epsilon}$ and follows just from counting the number of free parameters in the covariance matrices of $\mathbf{x}$, $\boldsymbol{\epsilon}$ and in $\mathbf{C}_L$, assuming Gaussianity of $\mathbf{z}$ and $\boldsymbol{\epsilon}$. Alternatively, identifiability is achieved using non-Gaussian distributions for $\mathbf{z}$. Kagan et al. (Theorem 10.4.1, 1973) states that when at least $m - 1$ latent variables are non-Gaussian, $\mathbf{C}_L$ is identifiable up to scale and permutation of its columns, i.e. we can identify $\widehat{\mathbf{C}}_L = \mathbf{C}_L \mathbf{S}_\mathrm{f} \mathbf{P}_\mathrm{f}$, where $\mathbf{S}_\mathrm{f}$ and $\mathbf{P}_\mathrm{f}$ are arbitrary scaling and permutation matrices, respectively. Comon (1994) provided an alternative

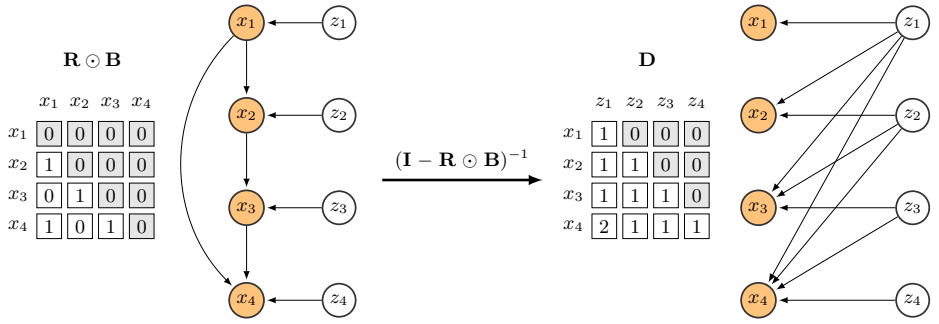


Figure 2: FM-DAG equivalence illustration. In the left side, a DAG model with four variables with corresponding connectivity matrix $\mathbf{R}$, $b_{ij} = 1$ when $r_{ij} = 1$ and $\mathbf{C}_D = \mathbf{I}$. In the right hand side, the equivalent factor model with mixing matrix $\mathbf{D}$. Note that the factor model is sparse even if its corresponding DAG is dense. The gray boxes in $\mathbf{D}$ and $\mathbf{R} \odot \mathbf{B}$ represent elements that must be zero by construction.

well-known proof for the particular case of $m - 1 = d$. The $\mathbf{S}_\mathrm{f}$ and $\mathbf{P}_\mathrm{f}$ symmetries are inherent in the factor model definition in all cases and will usually not affect interpretability. However, some researchers prefer to make the model completely identifiable, e.g. by making $\mathbf{C}_L$ triangular with non-negative diagonal elements (Lopes and West, 2004). In addition, if all components of $\boldsymbol{\epsilon}$ are Gaussian and the rank of $\mathbf{C}_L$ is m, then the distributions of $\mathbf{z}$ and $\boldsymbol{\epsilon}$ are uniquely defined to within common shift in mean (Theorem 10.4.3, Kagan et al., 1973). In this paper, we use the non-Gaussian $\mathbf{z}$ option for two reasons, (i) restricting the number of latent variables severely limits the usability of the model and (ii) non-Gaussianity is a more realistic assumption in many application areas such as for example biology.

For pure DAG models $\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{C}_D\mathbf{z}_D$, identifiability can be obtained using the factor model result from Kagan et al. (1973) by rewriting the DAG into an equivalent factor model $\mathbf{x} = \mathbf{D}\mathbf{z}$ with $\mathbf{D} = (\mathbf{I} - \mathbf{B})^{-1}\mathbf{C}_D$, see Figure 2. From the factor model result it only follows that $\mathbf{D}$ is identifiable up to a scaling and permutation. However, as mentioned above, due to the acyclicity there is at least one permutation matrix $\mathbf{P}$ such that $\mathbf{P}^\top\mathbf{B}\mathbf{P}$ is strictly lower triangular. Now, if $\mathbf{x}$ admits DAG representation, the same $\mathbf{P}$ makes the permuted $\widehat{\mathbf{D}} = (\mathbf{I} - \mathbf{P}^\top\mathbf{B}\mathbf{P})^{-1}\mathbf{C}_D$, triangular with $\mathbf{C}_D$ on its diagonal. The constraint on the number of non-zero elements in $\mathbf{D}$ due to triangularity removes the permutation freedom $\mathbf{P}_\mathrm{f}$ such that we can subsequently identify $\mathbf{P}$, $\mathbf{B}$ and $\mathbf{C}_D$. It also implies that any valid permutation $\mathbf{P}$ will produce exactly the same distribution for $\mathbf{x}$.

In the general case in equation (1), $\mathbf{D} = (\mathbf{I} - \mathbf{B})^{-1}\mathbf{C}$ is of size $d \times (d + m)$. What we will show is that even if $\mathbf{D}$ is still identifiable, we can no longer obtain $\mathbf{B}$ and $\mathbf{C}$ uniquely unless we "tag" the model by requiring the distributions of driving signals $\mathbf{z}_D$ and latent signals $\mathbf{z}_L$ to differ. In order to illustrate why we get non-identifiability, we can write $\mathbf{x} = \mathbf{D}\mathbf{z}$ inverting $\mathbf{D}$ explicitly. For simplicity we consider $m = 1$ and $\mathbf{P} = \mathbf{I}$ but generalizing to $m > 1$ is straight forward

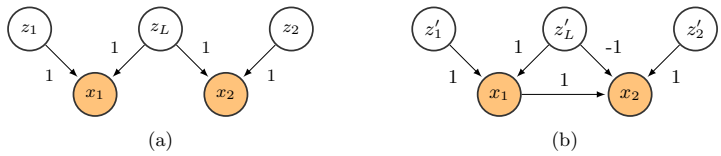


Figure 3: Two DAGs with latent variables. They are equivalent if $\mathbf{z}$ has the same distribution as $\mathbf{z}'$.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} c_{11} & 0 & 0 & \cdots & c_{1L} \\ b_{21}c_{11} & c_{22} & 0 & \cdots & b_{21}c_{1L} + c_{2L} \\ b_{31}c_{11} + b_{32}b_{21}c_{11} & b_{32}c_{22} & c_{33} & \cdots & b_{31}c_{1L} + b_{32}b_{21}c_{1L} + a_{32}c_{2L} + c_{3L} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{11} + \sum_{k=1}^{i-1} b_{ik}d_{k1} & \cdots & \cdots & \cdots & c_{iL} + \sum_{k=1}^{i-1} b_{ik}d_{kL} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_{d+1} \end{bmatrix}.$$

We see from this equation that if all latent variables have the same distribution and c_{1L} is non-zero then we may exchange the first and last column in $\mathbf{D}$ to get two equivalent distributions with different elements for $\mathbf{B}$ and $\mathbf{C}$. The model is thus non-identifiable. If the first i elements in latent column of $\mathbf{C}$ are zero then the $(i+1)$-th and last column can be exchanged. Hoyer et al. (2008) made the same basic observation through a number of examples. Interestingly, we also see from the triangularity requirement of the "driving signal" part of $\mathbf{D}$ that $\mathbf{P}$ is actually identifiable despite the fact that $\mathbf{B}$ and $\mathbf{C}$ are not. To illustrate that the non-identifiability may lead to quite severe confusion about inferences, consider a model with only two observed variables $\mathbf{x} = [x_1, x_2]^\top$ and $c_{11} = c_{22} = 1$. Two different hypothesis $\{b_{21}, c_{1L}, c_{2L}\} = \{0, 1, 1\}$ and $\{b_{21}, c_{1L}, c_{2L}\} = \{1, 1, -1\}$ with graphs shown in Figure 3 have equivalent factor models written as

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_L \end{bmatrix} \text{ and } \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} z'_1 \\ z'_2 \\ z'_L \end{bmatrix}.$$

The two models above have the same mixing matrix $\mathbf{D}$, up to permutation of columns $\mathbf{P}_\mathrm{f}$. In general we expect the number of solutions with equivalent distribution may be as large as 2^m, corresponding to the number of times a column of $\mathbf{D}$ from its latent part (last m columns) con be exchanged with a column from its observed part (first d columns). This readily assumes that the sparsity pattern in $\mathbf{D}$ is identified, which follows from the results of Kagan et al. (1973).

One way to get identifiability is to change the distributions $\mathbf{z}_D$ and $\mathbf{z}_L$ such that they differ and cannot be exchanged. Here it is not enough to change the scale of the variables, i.e. variance for continuous variables, because this effect can be countered by rescaling $\mathbf{C}$ with $\mathbf{S}_\mathrm{f}$. So we need distributions that differ beyond rescaling. In our examples we use Laplace and the more heavy-tailed Cauchy for $\mathbf{z}_D$ and $\mathbf{z}_L$, respectively. This specification is not unproblematic in practical situations however it can be sometimes restrictive and prone to model mismatch issues. We nevertheless show one practical example which leads to sensible inferences.

In time series applications for example, it is natural to go beyond an i.i.d. model for $\mathbf{z}$. One may for example use a Gaussian process prior for each factor to get smoothness over time, i.e. $z_{j1}, \ldots, z_{jN}|\nu_j \sim \mathcal{N}(0, \mathbf{K}_{\nu_j})$, where $\mathbf{K}_{\nu_j}$ is the covariance matrix with elements $k_{j,nn'} = k_{v_j,n}(n, n')$ and $k_{v_j,n}(\cdot)$ is the covariance function. For the i.i.d. Gaussian model the source distribution is only identifiable up to an arbitrary rotation matrix $\mathbf{U}$, i.e. the rotated factors $\mathbf{Uz}$ are still i.i.d. . We can show that contrary to the i.i.d. Gaussian model, the Gaussian process factor model is identifiable if the covariance functions differ. We need to show that $\widehat{\mathbf{Z}} = \mathbf{UZ}$ has a different covariance structure than $\mathbf{Z} = [\mathbf{z}_1 \ \ldots \ \mathbf{z}_N]$. We get $\overline{\mathbf{z}_n \mathbf{z}_{n'}^\top} = \mathrm{diag}(k_{1,nn'}, \ldots, k_{d+m,nn'})$ and $\overline{\widehat{\mathbf{z}}_n \widehat{\mathbf{z}}_{n'}^\top} = \mathbf{U}\overline{\mathbf{z}_n \mathbf{z}_{n'}^\top}\mathbf{U}^\top = \mathbf{U}\mathrm{diag}(k_{1,nn'}, \ldots, k_{d+m,nn'})\mathbf{U}^\top$ for the original and rotated variables, respectively. The covariances are indeed different and the model is thus identifiable if no covariance functions $k_{v_j,n}(n, n')$, $j = 1, \ldots, d+m$ are the same.

3. Prior specification

In this section we provide a detailed description of the priors used for each one of the elements of our sparse linear identifiable model already defined in equation (1). We start with $\boldsymbol{\epsilon}$, the noise term that allow us to quantify the mismatch between a set of N observations $\mathbf{X} = [\mathbf{x}_1 \ \ldots \ \mathbf{x}_N]$ and the model itself. For this purpose, we use uncorrelated Gaussian noise components $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \boldsymbol{\Psi})$ with conjugate inverse gamma priors for their variances as follows

$$\mathbf{X}|\mathbf{m}, \boldsymbol{\Psi} \sim \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\mathbf{m}, \boldsymbol{\Psi}) \ ,$$

$$\boldsymbol{\Psi}^{-1}|s_s, s_r \sim \prod_{i=1}^{d} \mathrm{Gamma}(\psi_i^{-1}|s_s, s_r) \ ,$$

where we have already marginalized out $\boldsymbol{\epsilon}$, $\boldsymbol{\Psi}$ is a diagonal covariance matrix denoting uncorrelated noise across dimensions and $\mathbf{m}$ is the mean vector such that $\mathbf{m}_{\mathrm{FM}} = \mathbf{C}\mathbf{z}_n$ and $\mathbf{m}_{\mathrm{DAG}} = \mathbf{B}\mathbf{x}_n + \mathbf{C}\mathbf{z}_n$. In the noise covariance hyperprior, s_s and s_r are the shape and rate, respectively. The selection of hyperparameters for $\boldsymbol{\Psi}$ should not be very critical as long as both "signal and noise" hypotheses are supported, i.e. diffuse enough to allow for small values of ψ_i as well as for $\psi_i = 1$ (assuming that the data is standardized in advance). We set $s_s = 20$ and $s_r = 1$ in the experiments for instance. Another issue to consider when selecting s_s and s_r is the Bayesian analogue of the Heywood problem in which likelihood functions are bounded below away from zero as ψ_i tends to zero, hence inducing multimodality in the posterior of ψ_i with one of the modes at zero. The latter can be avoided by specifying s_s and s_r such that the prior decays to zero at the origin, as we did above. It is well known, for example, that Heywood problems cannot be avoided using improper reference priors, $p(\psi_i) \propto 1/\psi_i$ (Martin and McDonald, 1975).

The remaining components of the model are described as it follows in five parts named sparsity, latent variables and driving signals, order search, allowing for correlated data and allowing for non-linearities. The first part addresses the interpretability of the model by means of parsimonious priors for $\mathbf{C}$ and $\mathbf{D}$. The second part describes the type of non-Gaussian distributions used on $\mathbf{z}$ in order to keep the model identifiable. The third part

considers how a search over permutations of the observed variables can be used in order to handle the constraints imposed on matrix $\mathbf{R}$. The last two parts describe how introducing Gaussian process process priors in the model can be used to model non-independent observations and non-linear dependencies in the DAGs.

3.1 Sparsity

The use of sparse models will in many cases give interpretable results and is often motivated by the principle of parsimony. Also, in many application domains it is also natural from a prediction point of view to enforce sparsity because the number of explanatory variables may exceed the number of examples by orders of magnitude. In regularized maximum likelihood type formulations of learning (maximum a-posteriori) it has become popular to use one-norm (L_1) regularization for example to achieve sparsity (Tibshirani, 1996). In the fully Bayesian inference setting (with averaging over variables), the corresponding Laplace prior will not lead to sparsity because it is very unlikely for a posterior summary like the mean, median or mode to be estimated as exactly zero even asymptotically. The same effect can be expected from any continuous distribution used for sparsity like Student's t, α-stable and bimodal priors (continuous slab and spike priors, Ishwaran and Rao, 2005). Exact zeros can only be achieved by placing a point mass at zero, i.e. explicitly specifying that the variable at hand is zero or not with some probability. This has motivated the introduction of many variants over the years of so-called slab and spike priors consisting of two component mixtures of a continuous part and a δ-function at zero (Lempers, 1971, Mitchell and Beauchamp, 1988, George and McCulloch, 1993, Geweke, 1996, West, 2003). In this paradigm, the columns of matrices $\mathbf{C}$ or $\mathbf{B}$ encode respectively, the connectivity of a factor or the set of parents associated to an observed variable. It is natural then to share information across elements in column j by assuming a common sparsity level $1-\nu_j$, suggesting the following hierarchy

$$\begin{aligned} c_{ij}|q_{ij},\cdot &\sim (1-q_{ij})\delta(c_{ij}) + q_{ij}\text{Cont}(c_{ij}|\cdot) \ , \\ q_{ij}|\nu_j &\sim \text{Bernoulli}(q_{ij}|\nu_j) \ , \\ \nu_j|\beta_m,\beta_p &\sim \text{Beta}(\nu_j|\beta_p\beta_m,\beta_p(1-\beta_m)) \ , \end{aligned} \tag{2}$$

where $\mathbf{Q}$, the binary matrix in equation (1) appears naturally, $\delta(\cdot)$ is a Dirac δ-function, $\text{Cont}(\cdot)$ is the continuous slab component, $\text{Bernoulli}(\cdot)$ and $\text{Beta}(\cdot)$ are Bernoulli and beta distributions, respectively. Reparameterizing the beta distribution as $\text{Beta}(\nu_j|\alpha\beta/m,\beta)$ and taking the number of columns m of $\mathbf{Q}\odot\mathbf{C}$ to infinity, leads to the non-parametric version of the slab and spike model with a so-called Indian buffet process prior over the (infinite) masking matrix $\mathbf{Q}=\{q_{ij}\}$ (Ghahramani et al., 2006). Note also that $q_{ij}|\nu_j$ is mainly used for clarity to make the binary indicators explicit, nevertheless in practice we can work directly with $c_{ij}|\nu_j,\cdot\sim(1-\nu_j)\delta(c_{ij})+\nu_j\text{Cont}(c_{ij}|\cdot)$ because q_{ij} can be marginalized out.

As illustrated and pointed out by Lucas et al. (2006) and Carvalho et al. (2008) the model with a shared beta-distributed sparsity level per factor introduces the undesirable side-effect that there is strong co-variation between the elements in each column of the masking matrix. For example, in high dimensions we might expect that only a finite number of elements are non-zero, implying a prior favoring a very high sparsity rate $1-\nu_j$. Because of the co-variation, even the parameters that are clearly non-zero will have a posterior

probability of being non-zero, $p(q_{ij} = 1|\mathbf{x}, \cdot)$, quite spread over the unit interval. Conversely, if our priors do not favor sparsity strongly, then the opposite situation will arise and the solution will become completely dense. In general, it is difficult to set the hyperparameters to achieve a sensible sparsity level. Ideally, we would like to have a model with a high sparsity level with high certainty about the non-zero parameters. We can achieve this by introducing a sparsity parameter η_{ij} for each element of $\mathbf{C}$ which has a mixture distribution with exactly this property

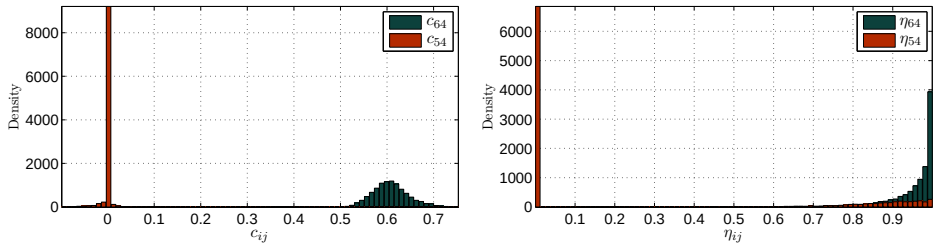
$$\begin{aligned} q_{ij}|\eta_{ij} &\sim \text{Bernoulli}(q_{ij}|\eta_{ij}) \ , \\ \eta_{ij}|\nu_j, \alpha_p, \alpha_m &\sim (1-\nu_j)\delta(\eta_{ij}) + \nu_j \text{Beta}(\eta_{ij}|\alpha_p\alpha_m, \alpha_p(1-\alpha_m)) \ . \end{aligned} \tag{3}$$

The distribution over η_{ij} expresses that we expect parsimony: either η_{ij} is zero exactly (implying that q_{ij} and c_{ij} are zero) or non-zero drawn from a beta distribution favoring high values, i.e. q_{ij} and c_{ij} are non-zero with high probability. We use $\alpha_p = 10$ and $\alpha_m = 0.95$ which has mean $\alpha_m = 0.95$ and variance $\alpha_m(1-\alpha_m)/(1+\alpha_p) \approx 0.086$. The expected sparsity rate of the modified model is $(1-\alpha_m)(1-\nu_j)$. This model has the additional advantage that the posterior distribution of η_{ij} directly measures the distribution of $p(q_{ij} = 1|\mathbf{x}, \cdot)$. This is therefore the statistic for ranking/selection purposes. Besides, we may want to reject interactions with high uncertainty levels when the probability of $p(q_{ij} = 1|\mathbf{x}, \cdot)$ is less or very close to the expected value, $\alpha_m(1-\nu_j)$.

To complete the specification of the prior, we let the continuous slab part in equation (2) be Gaussian distributed with inverse gamma prior on its variance. In addition, we scale the variances with ψ_i as

$$\begin{aligned} \text{Cont}(c_{ij}|\psi_i, \tau_{ij}) &= \mathcal{N}(c_{ij}|0, \psi_i\tau_{ij}) \ , \\ \tau_{ij}^{-1}|t_s, t_r &\sim \text{Gamma}(\tau_{ij}^{-1}|t_s, t_r) \ . \end{aligned} \tag{4}$$

This scaling makes the model easier to specify and tend to have better mixing properties (see Park and Casella, 2008). The slab and spike for $\mathbf{B}$ (DAG) is obtained from equations (2), (3) and (4) by simply replacing c_{ij} with b_{ij} and q_{ij} with r_{ij}. As already mentioned, we use $\alpha_p = 10$ and $\alpha_m = 0.95$ for the hierarchy in equation (3). For the column-shared parameter ν_j defined in equation (2) we set the precision to $\beta_p = 100$ and consider the mean values for factor models and DAGs separately. For the factor model we set a diffuse prior by making $\beta_m = 0.9$ to reflect that some of the factors can be in general nearly dense or empty. For the DAG we consider two settings, if we expect to obtain dense graphs we set $\beta_m = 0.99$, otherwise we set $\beta_m = 0.1$. Both settings can produce sparse graphs, however smaller values of β_m increase the overall sparsity rate and the gap between $p(r_{ij} = 0)$ and $p(r_{ij} = 1)$. A large separation between these two probabilities makes interpretation easier and also helps to spot non-zeros (edges) with high uncertainty. The hyperparameters for the variance of the non-zero elements of $\mathbf{B}$ and $\mathbf{C}$ are set to get a diffuse prior distribution bounded away from zero ($t_s = 2$ and $t_r = 1$), to allow for a better separation between slab and spike components. For the particular case of $\mathbf{C}_L$, in principle the prior should not have support on zero at all, i.e. the driving signal should not vanish, however for simplicity we allow this anyway as it has not given any problems in practice. Figure 4 shows a particular example of the posterior, $p(c_{ij}, \eta_{ij}|\mathbf{x}, \cdot)$ for two elements of $\mathbf{C}$ under the prior just described. In the example, $c_{64} \neq 0$ with high probability according to η_{ij}, whereas c_{54} is almost certainly zero



(a) (b)

Figure 4: Slab and spike prior example. (a) Posterior unnormalized densities for the magnitude of two particular elements of $\mathbf{C}$. (b) Posterior density for $\eta_{ij} = p(c_{ij} \neq 0|\mathbf{x}, \cdot)$. Here, $c_{64} \neq 0$ and $c_{54} = 0$ correspond to elements of the mixing matrix from the experiment shown in Figure 8.

since most of its probability mass is located exactly at zero, with some residual mass on the vicinity of zero, in Figure 4(a). In the one level hierarchy equation (2) sparsity parameters are shared, $\eta_{64} = \eta_{54} = \nu_4$. The result would then be less parsimonious with the posterior density of ν_4 being spread in the unit interval with a single mode located close to β_m.

3.2 Latent variables and driving signals

We consider two different non-Gaussian — heavy-tailed priors for $\mathbf{z}$, in order to obtain identifiable factor models and DAGs. A wide class of continuous, unimodal and symmetric distributions in one dimension can be represented as infinite scale mixtures of Gaussians, which are very convenient for Gibbs-sampling-based inference. We focus on Student's t and Laplace distributions which have the following mixture representation (Andrews and Mallows, 1974)

$$\text{Laplace}(z|\mu,\lambda) = \int_0^\infty \mathcal{N}(z|\mu,\upsilon)\text{Exponential}(\upsilon|\lambda^2)d\upsilon \,, \tag{5}$$

$$t(z|\mu,\theta,\sigma^2) = \int_0^\infty \mathcal{N}(z|\mu,\upsilon\sigma^2)\text{Gamma}\left(\upsilon^{-1}\left|\frac{\theta}{2},\frac{\theta}{2}\right.\right)d\upsilon \,, \tag{6}$$

where $\lambda > 0$ is the rate, $\sigma^2 > 0$ the scale, $\theta > 0$ is the degrees of freedom, and the distributions have exponential and gamma mixing densities accordingly. For varying degrees of freedom θ, the t distribution can interpolate between very heavy-tailed (power law and Cauchy when $\theta = 1$) and very light tailed, i.e. it becomes Gaussian when the degrees of freedom approaches infinity. The Laplace (or bi-exponential) distribution has tails which are intermediate between a t (with finite degrees of freedom) and a Gaussian. In this sense, the t distribution is more flexible but requires more careful selection of its hyperparameters because the model may become non-identifiable in the large θ limit (Gaussian).

An advantage of the Laplace distribution is that we can fix its parameter $\lambda = 1$ and let the model learn the appropriate scaling from $\mathbf{C}$ in equation (1). If we use the pure DAG model, we will need to have a hyperprior for λ^2 in order to learn the variances of the latent variables/driving signals, as in Henao and Winther (2009). A hierarchical prior for the

degrees of freedom in the t distribution is not easy to specify because there is no conjugate prior available with a standard closed form. Although a conjugate prior exists, is not straightforward to sample from it, since numerical integration must be used to compute its normalization constant. Another possibility is to treat θ as a discrete variable so computing the normalizing constant becomes straight forward.

Laplace and Student's t are not the only distributions admitting scale mixture representation. This mean that any other compatible type can be used as well, if the application requires it, and without considerable additional effort. Some examples include the logistic distribution (Andrews and Mallows, 1974), the stable family (West, 1987) and skewed versions of heavy-tailed distributions (Branco and Dey, 2001). Another natural extension to the mixtures scheme could be, for example, to set the mean of each component to arbitrary values and let the number of components be an infinite sum, thus ending up providing each factor with a Dirichlet process prior. This might be useful for cases when the latent factors are expected to be scattered in clusters due to the presence of subgroups in the data, as was shown by Carvalho et al. (2008).

3.3 Order search

We need to infer the order of the variables in the DAG to meet the constraints imposed on $\mathbf{R}$ in Section 2. The most obvious way is to try to solve this task by inferring all parameters $\{\mathbf{P}, \mathbf{B}, \mathbf{C}, \mathbf{z}, \boldsymbol{\epsilon}\}$ by a Markov chain Monte Carlo (MCMC) method such as Gibbs sampling. However, algorithms for searching over variable order prefer to work with models for which parameters other than $\mathbf{P}$ can be marginalized analytically (see Friedman and Koller, 2003, Teyssier and Koller, 2005). For our model, where we cannot marginalize analytically over $\mathbf{B}$ (due to $\mathbf{R}$ being binary), estimating $\mathbf{P}$ and $\mathbf{B}$ by Gibbs sampling would mean that we had to propose a new $\mathbf{P}$ for fixed $\mathbf{B}$. For example, exchanging the order of two variables would mean that they also exchange parameters in the DAG. Such a proposal would have very low acceptance, mainly as a consequence of the size of the search space and thus very poor mixing. In fact, for a given d number of variables there are $d!$ possible orderings $\mathbf{P}$, while there are $d!2^{(d(d+2m-1))/2}$ possible structures for $\{\mathbf{P}, \mathbf{B}, \mathbf{C}\}$. We therefore opt for an alternative strategy by exploiting the equivalence between factor models and DAGs shown in Section 2.1. In particular for $m = 0$, since $\mathbf{B}$ can be permuted to strictly lower triangular, then $\mathbf{D} = (\mathbf{I} - \mathbf{B})^{-1}\mathbf{C}_D$ can be permuted to triangular. This means that we can perform inference for the factor model to obtain $\mathbf{D}$ while searching in parallel for a set of permutations $\mathbf{P}$ that are in good agreement (in a probabilistic sense) with the triangular requirement of $\mathbf{D}$. Such a set of orderings is found during the inference procedure of the factor model. To set up the stochastic search, we need to modify the factor model slightly by introducing separate data (row) and factor (column) permutations, $\mathbf{P}$ and $\mathbf{P}_\mathrm{f}$ to obtain $\mathbf{x} = \mathbf{P}^\top\mathbf{D}\mathbf{P}_\mathrm{f}\mathbf{z} + \boldsymbol{\epsilon}$. The reason for using two different permutation matrices, rather than only one like in the definition of the DAG model, is that we need to account for the permutation freedom of the factor model (see Section 2.1). Using the same permutation for row and column would thus require an additional step to identify the columns in the factor model. We make inference for the unrestricted factor model, but propose $\mathbf{P}^\star$ and $\mathbf{P}_\mathrm{f}^\star$ independently according to $q(\mathbf{P}^\star|\mathbf{P})q(\mathbf{P}_\mathrm{f}^\star|\mathbf{P}_\mathrm{f})$. Both distributions draw a new permutation matrix by exchanging two randomly chosen elements, e.g. the order may change as

$[x_1, x_2, x_3, x_4]^\top \rightarrow [x_1, x_4, x_3, x_2]^\top$. In other words, the proposals $q(\mathbf{P}^\star|\mathbf{P})$ and $q(\mathbf{P}_\mathrm{f}^\star|\mathbf{P}_\mathrm{f})$ are uniform distributions over the space of transpositions for $\mathbf{P}$ and $\mathbf{P}_\mathrm{f}$. Assuming we have no a-priori preferred ordering, we may use a Metropolis-Hastings (M-H) acceptance probability $\min(1, \xi_{\rightarrow\star})$ with $\xi_{\rightarrow\star}$ as a simple ratio of likelihoods with the permuted $\mathbf{D}$ masked to match the triangularity assumption. Formally, we use the binary mask $\mathbf{M}$ (containing zeros above the diagonal of its d first columns) and write

$$\xi_{\rightarrow\star} = \frac{\mathcal{N}(\mathbf{X}|(\mathbf{P}^\star)^\top(\mathbf{M}\odot\mathbf{P}^\star\mathbf{D}(\mathbf{P}_\mathrm{f}^\star)^\top)\mathbf{P}_\mathrm{f}^\star\mathbf{Z}, \mathbf{\Psi})}{\mathcal{N}(\mathbf{X}|\mathbf{P}^\top(\mathbf{M}\odot\mathbf{P}\mathbf{D}\mathbf{P}_\mathrm{f}^\top)\mathbf{P}_\mathrm{f}\mathbf{Z}, \mathbf{\Psi})} \ , \tag{7}$$

where $\mathbf{M}\odot\mathbf{D}$ is the masked $\mathbf{D}$ and $\mathbf{Z} = [\mathbf{z}_1 \ \ldots \mathbf{z}_N]$. The procedure can be seen as a simple approach for generating hypotheses about good orderings, producing close to triangular versions of $\mathbf{D}$, in a model where the slab and spike prior provide the required bias towards sparsity. Once the inference is done, we end up having an estimate for the desired distribution over permutations $\mathbf{P} = \sum_i^{d!}\pi_i\delta_{\mathbf{P}_i}$, where $\boldsymbol{\pi} = [\pi_1 \ \pi_2 \ \ldots]$ is a sparse vector containing the probability for $\mathbf{P} = \mathbf{P}_i$, which in our case is proportional to the number of times permutation $\mathbf{P}_i$ was accepted by the M-H update during inference. Note that $\mathbf{P}_\mathrm{f}$ is just a nuisance variable that does not need to be stored or summarized.

3.4 Allowing for correlated data (CSLIM)

For the case where independence of observed variables cannot be assumed, for instance due to (time) correlation or smoothness, the priors discussed before for the latent variables and driving signals do not really apply anymore, however the only change we need to make is to allow elements in rows of $\mathbf{Z}$ to correlate. We can assume then independent Gaussian process (GP) priors for each latent variable instead of scale mixtures of Gaussians, to obtain what we have called correlated sparse linear identifiable modeling (CSLIM). For a set of N realizations of variable j we set

$$z_{j1}, \ldots, z_{jN}|\upsilon_j \ \sim \ \mathrm{GP}(z_{j1}, \ldots, z_{jN}|k_{\upsilon_j,n}(\cdot)) \ , \tag{8}$$

where the covariance function has the form $k_{\upsilon_j,n}(n, n') = \exp(-\upsilon_j(n-n')^2)$, $\{n, n'\}$ is a pair of observation indices or time points and υ_j is the length scale controlling the overall level of correlation allowed for each variable (row) in $\mathbf{Z}$. Conceptually, equation (8) implies that each latent variable j is sampled from a function and the GP acts as a prior over continuous functions. Since such a length scale is very difficult to set just by looking at the data, we further place priors on υ_j as

$$\upsilon_j|u_s, \kappa \ \sim \ \mathrm{Gamma}(\upsilon_j|u_s, \kappa) \ , \quad \kappa|k_s, k_r \ \sim \ \mathrm{Gamma}(\kappa|k_s, k_r) \ . \tag{9}$$

Given that the conditional distribution of $\boldsymbol{\upsilon} = [\upsilon_1, \ldots, \upsilon_m]$ is not of any standard form, Metropolis-Hastings updates are used. In the experiments we use that $u_s = k_s = 2$ and $k_r = 0.02$. The details concerning inference for this model are given in Appendix A.

It is also possible to easily expand the possible applications of GP priors in this context by, for instance, using more structured covariance functions through scale mixture of Gaussian representations to obtain a prior distribution for continuous functions with heavy-tailed behavior — a t-processes (Yu et al., 2007), or learning the covariance function as well using inverse Wishart hyperpriors.

3.5 Allowing for non-linearities (SNIM)

Provided that we know the true ordering of the variables, i.e. $\mathbf{P}$ is known then $\mathbf{B}$ is surely strictly lower triangular. It is very easy to allow for non-linear interactions in the DAG model from equation (1) by rewriting it as

$$\mathbf{Px} = (\mathbf{R} \odot \mathbf{B})\mathbf{Py} + (\mathbf{Q} \odot \mathbf{C})\mathbf{z} + \boldsymbol{\epsilon} \ , \tag{10}$$

where $\mathbf{y} = [y_1, \ldots, y_d]^\top$ and $y_{i1}, \ldots, y_{iN} | \upsilon_i \sim \mathrm{GP}(y_{i1}, \ldots, y_{iN} | k_{\upsilon_i, x}(\cdot))$ has a Gaussian process prior with for instance, but not limited to, a stationary covariance function like $k_{\upsilon_i,x}(\mathbf{x}, \mathbf{x}') = \exp(-\upsilon_i(\mathbf{x} - \mathbf{x}')^2)$, similar to equation (8) and with the same hyperprior structure as in equation (9). This is a straight forward extension that we call sparse non-linear multivariate modeling (SNIM) that is in spirit similar to Friedman and Nachman (2000), Hoyer et al. (2009), Zhang and Hyvärinen (2009, 2010), Tillman et al. (2009), however instead of treating the inherent multiple regression problem in equation (10) and the conditional independence of the observed variables independently, we proceed within our proposed framework by letting the multiple regressor be sparse, thus the conditional independences are encoded through $\mathbf{R}$. The main limitation of the model in equation (10) is that if the true ordering of the variables is unknown, the exhaustive enumeration of $\mathbf{P}$ is needed. This means that this could be done for very small networks, e.g. up to 5 or 6 variables. In principle, an ordering search procedure for the non-linear model only requires the latent variables $\mathbf{z}$ to have Gaussian process priors as well. The main difficulty is that in order to build covariance functions for $\mathbf{z}$ we need a set of observations that are not available because $\mathbf{z}$ is latent.

4. Model comparison

Quantitative model comparison between factor models and DAGs is a key ingredient in SLIM. The joint probability of data $\mathbf{X}$ and parameters for the factor model part in equation (1) is

$$p(\mathbf{X}, \mathbf{C}, \mathbf{Z}, \boldsymbol{\epsilon}, \cdot) = p(\mathbf{X}|\mathbf{C}, \mathbf{Z}, \boldsymbol{\epsilon})p(\mathbf{C}|\cdot)p(\mathbf{Z}|\cdot)p(\boldsymbol{\epsilon})p(\cdot) \ ,$$

where $(\cdot)$ indicates additional parameters in the hierarchical model. Formally the Bayesian model selection yardstick, the marginal likelihood for model $\mathcal{M}$

$$p(\mathbf{X}|\mathcal{M}) = \int p(\mathbf{X}|\boldsymbol{\Theta}, \mathbf{Z})p(\boldsymbol{\Theta}|\mathcal{M})p(\mathbf{Z}|\mathcal{M})d\boldsymbol{\Theta}d\mathbf{Z} \ ,$$

can be obtained by marginalizing the joint over the parameters $\boldsymbol{\Theta}$ and latent variables $\mathbf{Z}$. Computationally this is a difficult task because the marginal likelihood cannot be written as an average over the posterior distribution in a simple way. It is still possible using MCMC methods, for example by partitioning of the parameter space and multiple chains or thermodynamic integration (see Chib, 1995, Neal, 2001, Murray, 2007, Friel and Pettitt, 2008), but in general it must be considered as computationally expensive and non-trivial. On the other hand, evaluating the likelihood on a test set $\mathbf{X}^\star$, using predictive densities $p(\mathbf{X}^\star|\mathbf{X}, \mathcal{M})$ is simpler from a computational point of view because it can be written in terms of an average over the posterior of the *intensive variables*, $p(\mathbf{C}, \boldsymbol{\epsilon}, \cdot|\mathbf{X})$ and the prior

distribution of the *extensive variables* associated with the test points[2], $p(\mathbf{Z}^\star|\cdot)$ as

$$\mathcal{L}_{\text{FM}} \stackrel{\text{def}}{=} p(\mathbf{X}^\star|\mathbf{X},\mathcal{M}_{\text{FM}}) = \int p(\mathbf{X}^\star|\mathbf{Z}^\star,\boldsymbol{\Theta}_{\text{FM}},\cdot)p(\mathbf{Z}^\star|\cdot)p(\boldsymbol{\Theta}_{\text{FM}},\cdot|\mathbf{X})d\mathbf{Z}^\star d\boldsymbol{\Theta}_{\text{FM}}d(\cdot)\ , \tag{11}$$

where $\boldsymbol{\Theta}_{\text{FM}} = \{\mathbf{C},\boldsymbol{\epsilon}\}$. This average can be approximated by a combination of standard sampling and exact marginalization using the scale mixture representation of the heavy-tailed distributions presented in Section 3.2. For the full DAG model in equation (1), we will not average over permutations $\mathbf{P}$ but rather calculate the test likelihood for a number of candidates $\mathbf{P}^{(1)},\ldots,\mathbf{P}^{(c)},\ldots$ as

$$\begin{aligned}\mathcal{L}_{\text{DAG}} &\stackrel{\text{def}}{=} p(\mathbf{X}^\star|\mathbf{P}^{(c)},\mathbf{X},\mathcal{M}_{\text{DAG}})\ ,\\ &= \int p(\mathbf{X}^\star|\mathbf{P}^{(c)},\mathbf{X},\mathbf{Z}^\star,\boldsymbol{\Theta}_{\text{DAG}},\cdot)p(\mathbf{Z}^\star|\cdot)p(\boldsymbol{\Theta}_{\text{DAG}},\cdot|\mathbf{X})d\mathbf{Z}^\star d\boldsymbol{\Theta}_{\text{DAG}}d(\cdot)\ ,\end{aligned} \tag{12}$$

where $\boldsymbol{\Theta}_{\text{DAG}} = \{\mathbf{B},\mathbf{C},\boldsymbol{\epsilon}\}$. We use sampling to compute the test likelihoods in equations (11) and (12). With Gibbs, we draw samples from the posterior distributions $p(\boldsymbol{\Theta}_{\text{FM}},\cdot|\mathbf{X})$ and $p(\boldsymbol{\Theta}_{\text{DAG}},\cdot|\mathbf{X})$, where $(\cdot)$ is shorthand for example for the degrees of freedom θ, if Student t distributions are used. The average over the extensive variables associated with the test points $p(\mathbf{Z}^\star|\cdot)$ is slightly more complicated because naively drawing samples from $p(\mathbf{Z}^\star|\cdot)$ results in an estimator with high variance — for $\psi_i \ll \upsilon_{jn}$. Instead we exploit the infinite mixture representation to marginalize exactly $\mathbf{Z}^\star$ and then draw samples in turn for the scale parameters. Omitting the permutation matrices for clarity, in general we get

$$\begin{aligned}p(\mathbf{X}^\star|\boldsymbol{\Theta},\cdot) &= \int p(\mathbf{X}^\star|\mathbf{Z}^\star,\boldsymbol{\Theta},\cdot)p(\mathbf{Z}^\star|\cdot)d\mathbf{Z}^\star\ ,\\ &= \prod_n \int \mathcal{N}(\mathbf{x}_n^\star|\mathbf{m}_n,\boldsymbol{\Sigma}_n)\prod_j p(\upsilon_{jn}|\cdot)d\upsilon_{jn} \approx \frac{1}{N_{\text{rep}}}\prod_n\sum_r^{N_{\text{rep}}}\mathcal{N}(\mathbf{x}_n^\star|\mathbf{m}_n,\boldsymbol{\Sigma}_n)\ ,\end{aligned}$$

where N_{rep} is the number of samples generated to approximate the intractable integral ($N_{\text{rep}} = 500$ in the experiments). For the factor model $\mathbf{m}_n = \mathbf{0}$ and $\boldsymbol{\Sigma}_n = \mathbf{C}_D\mathbf{U}_n\mathbf{C}_D^\top + \boldsymbol{\Psi}$. For the DAG, $\mathbf{m}_n = \mathbf{B}\mathbf{x}_n^\star$ and $\boldsymbol{\Sigma}_n = \mathbf{C}\mathbf{U}_n\mathbf{C}^\top + \boldsymbol{\Psi}$. The covariance matrix $\mathbf{U}_n = \text{diag}(\upsilon_{1n},\ldots,\upsilon_{(d+m)n})$ with elements υ_{jn}, is sampled directly from the prior, accordingly. Once we have computed $p(\mathbf{X}^\star|\boldsymbol{\Theta}_{\text{FM}},\cdot)$ for the factor model and $p(\mathbf{X}^\star|\boldsymbol{\Theta}_{\text{DAG}},\cdot)$ for the DAG, we can use them to average over $p(\boldsymbol{\Theta}_{\text{FM}},\cdot|\mathbf{X},)$ and $p(\boldsymbol{\Theta}_{\text{DAG}},\cdot|\mathbf{X})$ to obtain the predictive densities $p(\mathbf{X}^\star|\mathbf{X},\mathcal{M}_{\text{FM}})$ and $p(\mathbf{X}^\star|\mathbf{X},\mathcal{M}_{\text{DAG}})$, respectively.

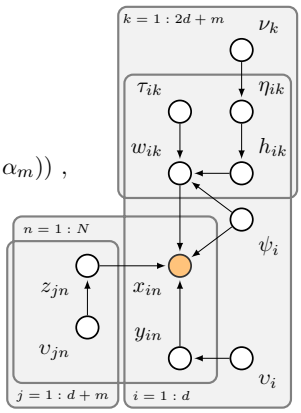
For the particular case in which $\mathbf{X}$ and consequently $\mathbf{Z}$ are correlated variables — CSLIM, we use a slightly different procedure for model comparison. Instead of using a test set, we randomly remove some proportion of the elements of $\mathbf{X}$ and perform inference with missing values, then we summarize the likelihood on the missing values. In particular, for the factor model we use $\mathbf{M}_{\text{miss}}\odot\mathbf{X} = \mathbf{M}_{\text{miss}}\odot(\mathbf{Q}_L\odot\mathbf{C}_L\mathbf{Z}+\boldsymbol{\epsilon})$ where $\mathbf{M}_{\text{miss}}$ is a binary masking matrix with zeros corresponding to test points, i.e. the missing values. See details in Appendix A. Note that this scheme is not exclusive to CSLIM thus can be also used with SLIM or when the observed data contain actual missing values.

2. Intensive means not scaling with the sample size. Extensive means scaling with sample size in this case the size of the test sample.

5. Model overview and practical details

The three models described in the previous section namely SLIM, CSLIM and SNIM can be summarized as a graphical model and as a probabilistic hierarchy as follows

$$
\begin{aligned}
\mathbf{x}_n|\mathbf{W},\mathbf{y}_n,\mathbf{z}_n,\mathbf{\Psi} &\sim \mathcal{N}(\mathbf{x}_n|\mathbf{W}[\mathbf{y}_n\ \mathbf{z}_n]^\top,\mathbf{\Psi})\ , \quad \mathbf{W}=[\mathbf{B}\ \mathbf{C}]\ ,\\
\psi_i^{-1}|s_s,s_r &\sim \text{Gamma}(\psi_i^{-1}|s_s,s_r)\ ,\\
w_{ik}|h_{ik},\psi_i,\tau_{ik} &\sim (1-h_{ik})\delta_0(w_{ik})+h_{ik}\mathcal{N}(w_{ik}|0,\psi_i\tau_{ik})\ ,\\
h_{ik}|\eta_{ik} &\sim \text{Bernoulli}(h_{ik}|\eta_{ik})\ , \quad \mathbf{H}=[\mathbf{R}\ \mathbf{Q}]\ ,\\
\eta_{ik}|\nu_k,\alpha_p,\alpha_m &\sim (1-\nu_k)\delta(\eta_{ik})+\nu_k\text{Beta}(\eta_{ik}|\alpha_p\alpha_m,\alpha_p(1-\alpha_m))\ ,\\
\nu_k|\beta_m,\beta_p &\sim \text{Beta}(\nu_k|\beta_p\beta_m,\beta_p(1-\beta_m))\ ,\\
\tau_{ik}^{-1}|t_s,t_r &\sim \text{Gamma}(\tau_{ik}^{-1}|t_s,t_r)\ ,\\
z_{j1},\ldots,z_{jN}|\upsilon &\sim \begin{cases}\prod_n \mathcal{N}(z_{jn}|0,\upsilon_{jn})\ , & \text{(SLIM)}\\ \text{GP}(z_{j1},\ldots,z_{jN}|k_{\upsilon_j,n}(\cdot))\ , & \text{(CSLIM)}\end{cases}\\
y_{i1},\ldots,y_{iN}|\upsilon &\sim \begin{cases}x_{i1},\ldots,x_{iN}\ , & \text{(SLIM)}\\ \text{GP}(y_{i1},\ldots,y_{iN}|k_{\upsilon_i,x}(\cdot))\ , & \text{(SNIM)}\end{cases}
\end{aligned}
$$



where we have omitted $\mathbf{P}$ and the hyperparameters in the graphical model. Latent variable and driving signal parameters υ can have one of several priors: Exponential$(\upsilon|\lambda^2)$ (Laplace), Gamma$(\upsilon^{-1}|\theta/2,\theta/2)$ (Student's t) or Gamma$(\upsilon|u_s,\kappa)$ (GP), see equations (5), (6) and (9), respectively. The latent variables/driving signals z_{jn} and the mixing/connectivity matrices with elements c_{ij} or b_{ij} are modeled independently. Each element in $\mathbf{B}$ and $\mathbf{C}$ has its own slab variance τ_{ij} and probability of being non-zero η_{ij}. Moreover, there is a shared sparsity rate per column ν_k. Variables υ_{jn} are variances if z_{jn} use a scale mixture of Gaussian's representation, or length scales in the GP prior case. Since we assume no sparsity for the driving signals, $\eta_{ik}=1$ for $d+i=k$ and $\eta_{ik}=0$ for $d+i\neq k$. In addition, we can recover the pure DAG by making $m=0$ and the standard factor model by making instead $\eta_{ik}=0$ for $k\leq 2d$. All the details for the Gibbs sampling based inference are summarized in appendix A.

5.1 Proposed workflow

We propose the workflow shown in Figure 1 to integrate all elements of SLIM, namely factor model and DAG inference, stochastic order search and model selection using predictive densities.

1. Partition the data into $\{\mathbf{X},\mathbf{X}^\star\}$.

2. Perform inference on the factor model and stochastic order search. One Gibbs sampling update consists of computing the conditional posteriors in equations (13), (14), (15), (16), (17), (18) and (19) in sequence, followed by several repetitions (we use 10) of the M-H update in equation 7 for the permutation matrices $\mathbf{P}$ and $\mathbf{P}_\text{f}$.

3. Summarize the factor model, mainly $\mathbf{C}$, $\{\eta_{ij}\}$ and $\mathcal{L}_{\text{FM}}$ using quantiles (0.025, 0.5 and 0.975).

4. Summarize the orderings, $\mathbf{P}$. Select the top m_{top} candidates according to their frequency during inference in step 2.

5. Perform inference on the DAGs for each one of the ordering candidates, $\mathbf{P}^{(1)}, \ldots, \mathbf{P}^{(m_{\text{top}})}$ using Gibbs sampling by computing equations (13), (14), (15), (16), (17), (18) and (19) in sequence, up to minor changes described in Appendix A.

6. Summarize the DAGs, $\mathbf{B}$, $\mathbf{C}_L$, $\{\eta_{ik}\}$ and $\mathcal{L}_{\text{DAG}}^{(1)}, \ldots, \mathcal{L}_{\text{DAG}}^{(m_{\text{top}})}$ using quantiles (0.025, 0.5 and 0.975). Note that $\{\eta_{ik}\}$ contains non-zero probabilities for $\mathbf{R}$ and $\mathbf{Q}$ corresponding to $\mathbf{B}$ and $\mathbf{C}_L$, respectively.

We use medians to summarize all quantities in our model because $\mathbf{D}$, $\mathbf{B}$ and $\{\eta_{ik}\}$ are bimodal while the remaining variables are in general skewed posterior distributions. Inference with GP priors for time series data (CSLIM) or non-linear DAGs (SNIM) is fairly similar to the i.i.d. case, see Appendix A for details. Source code for SLIM and all its variants proposed so far has been made available at `http://cogsys.imm.dtu.dk/slim/` as Matlab scripts.

5.2 Computational cost

The cost of running the linear DAG with latent variables or the factor model is roughly the same, i.e. $\mathcal{O}(N_s d^2 N)$ where N_s is the total number of samples including the burn-in period. The memory requirements on the other hand are approximately $\mathcal{O}(N_p d^2)$ if all the samples after the burn-in period N_p are stored. This means that the inference procedures scale reasonably well if N_s is kept in the lower ten thousands. The non-linear version of the DAG is considerably more expensive due to the GP priors, hence the computational cost rises up to $\mathcal{O}(N_s(d-1)N^3)$.

The computational cost of LiNGAM, being the closest to our linear models, is mainly dependent on the statistic used to prune/select the model. Using bootstrapping results in $\mathcal{O}(N_b^3)$, where N_b is the number of bootstrap samples. The Wald statistic leads to $\mathcal{O}(d^6)$, while Wald with χ^2 second order model fit test amounts to $\mathcal{O}(d^7)$. As for the memory requirements, bootstrapping is very economic whereas Wald-based statistics require $\mathcal{O}(d^6)$.

The method for non-linear DAGs described in Hoyer et al. (2009) is defined for a pair of variables, and it uses GP-based regression and kernelized independence tests. The computational cost is $\mathcal{O}(N_g N^3)$ where N_g is the number of gradient iterations used to maximize the marginal likelihood of the GP. This is the same order of complexity as our non-linear DAG sampler.

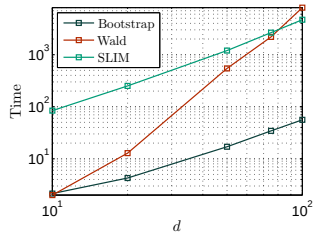


Figure 5: Runtime comparison.

Figure 5 shows average running times in a standard desktop machine (two cores, 2.6GHz and 4Gb RAM) over 10 different models with $N = 1000$ and $d = \{10, 20, 50, 100\}$. As expected, LiNGAM with bootstrap

is very fast compared to the others while our model approaches LiNGAM with Wald statistic as the number of observations increases. We did not include LiNGAM with second order model fit because for $d = 50$ it is already prohibitive. For this small test we used a C implementation of our model with $N_s = 19000$. We are aware that the performance of a C and a Matlab implementation can be different, however we still do the comparison because the most expensive operations in the Matlab code for LiNGAM are computed through BLAS routines not involving large loops, thus a C implementation of LiNGAM should not be noticeably faster than its Matlab counterpart.

6. Simulation results

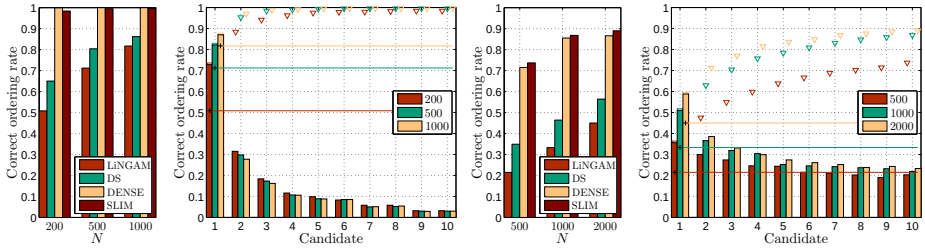
We consider six sets of experiments to illustrate the features of SLIM. In our comparison with other methods we focus on the DAG structure learning part because it is somewhat easier to benchmark a DAG than a factor model. However, we should stress that DAG learning is just one component of SLIM. Both types of model and their comparison are important, as will be illustrated through the experiments. For the reanalysis of flow cytometry data using our models, quantitative model comparison favors the DAG with latent variables rather than the standard factor model or the pure DAG which was the paradigm used in the structure learning approach of Sachs et al. (2005).

The first two experiments consist of extensive tests using artificial data in a setup originally from LiNGAM and network structures taken from the Bayesian net repository. We test the features of SLIM and compare with LiNGAM and some other methods in settings where they have proved to work well. The third set of experiments addresses model comparison, the fourth and fifth present results for our DAG with latent variables and the non-linear DAG (SNIM) on both artificial and real data. The sixth uses real data previously published by Sachs et al. (2005) and the last one provides simple results for a factor model using Gaussian process priors for temporal smoothness (CSLIM), tested on a time series gene expression data set (Kao et al., 2004). In all cases we ran 10000 samples after a burn-in period of 5000 for the factor model, and a single chain with 3000 samples and 1000 as burn-in iterations for the DAG, i.e. $N_s = 19000$ used in the computational cost comparison. As a summary statistic we use median values everywhere, and Laplace distributions for the latent factors if not stated otherwise.

6.1 Artificial data

We evaluate the performance of our model against LiNGAM[3], using the artificial model generator presented and fully explained in Shimizu et al. (2006). Concisely, the generator produces both dense and sparse networks with different degrees of sparsity, $\mathbf{Z}$ is generated from a heavy-tailed non-Gaussian distribution through a generalized Gaussian distribution with zero mean, unit variance and random shape, $\mathbf{X}$ is generated recursively using equation (1) with $m = 0$ and then randomly permuted to hide the correct order, $\mathbf{P}$. Approximately, half of the networks are fully connected while the remaining portion comprises sparsity levels between 10% and 80%. Having dense networks (0% sparsity) in the benchmark is crucial because in such cases the correct order of the variables is unique, thus more difficult to find.

3. Matlab package (v.1.42) available at http://www.cs.helsinki.fi/group/neuroinf/lingam/.



(a) (b) (c) (d)

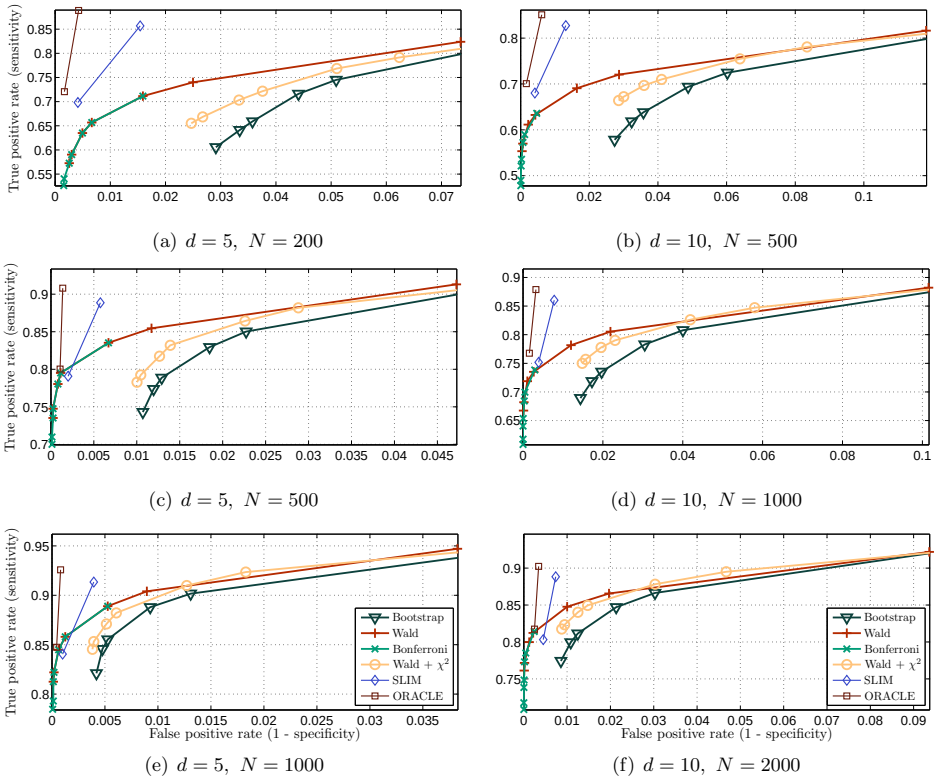
Figure 6: Ordering accuracies for LiNGAM suite using $d = 5$ in (a,b) and $d = 10$ in (c,d). (a,c) Total correct ordering rates where DENSE is our factor model without sparsity prior and DS corresponds to DENSE but using the deterministic ordering search used in LiNGAM. (b,c) Correct ordering rate vs. candidates from SLIM. The crosses and horizontal lines correspond to LiNGAM while the triangles are accumulated correct orderings across candidates used by SLIM.

This setup is particularly challenging because the model needs to identify both dense and sparse models. For the experiment we have generated 1000 different dataset/models using $d = \{5, 10\}$, $N = \{200, 500, 1000, 2000\}$ and the DAG was selected using the median of the training likelihood, $p(\mathbf{X}|\mathbf{P}_{\mathrm{r}}^{(k)}, \mathbf{R}^{(k)}, \mathbf{B}^{(k)}, \mathbf{C}_D^{(k)}, \mathbf{Z}, \boldsymbol{\Psi}, \cdot)$, for $k = 1, \ldots, m_{\mathrm{top}}$.

Order search. With this experiment we want to quantify the impact of using sparsity, stochastic ordering search and more than one ordering candidate, i.e. $m_{\mathrm{top}} = 10$ in total. Figure 6 evaluates the proportion of correct orderings for different settings. We have the following abbreviations for this experiment, DENSE is our factor model without sparsity prior, i.e. assuming that $p(r_{ij} = 1) = 1$ a priori. DS (deterministic search) assumes no sparsity as in DENSE but replaces our stochastic search for permutations with the deterministic approach used by LiNGAM, i.e. we replace the M-H update from equation (7) by the procedure described next: after inference we compute $\mathbf{D}^{-1}$ followed by a column permutation search using the Hungarian algorithm and a row permutation search by iterative pruning until getting a version of $\mathbf{D}$ as triangular as possible (Shimizu et al., 2006). Several comments can be made from the results, (i) For $d = 5$ there is no significant gain for increasing N, mainly because the size of the permutation space is small, i.e. 5!. (ii) The difference in performance between SLIM and DENSE is not significative because we look for triangular matrices in a probabilistic sense, hence there is no real need for exact zeros but just very small values, this does not mean that the sparsity in the factor model is unnecessary, on the contrary we still need it if we want to have readily interpretable mixing matrices. (iii) Using more than one ordering candidate considerably improves the total correct ordering rate, e.g. by almost 30% for $d = 5$, $N = 200$ and 35% for $d = 10$, $N = 500$. (iv) The number of accumulated correct orderings found saturates as the number of candidates used increases, suggesting that further increasing m_{top} will not considerably change the overall results. (v) The number of correct orderings tends to accumulate on the first candidate when

N increases since the uncertainty of the estimation of the parameters in the factor model decreases accordingly. (vi) When the network is not dense, it could happen that more than one candidate has a correct ordering, hence the total rates (triangles) are not just the sum of the bar heights in Figures 6(b) and 6(d). (vii) It seems that except for $d = 10,\ N = 5000$ it is enough to consider just the first candidate in SLIM to obtain as many correct orderings as LiNGAM does. (viii) From Figures 6(a) and 6(c), the three variants of SLIM considered perform better than LiNGAM, even when using the same single candidate ordering search proposed by Shimizu et al. (2006). (ix) In some cases the difference between SLIM and LiNGAM is very large, for example, for $d = 10$ using two candidates and $N = 1000$ is enough to obtain as many correct orderings as LiNGAM with $N = 5000$.

DAG learning. Now we evaluate the ability of our model to capture the DAG structure in the data, provided the permutation matrices obtained in the previous stage as a result of our stochastic order search. Results are summarized in Figure 7 using receiving operating characteristic (ROC) curves. The true and false positive rates are averaged over the number of trials (1000) for each setting to make the scaling in the plots more meaningful given the various levels of sparsity considered. The rates are computed in the usual way, however it must be noted that the true number of absent links in a network can be as large as $d(d-1)$, i.e. twice the number of links in a DAG, because in the case of an estimated DAG based in a wrong ordering the number of false positives can sum up to $d(d-1)/2$ even if the true network is not empty. For LiNGAM we use four different statistics to prune the DAG after the ordering has been found, namely bootstrapping, Wald, Bonferroni and Wald with second order χ^2 model fit test. In every case we run LiNGAM for 7 different p-value cutoffs, namely, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1 and 0.5 to build the ROC curve. For SLIM we consider the two settings for β_m discussed in Section 3.1, i.e. a diffuse prior supporting the existence of dense graphs, $\beta_m = 0.99$ and $\beta_m = 0.1$. In order to test how good SLIM is at selecting one DAG out of the m_{top} candidates, we also report the oracle results under the name of ORACLE, where in every case we select the candidate with less error instead of $\text{argmax}_k\ p(\mathbf{X}|\mathbf{P}_{\text{r}}^{(k)}, \mathbf{R}^{(k)}, \mathbf{B}^{(k)}, \mathbf{C}_D^{(k)}, \mathbf{Z}, \mathbf{\Psi}, \cdot)$. Using $\beta_m = 0.99$ is not very useful in practice because in a real situation we expect that the underlying DAG is sparse, however the LiNGAM suite has as many dense graphs as sparse ones making $\beta_m = 0.1$ a poor choice. From Figure 7, it is clear that for $\beta_m = 0.99$, SLIM is clearly superior, providing the best true positive rate (TPR) - false positive rate (FPR) tradeoff. For $\beta_m = 0.1$ there is no real difference between SLIM and some settings of LiNGAM (Wald and Bonferroni). Concerning SLIM's model selection procedure, it can be seen that the difference between SLIM and ORACLE nicely decreases as the number of observations increases. We also tested the DAG learning procedure in SLIM when the true ordering is known (results not shown) and we found only a very small difference compared to ORACLE. It is important to mention that further increasing or reducing β_m does not significantly change the results shown; this is because β_m does not fully control the sparsity of the model, thus even for $\beta_m = 1$ the model will be still sparse due to element-wise link confidence, α_m. As for LiNGAM, it seems that Wald performs better than Wald + χ^2, however just by looking at Figure 7, it is to be expected that for larger N the latter perform better because the Wald statistic alone will tend to select more dense models.



(a) $d = 5,\ N = 200$ (b) $d = 10,\ N = 500$

(c) $d = 5,\ N = 500$ (d) $d = 10,\ N = 1000$

(e) $d = 5,\ N = 1000$ (f) $d = 10,\ N = 2000$

Figure 7: Performance measures for LiNGAM suite. Results include the settings: $d = \{5, 10\}$, $N = \{200, 500, 1000, 2000\}$, four model selectors for LiNGAM (bootstrap, Wald, Bonferroni and Wald + χ^2 statistics) and seven p-value cutoffs for the statistics used in LiNGAM (0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5). ORACLE corresponds to oracle results for SLIM, both computed for two settings: diffuse $\beta_m = 0.99$ and sparse $\beta_m = 0.1$ priors. Markers close to the top-left corner denote better results in average.

Illustrative example. Finally we want to show some of the most important elements of SLIM taking one successfully estimated example from the LiNGAM suite. Figure 8 shows results for a particular DAG with 10 variables obtained using 500 observations, see Figures 8(d) and 8(e) for the ground truth and the estimated DAG, respectively. True and estimated mixing matrices $\mathbf{D}$ for the equivalent factor model are also shown in Figures 8(a) and 8(b), respectively. In total our algorithm produced 92 orderings out of 3.6×10^6 possible, from which all $m_{\text{top}} = 10$ candidates were correct. Figure 8(c) shows the first 50 candidates and their frequency during sampling, the shaded area encloses the $m_{\text{top}} = 10$ candidates. From

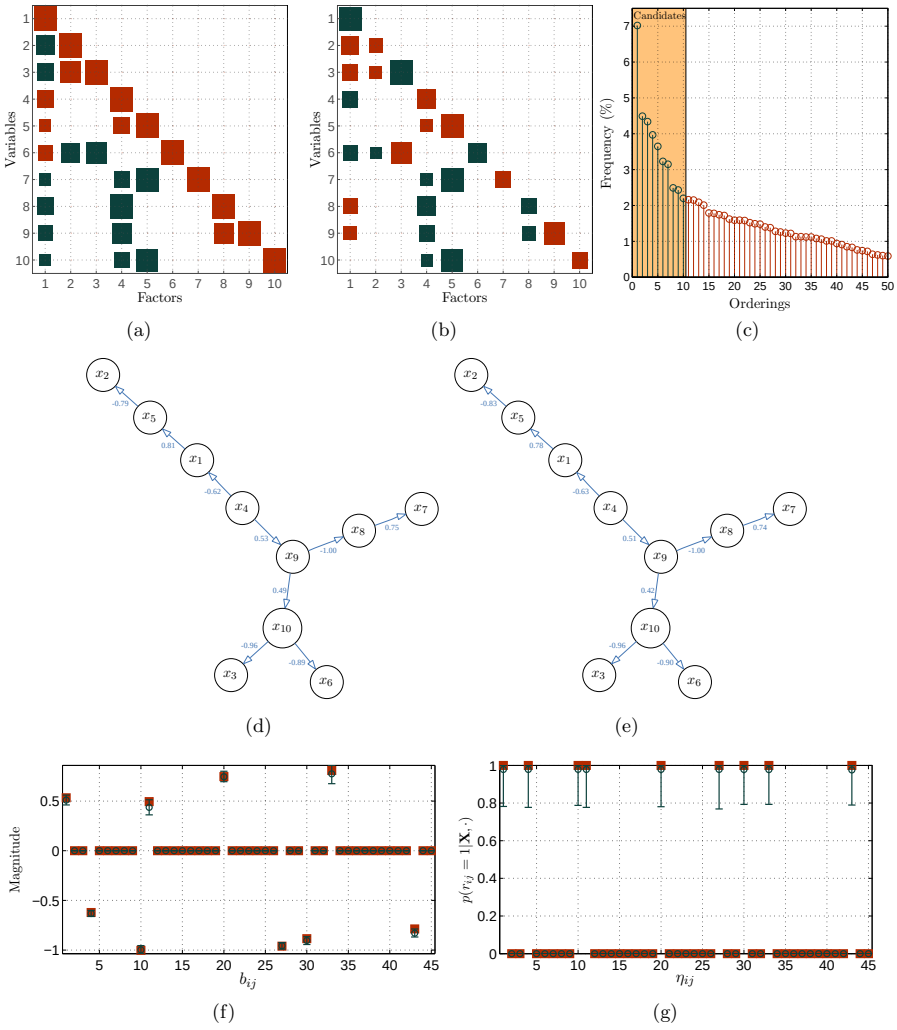
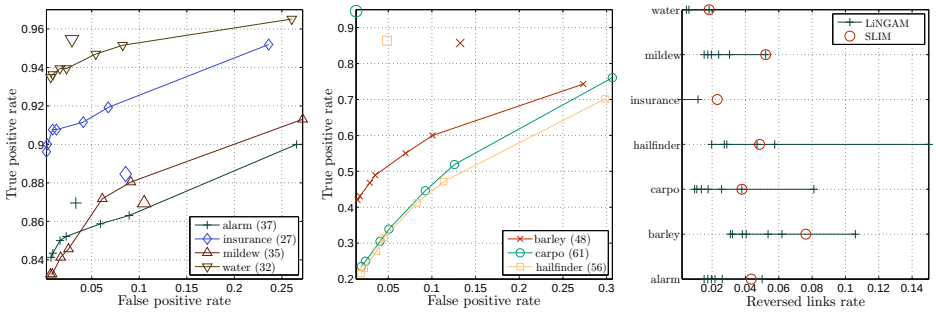


Figure 8: Ground truth and estimated structures. (a) Ground truth mixing matrix. (b) Estimated mixing matrix using our sparse factor model. Note the sign ambiguity in some of the columns. (c) First 50 (out of 92) ordering candidates produced by our method during inference and their frequency, the first m_{top} candidates were used for to learn DAGs. (d) Ground truth DAG. (e) Top candidate estimated using SLIM. (f) Estimated median weights for the DAG including 95% credible intervals and ground truth (squares). (g) Summary of link probabilities measured as $\eta_{ij} = p(r_{ij} = 1|\mathbf{X}, \cdot)$.



(a) (b) (c)

Figure 9: Performance measures for the Bayesian networks repository experiments. Each connected marker correspond to a different p-value in LiNGAM, starting left to right from 0.005. Disconnected markers denote SLIM results. Numbers in parentheses indicate number of variables.

Figure 8(f) we see that the elements of $\mathbf{B}$ are correctly estimated and their credible intervals are small, mainly due to the lack of model mismatch. Figure 8(g) shows a good separation between zero and non-zero elements of $\mathbf{B}$ as summarized by $p(r_{ij} = 1|\mathbf{X}, \cdot)$. It is worthwhile mentioning that using $\beta_m = 0.99$ instead of $\beta_m = 0.1$ in this example, still produces the right DAG, although the separation between zero and non-zero elements in Figure 8(g) will be smaller and with higher uncertainty, i.e. larger credible intervals.

6.2 Bayesian networks repository

Next we want to compare our method against LiNGAM on some realistic structures. We consider 7 well known benchmark structures from the Bayesian network repository[4], namely alarm, barley, carpo, hailfinder, insurance, mildew and water ($d = 37$, 48, 61, 56, 27, 35, 32 respectively). Since we do not have continuous data for any of the structures, we generated 10 datasets of size $N = 500$ for each of them using heavy-tailed distributions with different parameters and equation (1) with $m = 0$, in a similar way as we did for the previous set of experiments, with $\mathbf{R}$ set to the ground truth and $\mathbf{B}$ from $\text{sign}(\mathcal{N}(0,1)) + \mathcal{N}(0,0.2)$. For LiNGAM, we only use Wald statistics because as seen in the previous experiment, it performs significantly better that bootstrapping. Again, we estimate models for different p-value cutoffs (0.0005, 0.001, 0.005, 0.01, 0.05, 0.1 and 0.5). For SLIM, we set $\beta_m = 0.1$ since all the networks in the repository are sparse. Figures 9(a), 9(b) and 9(c) show averaged performance measures respectively as ROC curves and the proportion of links reversed in the estimated model due to ordering errors.

In this case, the results are mixed when looking at the performances obtained. Figure 9(b) shows that SLIM is better than LiNGAM in the larger datasets with a significant difference. Figure 9(a) shows for the remaining four datasets, that LiNGAM is better in

4. Network structures available at `http://compbio.cs.huji.ac.il/Repository/`.

two cases corresponding to the insurance and mildew networks. In general, both methods perform reasonably well given the size of the problems and the amount of data used to fit the models. However, SLIM tends to be more stable, when looking at the range of the true positive rates. It is important to note that the best and worst case for SLIM correspond to the largest and smallest network, respectively. We do not have a sensible explanation about why SLIM is performing that poorly on the insurance network. Figure 9(c) implicitly reveals that both methods are unable to find the right ordering of the variables.

We also tried the following methods with encoded Gaussian assumptions: standard DAG search, order search, sparse candidate pruning then DAG search (Friedman et al., 1999), L1MB then DAG search (Schmidt et al., 2007), and sparse candidate pruning then order search (Teyssier and Koller, 2005). We observed (results not shown) that these methods produce similar results to those obtained by either LiNGAM or SLIM when only looking at the resulting undirected graph, i.e. removing the directionality of the links. Evaluation of directionality in Gaussian models is out of the question because such methods can only find DAGs up to Markov equivalence classes, thus evaluation must be made using partially directed acyclic graphs (PDAGs). It is still possible to modify some of the methods mentioned above to handle non-Gaussian data by for instance using some other appropriate conditional independence tests, however this is out of the scope of this paper.

6.3 Model comparison

In this experiment we want to evaluate the model selection procedure described in Section 4. For this purpose we have generated 1000 different datasets/models with $d = 5$ and $N = \{500, 1000\}$ following the same procedure described in the first experiment, but this time we selected the true model to be either a factor model or a DAG with equal probability. In order to generate a factor model, we basically just need to ensure that $\mathbf{D}$ cannot be permuted to a triangular form, so the data generated from it does not admit a DAG representation. We kept 20% of the data to compute the predictive densities to then select between all estimated DAG candidates and the factor model. We found that for $N = 500$ our approach was able to select true DAGs 96.78% of the times and true factor models 87.05%, corresponding to an overall accuracy of 91.9%. Increasing the number of observations, i.e. for $N = 1000$, the true DAG, true factor model rates and overall error increased to 98.99%, 95.0% and 96.99%, respectively. Figure 10 shows separately the empirical log-likelihood ratio distributions obtained from the 1000 datasets for DAGs and factor models. The shaded areas correspond to the true DAG/factor model regions, with zero as their boundary. Note that when the wrong model is selected the likelihood ratio is nicely close to the boundary and the overlap of the two distributions decreases with the number of observations used, since the quality of the predictive density increases accordingly. The true DAG rates tend to be larger than for factor models because it is more likely that the latter is confused with a DAG due to estimation errors or closeness to a DAG representation, than a DAG being confused with a factor model which is naturally more general. This is precisely why the likelihood ratios tend to be larger on the factor model side of he plots. All in all, these results demonstrate that our approach is very effective at selecting the true underlying structure when the data is generated by one of the two hypotheses.

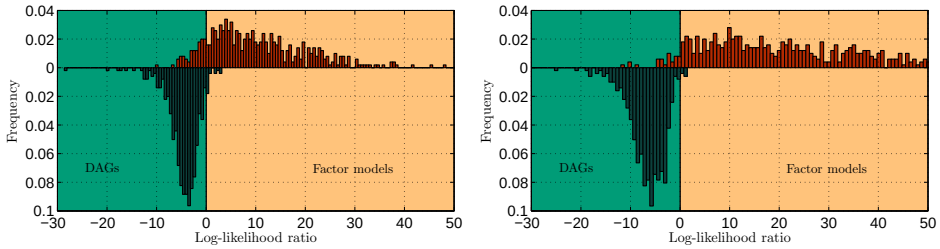


Figure 10: Log-likelihood ratio empirical distributions for, (a) $N = 500$ and (b) $N = 1000$. Top bars correspond to true factor models, bottom bars to true DAGs and the ratio is computed as described in Section 4. Top bars lying below zero are true factor models predicted to be better explained by DAGs, thus model comparison errors.

6.4 DAGs with latent variables

We will start by illustrating the identifiability issues of the model in equation (1) discussed in Section 2.1 with a very simple example. We generated $N = 500$ observations from the graph in Figure 3(b) and kept 20% of the data to compute test likelihoods. Now, we perform inference on two slightly different models, namely, (u) where $\mathbf{z}' = [z_1' \; z_2' \; z_L']$ is provided with Laplace distributions with unit variance, i.e. $\lambda = 2$, and (i) where z_1, z_2 have Laplace distributions with unit variance and z_L is Cauchy distributed. We want to show that even if both models match the true generating process, (u) is non-identifiable whereas (i) can be successfully estimated. In order to keep the experiment controlled as much as possible, we set $\beta_m = 0.99$ to reflect that the ground truth is dense and we did not infer $\mathbf{C}_D$ and set it to the true values, i.e. the identity. Then, we ran 10 independent chains for each one of the models and summarized $\mathbf{B}$, $\mathbf{C}_L$, $\mathbf{D}$ and the test likelihoods in Figure 11.

Figure 11(a) shows that model (u) finds the DAG in Figure 3(b) (the ground truth) in 3 cases, and in the remaining 7 cases it finds the DAG in Figure 3(a). Note also that the test likelihoods in Figure 11(c) are almost identical, as must be expected due to the lack of identifiability of the model, so they cannot be used to select among the two alternatives. Model (i) finds the right structure all the times as shown in Figure 11(d). The mixing matrix of the equivalent factor model, $\mathbf{D}$ is shown in Figures 11(b) and 11(e) for (u) and (i), respectively. In Figure 11(b), the first and third column of $\mathbf{D}$ exchange positions because all the components of $\mathbf{z}$ have the same distribution, which is not the case of Figure 11(e). The small quantities in $\mathbf{D}$ are due to estimation errors when computing $b_{21}c_{1L} + c_{2L}$, and this cancels out in the true model. The sign changes in Figures 11(a) and 11(d) are caused by the sign ambiguity of $\mathbf{z}_L$ in the product $\mathbf{C}_L\mathbf{z}_L$. We also tested the alternative model in Figure 3(b) obtaining equivalent results, i.e. 4 successes for model (u) and 10 for model (i). This small example shows how non-identifiability may lead to two very different DAG solutions with distinct interpretations of the data.

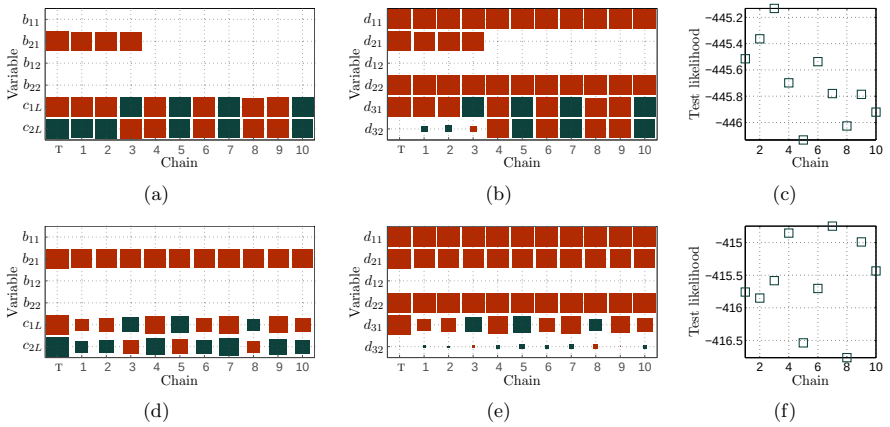


Figure 11: Identifiability experiment for the DAG with latent variables. Connectivities $\mathbf{B}$ and $\mathbf{C}_L$ are shown for (u) in (a) and (i) in (d). Equivalent mixing matrix $\mathbf{D}$ for (u) in (b) and for (i) in (d). Test likelihoods for (u) and (i) are shown in (c) and (f) respectively. The first column in (a,b,d,e) denoted as T is the ground truth. Dark and light boxes are negative and positive numbers, accordingly.

Hoyer et al. (2008) recently presented an approach to DAGs with latent variables based on LiNGAM (Shimizu et al., 2006). Their procedure uses probabilistic ICA and bootstrapping to infer the equivalent factor model distribution $p(\mathbf{D}|\mathbf{X})$, then greedily selects m columns of $\mathbf{D}$ to be latent variables until the remaining ones can be permuted to triangular and the resulting DAG is compatible with the faithfulness assumption (see, Pearl, 2000). If we assume that their procedure is able to find the exact $\mathbf{D}$ for the graphs in Figures 3(a) and 3(b), due to the faithfulness assumption, the DAG in Figure 3(a) will be always selected regardless of the ground truth[5]. In practice, the solution obtained for $\mathbf{D}$ is dense and needs to be pruned, hence we rely on $p(\mathbf{X}, \mathbf{D})$ being larger for the ground truth in Figure 3(b) than for the graph in Figure 3(a), however since both models differ only by a permutation of the columns of $\mathbf{D}$, they have exactly the same joint density $p(\mathbf{X}, \mathbf{D})$ — they are non-identifiable, thus the algorithm will select one of the options by chance. Since the source of non-identifiability of their algorithm is permutations of columns of $\mathbf{D}$, it does not matter if probabilistic ICA match or not the distribution of the underlying process as in our model. Anyway, we decided to try models (u) and (i) described above using the algorithm just described [6]. Regardless of the ground truth, Figures 3(a) or 3(b), the algorithm always selected the DAG in Figure 3(b), which in this particular case is due to $p(\mathbf{X}, \mathbf{D})$ being slightly larger for the denser model.

5. See Robins et al. (2003) for a very interesting explanation of faithfulness using the same example presented here.
6. Matlab package (v.1.1) freely available at `http://www.cs.helsinki.fi/group/neuroinf/lingam/`.

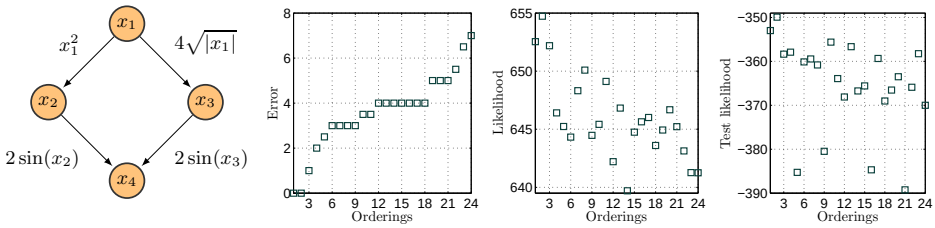
Now we test the model in a more general setting. We generate 100 models and datasets of size $N = 500$ using a similar procedure to the one in the artificial data experiment. The models have $d = 5$ and $m = 1$, no dense structures are generated and the distributions for $\mathbf{z}$ are heavy-tailed, drawn from a generalized Gaussian distribution with random shape. For SLIM, we use the following settings, $\beta_m = 0.1$, $\mathbf{z}_D$ is Laplace with unit variances and $\mathbf{z}_L$ is Cauchy. Furthermore, we have doubled the number of iterations of the DAG sampler, i.e. 6000 samples and a burn-in period of 2000, so as to compensate for the additional parameters that need to be inferred due to inclusion of latent variables. Our ordering search procedure was able to find the right ordering 78 out of 100 times. The true positive rates, true negative rates and median AUC are 88.28%, 96.40% and 0.929, respectively, corresponding to approximately 1.5 structure errors per network. Using Hoyer et al. (2008) we obtained 1 true ordering out of 100, 91.63% true positive rate, 65.18% true negative rate and 0.800 median AUC, showing again the preference of the algorithm for denser models. We regard these results as very satisfactory for both methods considering the difficulty of the task and the lack of identifiability of the model by Hoyer et al. (2008).

6.5 Non-linear DAGs

For Sparse Non-linear Identifiable Modeling (SNIM) described in Section 3.5, first we want to show that our method can find and select from DAGs with non-linear interactions. We used the artificial network from Hoyer et al. (2009) shown here in Figure 12(a) and generated 10 different datasets corresponding to $N = 100$ observations, each time using driving signals sampled from different heavy-tailed distributions. Since we do not yet have an ordering search procedure for non-linear DAGs, we perform DAG inference for all possible orderings and datasets. The results obtained are evaluated in two ways, first we check if we can find the true connectivity matrix when the ordering is correct. Second, we need to validate that the likelihood is able to select the model with less error and correct ordering among all possible candidates so we can use it in practice. Figures 12(b), 12(c) and 12(d) show the median errors, training and test likelihoods (using 20% of the data) for each one of the orderings, respectively. In this particular case we only have two correct orderings, namely, $(1, 2, 3, 4)$ and $(1, 3, 2, 4)$, corresponding to the first and second candidates in the plots. Figure 12(b) shows that the error is zero only for the two correct orderings, then our model is able to infer the structure once the right ordering is given as desired. As a result of the identifiability, data and test likelihoods shown in Figures 12(c) and 12(d) correlate nicely with the structural error in Figure 12(b). This means that we can use use the likelihoods as a proxy for the structural error just as in the linear case.

We also tested the network in Figure 12(a) using three non-linear structure learning procedures namely greedy standard hill-climbing DAG search, the "ideal parent" algorithm (Elidan et al., 2007) and kernel PC (Tillman et al., 2009). The first two methods use a scaled sigmoid function to capture the non-linearities in the data. In particular, they assume that a variable x can be explained as scaled sigmoid transformation of a linear combination of its parents. The best median result we could obtain after tuning the parameters of the algorithms was 2 errors and 2 reversed links[7]. Both methods perform similarly in this

7. Maximum number of iterations, random restarts to avoid local minima, regularization of the non-linear regression and the number of ranking candidates in ideal parent algorithm.

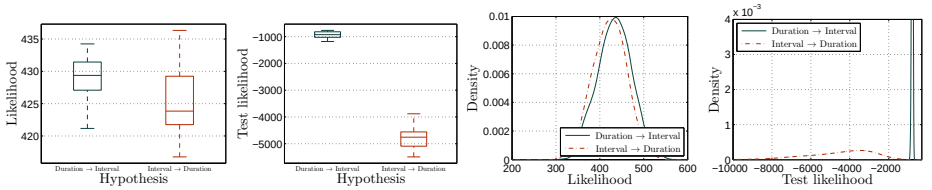


(a) (b) (c) (d)

Figure 12: Non-linear DAG artificial example. (a) Network with non-linear interactions between observed nodes used as ground truth. (b,c,d) Median error, likelihood and test likelihood for all possible orderings and 10 independent repetitions. The plots are sorted according to number of errors and only the first two are valid according to the ground truth in (a), i.e. $(1, 2, 3, 4)$ and $(1, 3, 2, 4)$. Note that when the error is zero in (b) the likelihoods are larger with respect to the remaining orderings in (c) and (d).

particular example, the only significant difference being their computational cost, which is considerably smaller for the "ideal parent" algorithm, as it was also pointed out by Elidan et al. (2007). The reason why we consider these algorithms do not perform well here is that the sigmoid function can be very limited at capturing certain non-linearities due to its parametric form whereas the nonparametric GP gives flexible non-linear functions. The third method uses non-linear independence tests together with non-linear regression (relevance vector machines) and the PC algorithm to produce mixed DAGs. The best median result we could get in this case was 2 errors, 0 reversed links and 1 bidirectional links. These three non-linear DAG search algorithms have the great advantage of not requiring exhaustive enumeration of the orderings as our method and others available in the literature. Zhang and Hyvärinen (2009) provides theoretical evidence of the possibility for flexible non-linear modeling without exhaustive order search but not a way to do it in practice. Yet another possibility not tried here will be to take the best parts of both strategies by taking the outcome of the non-linear DAG search algorithm and refine it using a nonparametric method like SNIM. However, it is not entirely clear how the non-linearities can affect the ordering of the variables. In the remaining part of this section we only focus on tasks for pairs of variables where the ordering search is not an issue.

The dataset known as Old Faithful (Asuncion and Newman, 2007) contains 272 observations of two variables measuring waiting time between eruptions and duration of eruptions for the Old Faithful geyser in Yellowstone National Park, USA. We want to test the two possible orderings, duration $\rightarrow$ interval and interval $\rightarrow$ duration. Figures 13(a) and 13(b) show training and test likelihood boxplots for 10 independent randomizations of the dataset with 20% of the observations used to compute test likelihoods. Our model was able to find the right ordering, i.e. duration $\rightarrow$ interval in all cases when the test likelihood was used but only 7 times with the training likelihood due to the proximity of the densities, see Figure 13(c). On the other hand, the predictive density is very discriminative, as shown for instance



(a) (b) (c) (d)

Figure 13: Testing {duration, interval} in Old Faithful dataset. (a,b) Data and test likelihood boxplots for 10 independent repetitions. (c,d) Training and test likelihood densities for one of the repetitions. The test likelihood separates consistently the two tested hypotheses.

in Figure 13(d). This is not a very surprising result since making the duration a function of the interval results in a very non-linear function, whereas the alternative function is almost linear (data not shown).

Abalone is one of the datasets from the UCI ML repository (Azzalini and Bowman, 1990). It is targeted to predict the age of abalones from a set of physical measurements. The dataset contains 9 variables and 4177 observations. First we want to test the pair {age, length}. For this purpose, we use 10 subsets of $N = 200$ observations to build the models and compute likelihoods just as before. Figures 14(a) and 14(b) show training and test likelihoods respectively as boxplots. Both training and test likelihoods pointed to the right ordering in all 10 repetitions. In this experiment, the separation of the densities for the two hypotheses considered is very large, making age $\rightarrow$ length significantly better supported by the data. Figures 14(c) and 14(d) show predictive densities for one of the trials indicating again that age $\rightarrow$ length is consistently preferred. We also decided to try another three sets of hypotheses: {age, diameter}, {age, weight} and {age, length, weight} for which we found the right orderings $\{10, 10\}$, $\{10, 10\}$ and $\{10, 6\}$ out of 10 by looking at the training and the test likelihoods, respectively. In the model with three variables, increasing the number of observations used to fit the model from $N = 200$ to $N = 400$, increased the number of cases in which the test likelihood selected the true hypothesis from 6 to 8 times, which is more than enough to make a decision about the leading hypothesis.

To conclude this set of experiments we test SNIM against another three recently proposed methods[8], namely Non-linear Additive Noise (NAN) model (Hoyer et al., 2009), Post-Non-Linear (PNL) model (Zhang and Hyvärinen, 2009) and Informational Geometric Causal Inference (IGCI) (Daniusis et al., 2010), using an extended version of "cause-effect pairs" task for the NIPS 2008 causality competition[9] (Mooij and Janzing, 2010). The task consists on distinguishing the cause from the effect of 51 different pairs of observed variables. NAN and PNL rely on an independence test (HSIC, Hilbert-Schmidt Independence Criterion, Gretton et al., 2008) to decide which of the two variable is the cause. NAN was able to take 10 decisions all being accurate. PNL was accurate 40 times out of 42 decisions

8. Matlab packages available at `http://webdav.tuebingen.mpg.de/causality/`.
9. Data available at `http://webdav.tuebingen.mpg.de/cause-effect/`.

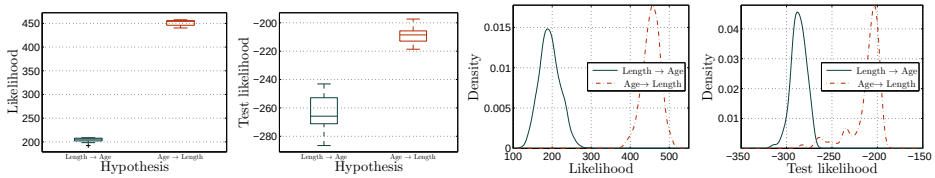


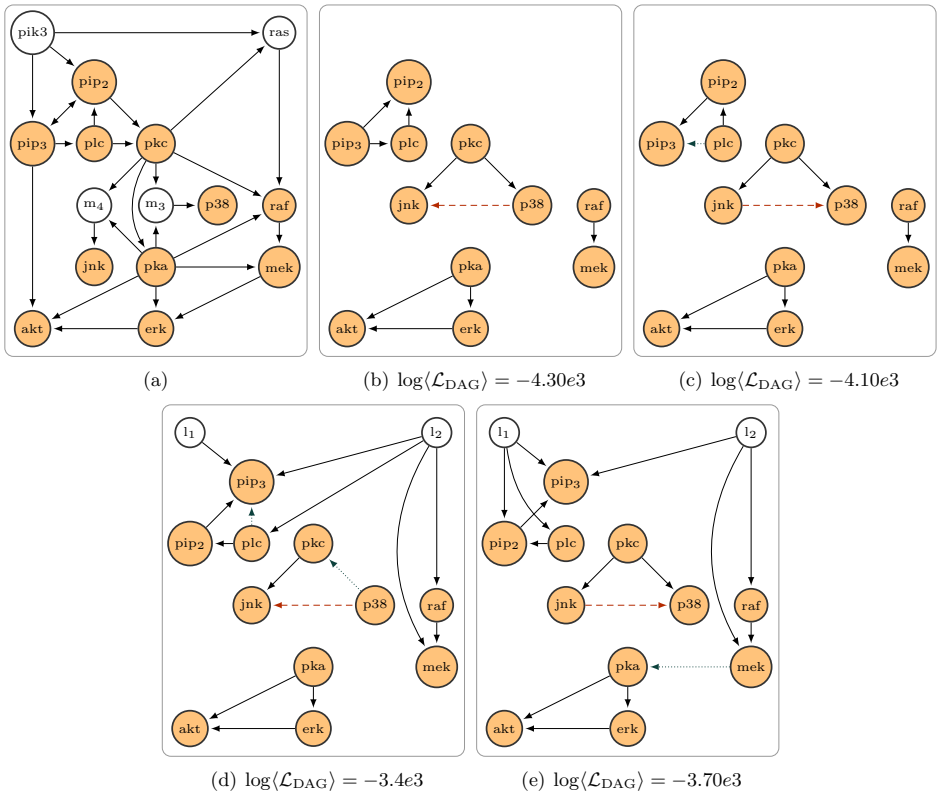
Figure 14: Testing {length, age} in Abalone dataset. (a,b) Data and test likelihood boxplots for 10 independent repetitions. (c,d) Training and test likelihood densities for one of the repetitions. The likelihoods largely separate the two tested hypotheses.

made. IGCI and SNIM obtained an accuracy of 40 and 39 pairs, respectively[10]. The results indicate (i) that NAN and PNL are very accurate when the independence test used is able to reach a decision and (ii) in terms of accuracy, the results obtained by PNL, IGCI and SNIM are comparable. For SNIM we decide based upon the test likelihood and for IGCI we used a uniform reference measure (rescaling the data between 0 and 1). From the four tested methods we can identify two main trends. One is to explicitly model the data and decide the cause-effect direction using independence tests or test likelihoods like in NAN, PNL and SNIM. The second is to directly define a measure for directionality as in IGCI. The first option has the advantage of being able to convey more information about the data at hand whereas the second option is orders of magnitude faster than the other three because it only tests for directionality.

6.6 Protein-signaling network

This experiment demonstrates a typical application of SLIM in a realistic biological large N, small d setting. The dataset introduced by Sachs et al. (2005) consists of flow cytometry measurements of 11 phosphorylated proteins and phospholipids (raf, erk, p38, jnk, akt, mek, pka, pkc, pip_2, pip_3, plc). Each observation is a vector of quantitative amounts measured from single cells. Data was generated from a series of stimulatory cues and inhibitory interventions. Hence the data is composed of three kinds of perturbations: general activators, specific activators and specific inhibitors. Here we are only using the 1755 observations — clearly non-Gaussian, e.g. see Figure 16(a), corresponding to general stimulatory conditions. It is clear that using the whole dataset, i.e. using specific perturbations, will produce a richer model, however handling interventional data is out of the scope of this paper mainly because handling that kind of data with a factor model is not an easy task. Thus our current order search procedure is not appropriate. Focused only on the observational data, we want to test all the possibilities of our model in this dataset, namely, standard factor models, pure DAGs, DAGs with latent variables, non-linear DAGs and quantitative model comparison using test likelihoods. The textbook DAG structure taken from Sachs et al. (see Figure 2

10. Results for NAN, PNL and IGCI were taken from Daniusis et al. (2010) because we were unable to entirely reproduce their results with the software provided by the authors.



(a) (b) $\log\langle\mathcal{L}_{\text{DAG}}\rangle = -4.30e3$ (c) $\log\langle\mathcal{L}_{\text{DAG}}\rangle = -4.10e3$

(d) $\log\langle\mathcal{L}_{\text{DAG}}\rangle = -3.4e3$ (e) $\log\langle\mathcal{L}_{\text{DAG}}\rangle = -3.70e3$

Figure 15: Result for protein-signaling network data. (a) Textbook signaling network as reported in Sachs et al. (2005). Estimated structure using SLIM: (b) using the true ordering, (c) obtaining the ordering from the stochastic search, (d) top DAG with 2 latent variables and (e) the runner-up (in test likelihood). False positives are shown in red dashed lines and reversed links in green dotted lines. Below each structure we also report the median test likelihood (larger is better).

and Table 3, 2005) is shown in Figure 15(a) and the models are estimated using the true ordering and SLIM in Figures 15(b) and 15(c), respectively.

The DAG found using the right ordering of the variables shown in Figure 15(b) turned out to be the same structure found by the discrete Bayesian network from Sachs et al. (2005) without using interventional data (see supplementary material, Figure 4(a)), with one important difference: the method presented by Sachs et al. (2005) is not able to infer the directionality of the links in the graph without interventional data, i.e. their resulting graph is undirected. SLIM in Figure 15(c) finds a network almost equal to the one in Figure 15(b) apart from one reversed link, plc $\rightarrow$ pip3. Surprisingly this was also found reversed

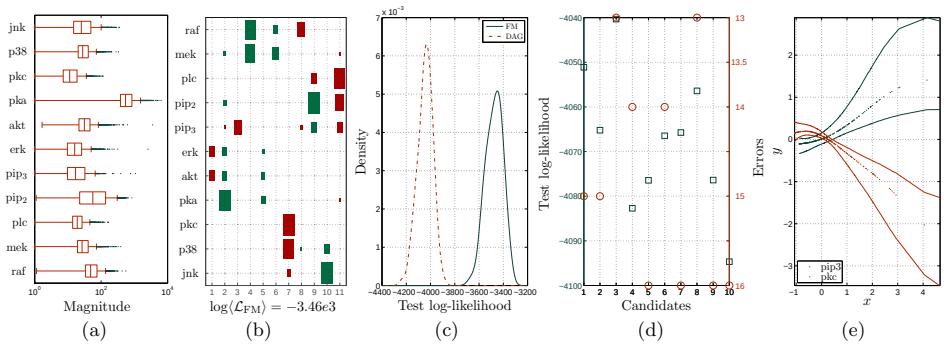


Figure 16: Results for protein-signaling network data. (a) Boxplot for each one of the 11 variables in the dataset. (b) Estimated factor model. (c) Test likelihoods for the best DAG (dashed) and the factor model (solid). (d) Test likelihoods (squares) and structure errors (circles) included reversed links for all candidates. (e) Non-linear variables y obtained as a function of the observed variables x for pip3 and pkc. Each dot in the plot is an observation and the solid lines are 95% credible intervals.

by Sachs et al. (2005) using interventional data. In addition, there is just one false positive, the pair {jnk, p38}, even with a dedicated latent variable in the factor model mixing matrix shown in Figure 16(b), thus we cannot attribute such a false positive to estimation errors. A total of 211 ordering candidates were produced during the inference out of approximately 10^7 possible and only $m_{\text{top}} = 10$ of them were used in the structure search step. Note from Figure 16(d) that the predictive densities for the DAGs correlate well with the structural accuracy, apart from candidate 8. Candidates 3 and 8 have the same number of structural errors, however candidate 8 has 3 reversed links instead of 1 as shown in Figure 15(c). The predictive densities for the best candidate, third in Figure 16(d) are shown in Figure 16(c) and suggest that the factor model fits the data better. This makes sense considering that estimated DAG in Figure 15(c) is a substructure of the ground truth. We also examined the estimated factor model in Figure 16(b) and we found that several factors could correspond respectively to three unmeasured proteins, namely pi3k in factors 9 and 11, m_3 (mapkkk, mek4/7) and m_4 (mapkkk, mek3/6) in factor 7, ras in factors 4 and 6.

We also wanted to assess the performance of our method and several others using this dataset, including LiNGAM and those mentioned in the Bayesian network repository experiment, even knowing that this dataset contains non-Gaussian data. We found that all of them have similar results in terms of true and false positive rates when comparing them to SLIM. However the number of reversed links was not in any case less than 6, which corresponds to more than 50% of the true positives found in every case. This means that they are essentially able to find the skeleton in Figure 15(b). Besides, we do not have knowledge of any other method for DAG learning using only the observational data that also provides results substantially better than the ones shown in Figure 15(c). The poor performance of

LiNGAM is difficult to explain but the large amount of reversed links may be due to the FastICA based deterministic ordering search procedure.

We also tried DAG models with latent variables in this dataset. The results obtained by the DAG with 2 a priori assumed latent variables are shown in Figures 15(d) and 15(e), corresponding to the first and second DAG candidates in terms of test likelihoods. The first option is different to the pure DAG in Figure 15(c) only in the reversed link, p38 $\rightarrow$ pkc, but captures some of the behavior of pik3 and ras in l_1 and l_2 respectively. It is very interesting to see how, due to the link between pik3 and ras that is not possible to model with our model, the second inferred latent variable is detecting signals pointing towards pip_2 and plc. We also considered a second option because l_1 in the top model is only connected to a single variable pip_3 and thus could be regarded as an estimation error since it can be easily confounded with a driving signal. Comparing Figures 15(c) and 15(e) reveals two differences in the observed part, a false negative $pip_3 \rightarrow$ plc and a new true (reversed) positive mek $\rightarrow$ pka. This candidate is particularly interesting because the first latent variable captures the connectivity of pik3 while connecting itself to plc due to the lack of connectivity between pip_3 and plc. Moreover, the second latent variable resembles ras and the link between pik3 and ras as a link from itself to pip_3. In both solutions there is a connection between l_2 and mek that might be explained as a link through a phosphorylation of raf different to the observed one, i.e. ras_{s259}. In terms of median test likelihoods, the model in Figure 15(d) is only marginally better than the factor model in Figure 16(b) and in turn marginally worse than the DAG in Figure 15(e).

For SNIM we started from the true ordering of the variables but we could not find any improvement compared to the structure in Figure 15(c). In particular there are only two differences, plc $\rightarrow pip_2$ and jnk $\rightarrow$ p38 are missing, meaning that at least in this case there are no false positives in the non-linear DAG. Looking at the parameters of the covariance function used, $\boldsymbol{\upsilon}$ (not shown) with acceptance rates of approximately $\approx 20\%$ and reasonable credible intervals, we can say that our model found almost linear functions since all the parameters of the covariance functions are rather small. Figure 16(e) shows two particular non-linear variables learned by the model, corresponding to pip3 and plc. In each case the uncertainty of the estimation nicely increases with the magnitude of the observed variable and although the functions are fairly linear they resemble the saturation effect we can expect in this kind of biological data. From the three non-linear methods non-requiring exhaustive order search described in the previous section (DAG search, "ideal parent" and kPC), the best result we obtained was 11 structural errors, 10 true positives, 34 true negatives, 2 reversed and 6 bidirectional links for kPC vs 12, 9, 34, 1 and 0 by SLIM and 12, 8, 35, 0 and 0 by SNIM.

6.7 Time series data

We illustrate the use Correlated Sparse Linear Identifiable Modeling (CLSIM) on the dataset introduced by Kao et al. (2004) consisting of temporal gene expression profiles of *E. coli* during transition from glucose to acetate measured using DNA microarrays. Samples from 100 genes were taken at 5, 10, 15, 30, 60 minutes and every hour until 6 hours after transition[11]. The general goal is to reconstruct the unknown transcription factor activities from the ex-

11. Data available at `http://www.seas.ucla.edu/~liaoj/NCA_module_Data`.

pression data and some prior knowledge. In Kao et al. (2004) the prior knowledge consisted of taking the set of transcription factors (ArcA, CRP, CysB, FadR, FruR, GatR, IcIR, LeuO, Lrp, NarL, PhoB, PurB, RpoE, RpoS, TrpR and TyrR) controlling the observed genes and the (up-to-date) connectivity between genes and transcription factors from RegulonDB[12] (Gama-Castro et al., 2008). From this setting, we can immediately relate the transcriptions factors with $\mathbf{Z}$, such a connectivity with $\mathbf{Q}_L$, and their relative strengths with $\mathbf{C}_L$, hence the problem can be seen as a standard factor model. In Kao et al. (2004) they applied a method called Network Component Analysis (NCA), that uses a least-squares based algorithm to solve a problem similar to the one in equation (1), but assuming that the sparsity pattern (masking matrix $\mathbf{Q}_L$) of $\mathbf{C}_L$ is fixed and known. It is well-known that the information in RegulonDB is still incomplete and hard to obtain for organisms different than *E. coli*. Our goal here is thus to obtain similar transcription factor activities to those found by Kao et al. (2004) without using the information from RegulonDB, but taking into account that the data at hand is a time series by letting each transcription factor activity have an independent Gaussian process prior as described for CSLIM in Section 3.4. We will not attempt to use $\mathbf{Q}_L$ to recover the ground truth connectivity information since RegulonDB is collected from a wide range of experimental conditions and not only from the transcriptional activity produced by the *E. coli* during its transition from glucose to acetate. The results are shown in Figure 17.

Results in Figure 17(e) show the source matrix $\mathbf{Z}$ recovered by our model together with those from NCA[13]. In this experiment we ran a single chain and collected 6000 samples after a burn-in period of 2000 samples (approximately 10 minutes in a desktop machine). Most of the profiles obtained by our method are similar to those obtained by NCA (Kao et al., 2004). We ran two versions of our model, one with $\mathbf{Q}_L$ fixed to the RegulonDB values, i.e. similar in spirit to NCA, and another when we infer $\mathbf{Q}_L$ without any restriction. The results of NCA and our model with fixed $\mathbf{Q}_L$ are directly comparable (up to scaling) whereas we had to match the permutation $\mathbf{P}_{\mathrm{f}}$ of the unrestricted model to those found by NCA in order to compare, using the Hungarian algorithm. Figure 17(a) shows the mixing matrices obtained by NCA and our two models. Figures 17(a) and 17(b) are very similar due to the restriction imposed on $\mathbf{Q}_L$. The mixing matrix obtained by our unrestricted model in Figure 17(c) is clearly denser than the other two, suggesting that there are different ways of connecting genes and transcription factors and still reconstruct the transcription factor activities given the observed gene expression data. When looking to the test log-likelihood densities obtained by our two models in Figure 17(d) they are very similar, which suggests that there is no evidence that one of the models makes a better fit on test data. In terms of Mean Squared Error (MSE), NCA obtained 0.0146 while our model reached 0.0264 and 0.0218 on the restricted and unrestricted models, respectively, when using 90% of the data for inference. In addition, the 95% credible intervals for the MSE were $(0.0231, 0.0329)$ and $(0.0164, 0.0309)$ respectively. The latter shows again that there is no evidence that one of the three models is better than the other two, considering that: (i) NCA is trained on the entire dataset and (ii) our unrestricted model could, in principle, produce mixing matrices arbitrarily denser than the connectivity matrix extracted from RegulonDB, and thus, again in principle, lower MSE values.

12. http://regulondb.ccg.unam.mx/.
13. Matlab package (v.2.3) available at http://www.seas.ucla.edu/~liaoj/download.htm.

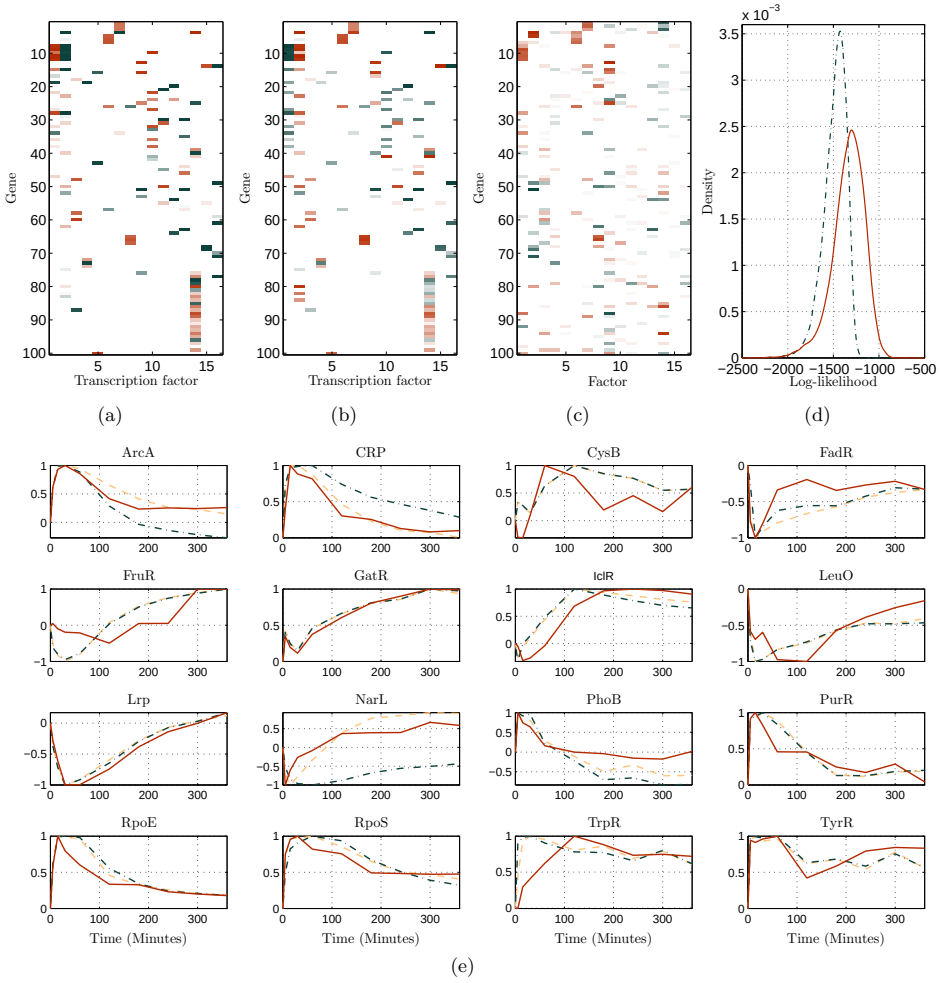


Figure 17: Results for *E. coli* dataset. Mixing matrices estimated using: (a) NCA, (b) our formulation when restricting $\mathbf{Q}_L$ using RegulonDB information and (c) the factor model. (d) Model comparison results using test likelihoods. The restricted model (dash-dotted line) obtained a median negative log-likelihood of 1463.4 whereas the unrestricted model (solid line) obtained 1317.1, suggesting no significant model preferences. (e) Estimated transcription factor activities, $\mathbf{Z}$. Our methods (solid and dash-dotted lines for unrestricted and restricted model respectively) produce similar results to those produced by NCA (dashed line).

7. Discussion

We have proposed a novel approach called SLIM (Sparse Linear Identifiable Multivariate modeling) to perform inference and model comparison of general linear Bayesian networks within the same framework. The key ingredients for our Bayesian models are slab and spike priors to promote sparsity, heavy-tailed priors to ensure identifiability and predictive densities (test likelihoods) to perform the comparison. A set of candidate orderings is produced by stochastic search during the factor model inference. Subsequently, a linear DAG with or without latent variables is learned for each of the candidates. To the authors' knowledge this is the first time that a method for comparing such closely related linear models has been proposed. This setting can be very beneficial in situations where the prior evidence suggests both DAG structure and/or unmeasured variables in the data. We also show that the DAG with latent variables can be fully identifiable and that SLIM can be extended to the non-linear case (SNIM - Sparse Non-linear Identifiable Multivariate modeling), if the ordering of the variables is provided or can be tested by exhaustive enumeration. For example in the protein-signaling network (Sachs et al., 2005), the textbook ground truth suggests both DAG structure and a number of unmeasured proteins. The previous approach (Sachs et al., 2005) only performed structure learning in pure DAGs but our results using observational data alone suggest that the data is better explained by a (possibly non-linear) DAG with latent variables. Our extensive results on artificial data showed one by one the features of our model in each one of its variants, and demonstrated empirically their usefulness and potential applicability. When comparing against LiNGAM, our method always performed at least as well in every case with a comparable computational cost. The presented Bayesian framework also allows easy extension of our model to match different prior beliefs about the problems at hand without significantly changing the model and its conceptual foundations, as in CSLIM and SNIM.

We believe that the priors that give raise to sparse models in the fully Bayesian inference setting, like the two-level slab (continuous) and spike (point-mass in zero) priors used are very powerful tools for simultaneous model and parameter inference. They may be useful in many settings in machine learning where sparsity of parameters is desirable. Although the posterior distributions for slab and spike priors will be non-convex, it is our experience that inference with blocked Gibbs sampling actually has very good convergence properties. In the two-level approach, one uses a hierarchy of two slab and spike priors. The first is on the parameter and the second is on the mixture parameter (i.e. the probability that the parameter is non-zero). Instead of letting this parameter be controlled by a single Beta-distribution (one level approach) we have a slab and spike distribution on it with a Beta-distributed slab component biased towards one. This makes the model more parsimonious, i.e. the probability that parameters are zero or non-zero is closer to zero and one and parameter settings are more robust.

In the following we will discuss open questions and future directions. From the Bayesian network repository experiment it is clear that we need to improve our ordering search procedure if we want to use SLIM for problems with more than say 50 variables. This basically amounts to finding proposal distributions that better exploit the particularities of the model at hand. Another option could be to provide the proposal distribution with some

notion of memory to avoid permutations with low probability and/or expand the coverage of the searching procedure.

It is well studied in the literature on sparse models that for increasing number of observations any model tends to loose its sparsity capabilities. This is because the likelihood starts dominating the inference, making the prior distribution less informative. The easiest way to handle such an effect is to make the hyperparameters of the sparsity prior dependent on N. We have not explored this phenomenon in SLIM but it should certainly be taken into account in the specification of sparsity priors.

Directly specifying the distributions of the latent variables in order to obtain identifiability in the general DAG with latent variables requires having different distributions for the driving signals of the observed variables and latent variables. This may introduce model mismatch or be restrictive in some cases as one will not have this kind of knowledge a priori. We thus need more principled ways to specify distributions for $\mathbf{z}$ ensuring identifiably, without restricting some of its components to having a particular behavior, like having heavier tails than the driving signals for instance. We conjecture that providing $\mathbf{z}$ with a parameterization of Dirichlet process priors with appropriate base measures would be enough but we are not certain whether this would be sufficient in practice.

We set a priori that the components of $\mathbf{z}$ are independent. Although this is a very reasonable assumption, it does not allow for connectivity between latent variables as we see for example in the protein signaling network, see Figure 15(a). It is straight forward to specify such a model, although identifiability becomes even harder to ensure in this case.

We do not have an ordering search procedure for the non-linear version of SLIM. This is a necessary step since exhaustive enumeration of all possible orderings is not an option beyond say 10 variables. The main problem is that the non-linear DAG has no equivalent factor model representation so we cannot directly exploit the permutation candidates we find in SLIM. However, as long as the non-linearities are weak, one might in principle use the permutation candidates found in a factor model, i.e. the linear effects will determine the correct ordering of the variables.

SLIM cannot handle experimental (interventional) data, and consequently around 80% of the data from the Sachs et al. (2005) study is not used. It is well-established how to learn with interventions in DAGs (see Sachs et al., 2005). The problem remains of how to formulate effective inference with interventional data in the factor model.

Acknowledgments

We thank the editor and the three anonymous referees for their helpful comments and discussions that improved the presentation of this paper.

Appendix A. Gibbs sampling

Given a set of N observations in d dimensions, the data $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ and m latent variables, MCMC analysis is standard and can be implemented through Gibbs sampling. Note that in the following, $\mathbf{X}_{i:}$ and $\mathbf{X}_{:i}$ are rows and columns of $\mathbf{X}$, respectively, and i, j, n are indexes for dimensions, factors and observations, respectively. In the following

we describe the conditional distributions needed to sample from the standard factor model hierarchy. Below we will briefly discus the modifications needed for the DAG.

Noise variance We can sample each element of $\mathbf{\Psi}$ independently using

$$\psi_i^{-1}|\mathbf{X}_{i:}, \mathbf{C}_{i:}, \mathbf{Z}, \mathbf{V}_i, s_s, s_r \sim \text{Gamma}\left(\psi_i^{-1}\middle| s_s + \frac{N+d}{2}, s_r + u\right) , \tag{13}$$

where $\mathbf{V}_i$ is a diagonal matrix with entries τ_{ij} and

$$u = \frac{1}{2}(\mathbf{X}_{i:} - \mathbf{C}_{i:}\mathbf{Z})(\mathbf{X}_{i:} - \mathbf{C}_{i:}\mathbf{Z})^\top + \frac{1}{2}\mathbf{C}_{i:}\mathbf{V}_i^{-1}\mathbf{C}_{i:}^\top .$$

Factors The conditional distribution of the latent variables $\mathbf{Z}$ using the scale mixtures of Gaussians representation can be computed independently for each element of z_{jn} using

$$z_{jn}|\mathbf{X}_{:n}, \mathbf{C}_{:j}, \mathbf{Z}_{:n}, \mathbf{\Psi}, \upsilon_{jn} \sim \mathcal{N}(z_{jn}|\mathbf{C}_{:j}^\top \mathbf{\Psi}^{-1}\boldsymbol{\epsilon}_{\setminus jn}, u_{jn}) , \tag{14}$$

where $u_{jn} = (\mathbf{C}_{:j}^\top \mathbf{\Psi}^{-1}\mathbf{C}_{:j} + \upsilon_{jn}^{-1})^{-1}$ and $\boldsymbol{\epsilon}_{\setminus jn} = \mathbf{X}_{:n} - \mathbf{C}\mathbf{Z}_{:n}|_{z_{jn}=0}$. If the latent factors are Laplace distributed the mixing variances υ_{jn} have exponential distribution, thus the resulting conditional is

$$\upsilon_{jn}^{-1}|z_{jn}, \lambda \sim \text{IG}\left(\upsilon_{jn}^{-1}\middle|\frac{\lambda}{|z_{jn}|}, \lambda^2\right) ,$$

and for the Student's t, with corresponding gamma densities as

$$\upsilon_{jn}^{-1}|z_{jn}, \sigma^2, \theta \sim \text{Gamma}\left(\upsilon_{jn}^{-1}\middle|\frac{\theta+1}{2}, \frac{\theta}{2} + \frac{z_{jn}^2}{2\sigma^2}\right) ,$$

where $\text{IG}(\cdot|\mu, \lambda)$ is the inverse Gaussian distribution with mean μ and scale parameter λ (Chhikara and Folks, 1989).

Gaussian processes In practice, the prior distribution for each row of the matrix $\mathbf{Z}$ in CSLIM has the form $z_{j1}, \ldots, z_{jN} \sim \mathcal{N}(0, \mathbf{K}_j)$, where $\mathbf{K}_j$ is a covariance matrix of size $N \times N$ built using $k_{\upsilon_j,n}(n, n')$. The conditional distribution for $z_{j1}, \ldots, z_{jN}$ can be computed using

$$z_{j1}, \ldots, z_{jN}|\mathbf{X}, \mathbf{C}_{j:}, \mathbf{Z}_{\setminus j}, \mathbf{\Psi} \sim \mathcal{N}(z_{j1}, \ldots, z_{jN}|\mathbf{C}_{:j}^\top \mathbf{\Psi}^{-1}\boldsymbol{\epsilon}_{\setminus j}\mathbf{V}, \mathbf{V}) ,$$

where $\mathbf{Z}_{\setminus j}$ is $\mathbf{Z}$ without row j, $\mathbf{V} = (\mathbf{U} + \mathbf{K}_j^{-1})^{-1}$, $\mathbf{U}$ is a diagonal matrix with elements $\mathbf{C}_{:j}^\top \mathbf{\Psi}^{-1}\mathbf{C}_{:j}$ and $\boldsymbol{\epsilon}_{\setminus j} = \mathbf{X} - \mathbf{C}\mathbf{Z}|_{z_{j1},\ldots,z_{jN}=0}$. The computation of $\mathbf{V}$ can be done in a numerically stable way by rewriting $\mathbf{V} = \mathbf{K}_j - \mathbf{K}_j(\mathbf{U}^{-1} + \mathbf{K}_j)^{-1}\mathbf{K}_j$ and then using Cholesky decomposition and back substitution to obtain in turn $\mathbf{L}\mathbf{L}^\top = \mathbf{U}^{-1} + \mathbf{K}_j$ and $\mathbf{L}^{-1}\mathbf{K}_j$. The hyperparameters of the covariance function in equation (9) can be sampled using

$$\kappa|\boldsymbol{\upsilon}, k_s, k_r \sim \text{Gamma}\left(\kappa\middle| k_s + mu_s, k_r + \sum_{j=1}^{m} \upsilon_j\right) .$$

For the inverse length-scales we use Metropolis-Hastings updates with proposal $q(\upsilon_j^\star|\upsilon_j) = p(\upsilon_j^\star)$ and acceptance ratio

$$\xi_{\to\star} = \frac{\mathcal{N}(z_{j1},\ldots,z_{jN}|\mathbf{0},\mathbf{K}_j^\star)}{\mathcal{N}(z_{j1},\ldots,z_{jN}|\mathbf{0},\mathbf{K}_j)} \ ,$$

where $\mathbf{K}_j^\star$ is obtained using $k_{\upsilon_j^\star,n}(n,n')$. For SNIM, we only need to replace $\mathbf{C}$ by $\mathbf{B}$, $\mathbf{Z}$ by $\mathbf{Y} = [\mathbf{y}_1 \ \ldots \mathbf{y}_N]$ and $k_{\upsilon_j,n}(n,n')$ by $k_{\upsilon_i,x}(\mathbf{x},\mathbf{x}')$.

Mixing matrix In order to sample each c_{ij} from the conditional distribution of the matrix $\mathbf{C}$ we use

$$c_{ij}|\mathbf{X}_{i:},\mathbf{C}_{\setminus ij},\mathbf{Z}_{j:},\psi_i,\tau_{ij} \ \sim \mathcal{N}(c_{ij}|u_{ij}\boldsymbol{\epsilon}_{\setminus ij}\mathbf{Z}_{j:}^\top, u_{ij}\psi_i) \ , \tag{15}$$

where $u_{ij} = (\mathbf{Z}_{j:}\mathbf{Z}_{j:}^\top + \tau_{ij}^{-1})^{-1}$ and $\boldsymbol{\epsilon}_{\setminus ij} = \mathbf{X}_{i:} - \mathbf{C}_{i:}\mathbf{Z}|_{d_{ij}=0}$. Note that we only need to sample those c_{ij} for which $r_{ij} = 1$, i.e. just the slab distribution. Sampling from the conditional distributions for τ_{ij} can be done using

$$\tau_{ij}^{-1}|d_{jn},t_s,t_r \ \sim \text{Gamma}\left(\tau_{ij}^{-1}\middle| t_s + \frac{1}{2}, t_r + \frac{d_{ij}^2}{2\psi_i}\right) \ . \tag{16}$$

The conditional distributions for the remaining parameters in the slab and spike prior can be written first for the masking matrix $\mathbf{Q}$ as

$$q_{ij}|\mathbf{X}_{i:},\mathbf{D}_{i:},\mathbf{Z},\psi_i,\tau_{ij},\eta_{ij} \ \sim \text{Bernoulli}\left(q_{ij}\middle|\frac{\xi_{\eta_{ij}}}{1+\xi_{\eta_{ij}}}\right) \ , \tag{17}$$

where

$$\xi_{\eta_{ij}} \ = \frac{\alpha_m\nu_j}{1-\alpha_m\nu_j}\frac{\psi_i^{1/2}}{(\mathbf{Z}_{j:}\mathbf{Z}_{j:}^\top + \tau_{ij}^{-1})^{1/2}}\exp\left(\frac{(\boldsymbol{\epsilon}_{\setminus ij}\mathbf{Z}_{j:}^\top)^2}{2\psi_i(\mathbf{Z}_{j:}\mathbf{Z}_{j:}^\top + \tau_{ij}^{-1})}\right) \ ,$$

and the probability of each element of $\mathbf{C}$ of being non-zero as

$$\eta_{ij}|u_{ij},q_{ij},\alpha_p,\alpha_m \ \sim (1-u_{ij})\delta(\eta_{ij}) + u_{ij}\text{Beta}(\eta_{ij}|\alpha_p\alpha_m + q_{ij}, \alpha_p(1-\alpha_m) + 1 - q_{ij}) \ , \tag{18}$$

where $u_{ij} \sim \text{Bernoulli}(h_{ij}|r_{ij} + (1-r_{ij})\nu_j(1-\alpha_m)/(1-\nu_j\alpha_m))$, i.e. we set $u_{ij} = 1$ if $q_{ij} = 1$. Finally, for the column-wise shared sparsity rate we have

$$\nu_j|\mathbf{u}_j,\beta_p,\beta_m \ \sim \text{Beta}\left(\nu_j\middle|\beta_p\beta_m + \sum_{i=1}^d u_{ij}, \beta_p(1-\beta_m) + \sum_{i=1}^d(1-u_{ij})\right) \ . \tag{19}$$

Sampling from the DAG model only requires minor changes in notation but the conditional posteriors are essentially the same. The changes mostly amount to replacing accordingly $\mathbf{C}$ by $\mathbf{B}$ and $\mathbf{Q}$ by $\mathbf{R}$. Note that $\mathbf{Q}_L$ is the identity and $\mathbf{R}$ is strictly lower triangular a priori, thus we only need to sample their active elements.

Inference with missing values We introduce a binary masking matrix indicating whether an element of $\mathbf{X}$ is missing or not. For the factor model we have the following modified likelihood

$$p(\mathbf{X}_{\text{tr}}|\mathbf{C}, \mathbf{Z}, \mathbf{\Psi}, \mathbf{M}_{\text{miss}}) = \mathcal{N}(\mathbf{M}_{\text{miss}} \odot \mathbf{X}|\mathbf{M}_{\text{miss}} \odot (\mathbf{CZ}), \mathbf{\Psi}) \ .$$

Testing on the missing values, $\mathbf{M}^{\star}_{\text{miss}} = \mathbf{1}\mathbf{1}^{\top} - \mathbf{M}$ requires averaging the test likelihood

$$p(\mathbf{X}^{\star}|\mathbf{C}, \mathbf{Z}, \mathbf{\Psi}, \mathbf{M}^{\star}_{\text{miss}}) = \mathcal{N}(\mathbf{M}^{\star}_{\text{miss}} \odot \mathbf{X}|\mathbf{M}^{\star}_{\text{miss}} \odot (\mathbf{CZ}), \mathbf{\Psi}) \ ,$$

over $\mathbf{C}, \mathbf{Z}, \mathbf{\Psi}$ given $\mathbf{X}_{\text{tr}}$ (training). We can approximate the predictive density $p(\mathbf{X}^{\star}|\mathbf{X}_{\text{tr}}, \cdot)$ by computing the likelihood above during sampling using the conditional posteriors of $\mathbf{C}$, $\mathbf{Z}$ and $\mathbf{\Psi}$ and then summarizing using for example the median. Drawing from $\mathbf{C}$, $\mathbf{Z}$, $\mathbf{\Psi}$ can be achieved by sampling from their respective conditional distributions as described before with some minor modifications.

References

D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society: Series B (Methodology)*, 36(1):99–102, 1974.

A. Asuncion and D .J. Newman. UCI machine learning repository, 2007.

A. Azzalini and A. W. Bowman. A look at some data on the Old Faithful geyser. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 39(3):357–365, 1990.

P. Bekker and J. M. F. ten Berge. Generic global indentification in factor analysis. *Linear Algebra and its Applications*, 264(1–3):255–263, 1997.

M. Branco and D. K. Dey. A general class of multivariate skew-elliptical distributions. *Journal of Multivariate Analysis*, 79(1):99–113, 2001.

C. M. Carvalho, J. Chang, J. E. Lucas, J. R. Nevins, Q. Wang, and M. West. High-dimensional sparse factor modeling: Applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484):1438–1456, 2008.

R. S. Chhikara and L. Folks. *The Inverse Gaussian Distribution: Theory, Methodology, and Applications*. M. Dekker, New York, 1989.

S. Chib. Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association*, 90(732):1313–1321, 1995.

D. M. Chickering. Learning Bayesian networks is NP-complete. In D. Fisher and H.-J. Lenz, editors, *Learning from Data: AI and Statistics*, pages 121–130. Springer-Verlag, 1996.

P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3): 287–314, 1994.

G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

P. Daniusis, J. Janzing, J. Mooij, J. Zscheischler, B. Steudel, K. Zhang, and B. Schölkopf. Inferring deterministic causal relations. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 2010.

A. P Dawid and S. L Lauritzen. Hyper Markov laws in the statistical analysis of decomposable graphical models. *Annals of Statistics*, 21(3):1272–1317, 1993.

A. P. Dempster. Covariance selection. *Biometrics*, 28:157–175, 1972.

G. Elidan, I. Nachman, and N. Friedman. "Ideal Parent" structure learning for continuous variable Bayesian networks. *Journal of Machine Learning Research*, 8:1799–1833, 2007.

N. Friedman and D. Koller. Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50(1–2):95–125, 2003.

N. Friedman and I. Nachman. Gaussian process networks. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 211–219. 2000.

N. Friedman, I. Nachman, and D. Pe'er. Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm. In K. B. Laskey and H. Prade, editors, *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 206–215, 1999.

N. Friel and A. N. Pettitt. Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Methodology)*, 70(3):589–607, 2008.

S. Gama-Castro, V. Jiménez-Jacinto, M. Peralta-Gil, A. Santos-Zavaleta, M. I. Peñaloza-Spinola, B. Contreras-Moreira, J. Segura-Salazar, L. Muñiz-Rascado, I. Martínez-Flores, H. Salgado, C. Bonavides-Martínez, C. Abreu-Goodger, C. Rodríguez-Penagos, J. Miranda-Ríos, E. Morett, E. Merino, A. M. Huerta, L. Treviño-Quintanilla, and J. Collado-Vides. RegulonDB (version 6.0): gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and textpresso navigation. *Nucleic Acids Research*, 36(Database Issue):120–124, 2008.

E. I. George and R. E. McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

J. Geweke. Variable selection and model comparison in regression. In J. Berger, J. Bernardo, A. Dawid, and A. Smith, editors, *Bayesian Statistics 5*, pages 609–620. Oxford University Press, 1996.

Z. Ghahramani, T. L. Griffiths, and P. Sollich. Bayesian nonparametric latent feature models. In J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, editors, *Bayesian Statistics 8*, pages 201–226. Oxford University Press, 2006.

P. Giudici and P. J Green. Decomposable graphical Gaussian model determination. *Biometrika*, 86(4):785–801, 1999.

A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. Smola. A kernel statistical test of independence. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 585–592. MIT Press, 2008.

D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1:49–75, 2000.

R. Henao and O. Winther. Bayesian sparse factor models and DAGs inference and comparison. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 736–744. The MIT Press, 2009.

P. O. Hoyer, S. Shimizu, A. J. Kerminen, and M. Palviainen. Estimation of causal effects using linear non-Gaussian causal models with hidden variables. *Interantional Journal of Approximate Reasoning*, 49(2):362–378, 2008.

P .O. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Schölkopf. Nonlinear causal discovery with additive noise models. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 689–696. 2009.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley-Interscience, 2001.

H. Ishwaran and J. S. Rao. Spike and slab variable selection: Frequentist and Bayesian strategies. *Annals of Statistics*, 33(2):730–773, 2005.

I. T. Jolliffe, N. T. Trendafilov, and M. Uddin. A modified principal component technique based on the LASSO. *Journal of Computational and Graphical Statistics*, 12(3):531–547, 2003.

A. M. Kagan, YU. V Linnik, and C. Radhakrishna Rao. *Characterization Problems in Mathematical Statistics*. Probability and Mathematical Statistics. Wiley, New York, 1973.

K. C. Kao, Y-L. Yang, R. Boscolo, C. Sabatti, V. Roychowdhury, and J. C. Liao. Transcriptome-based determination of multiple transcription regulator activities in Escherichia Coli by using network component analysis. *PNAS*, 101(2):641–646, 2004.

D. Knowles and Z. Ghahramani. Infinite sparse factor analysis and infinite independent components analysis. In M. E. Davies, C. C. James, S. A. Abdallah, and M. D. Plumbley, editors, *7th International Conference on Independent Component Analysis and Signal Separation*, volume 4666 of *Lecture Notes in Computer Science*, pages 381–388. Springer-Verlag, Berlin, 2007.

F. B. Lempers. *Posterior Probabilities of Alternative Linear Models*. Rotterdam University Press, 1971.

H. F. Lopes and M. West. Bayesian model assessment in factor analysis. *Statistica Sinica*, 14(1):41–67, 2004.

J. Lucas, C. Carvalho, Q. Wang, A. Bild, J. R. Nevins, and M. West. *Bayesian Inference for Gene Expression and Proteomics*, chapter Sparse Statistical Modeling in Gene Expression Genomics, pages 155–176. Cambridge University Press, 2006.

J. K. Martin and R. P. McDonald. Bayesian estimation in unrestricted factor analysis: A treatment for heywood cases. *Psychometrika*, 40(4):505–517, 1975.

T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

J. Mooij and D. Janzing. Distinguishing between cause and effect. In *JMLR Workshop and Conference Proceedings*, volume 6, pages 147–156, 2010.

I. Murray. *Advances in Markov Chain Monte Carlo Methods*. PhD thesis, Gatsby computational neuroscience unit, University College London, 2007.

R. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

T. Park and G. Casella. The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

P. Rai and H. Daume III. The infinite hierarchical factor regression model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1321–1328. The MIT Press, 2009.

B. Rajaratman, H. Massam, and C. Carvalho. Flexible covariance estimation in graphical gaussian models. *Annals of Statistics*, 36(6):2818–2849, 2008.

J. M. Robins, R. Scheines, P. Spirtes, and L. Wasserman. Uniform consistency in causal inference. *Biometrika*, 90(3):491–515, 2003.

K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

M. W. Schmidt, A. Niculescu-Mizil, and K. P. Murphy. Learning graphical model structure using L1-regularization paths. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, pages 1278–1283, 2007.

S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7:2003–2030, 2006.

R. Silva. *Causality in the Sciences*, chapter Measuring Latent Causal Structure. Oxford University Press, 2010.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. The MIT Press, second edition, 2001.

M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pages 548–549, 2005.

R. Thibaux and M. I. Jordan. Hierarchical beta processes and the indian buffet process. In M. Meila and X. Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, pages 564–571, 2007.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodology)*, 58(1):267–288, 1996.

R. Tillman, A. Gretton, and P. Spirtes. Nonlinear directed acyclic structure learning with weakly additive noise models. In *Advances in Neural Information Processing Systems 22*, pages 1847–1855. Y. Bengio and D. Schuurmans and J. Lafferty and C. K. I. Williams and A. Culotta, 2009.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

M. West. On scale mixtures of normal distributions. *Biometrika*, 74(3):646–648, 1987.

M. West. Bayesian factor regression models in the "large p, small n" paradigm. In J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, editors, *Bayesian Statistics 7*, pages 723–732. Oxford University Press, 2003.

S. Yu, V. Tresp, and K. Yu. Robust multi-task learning with t-processes. In *Proceedings of the 24th International Conference on Machine Learning*, volume 227, pages 1103–1110, 2007.

K. Zhang and A. Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 647–655. AUAI Press, 2009.

K. Zhang and A. Hyvärinen. Distinguishing causes from effect using nonlinear acyclic causal models. In *JMLR Workshop and Conference Proceedings*, volume 6, pages 157–164, 2010.

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):262–286, 2006.

Appendix D

Predictive Active Set Selection Methods for Gaussian Processes

---

Submitted to
Neurocomputing

Available from DTU Informatics at
`http://imm.dtu.dk/~rh/asetgp.pdf`

Complementary website at
`http://cogsys.imm.dtu.dk/passgp`

Predictive Active Set Selection Methods for Gaussian Processes

Ricardo Henao, Ole Winther

DTU Informatics, Technical University of Denmark, Denmark
Bioinformatics Centre, University of Copenhagen, Denmark

Abstract

We propose an active set selection framework for Gaussian process classification for cases when the dataset is large enough to render its inference prohibitive. Our scheme consists on a two step alternating procedure of active set update rules and hyperparameter optimization based upon marginal likelihood maximization. The active set update rules rely on the ability of the predictive distributions of a Gaussian process classifier to estimate the relative contribution of a datapoint when being either included or removed from the model. This means that we can use it to include points with potentially high impact to the classifier decision process while removing those that are less relevant. We introduce two active set rules based on different criteria, the first one prefers a model with interpretable active set parameters whereas the second puts computational complexity first, thus a model with active set parameters that directly control its complexity. We also provide both theoretical and empirical support for our active set selection strategy being a good approximation of a full Gaussian process classifier. Our extensive experiments show that our approach can compete with state-of-the-art classification techniques with reasonable time complexity. Source code publicly available at `http://cogsys.imm.dtu.dk/passgp`.

Keywords: Gaussian process classification, active set selection, predictive distribution, expectation propagation

1. Introduction

Classification with Gaussian process (GP) priors has many attractive features, for instance it is non-parametric, exceptionally flexible through covariance function designs, provides fully probabilistic outputs and Bayesian model comparison as principled framework for automatic hyperparameter elicitation and variable selection. However, such a set of features comes in with a great disadvantage since the computational cost of performing inference scales cubically with the size N of the training set. In addition, the memory requirements scale quadratically also with N. This means that applicability of Gaussian process classification (GPC) is sadly limited to problems with dataset sizes in the lower ten thousands. The poor scaling of specially non-linear classification methods has inspired a considerable amount of research effort focused on sparse approximations [1, 2, 3, 4, 5, 6]. See particularly [1, 2] for a detailed overview of sparse approximations in GPC. These methods attempt in general to decrease the computational cost of inference in one degree w.r.t. N, i.e. $\mathcal{O}(NM^2)$, where $M < N$ and M is the size of a working set consisting on a subset of the training data or a set of auxiliary unobserved variables. Both ways of defining the working set basically target the same objective of getting as close as possible to the classifier that uses the information of the entire training set, however they approach it from different angles. Using a subset from the entire data pool amounts to keep those data points that better contribute to the classification task and discard the remaining ones through some suitable data selection/ranking procedure [7, 8, 9, 6]. Alternatively, building an auxiliary set tries to directly reduce the difference in distribution between the the classifier using N points and the one using only M, by estimating the location of an auxiliary set in the input space, usually called pseudo-input set [1, 4]. The latter approach is evidently more principled, however the number of parameters to be learnt grows with the number and size of the auxiliary set, making it unfeasible for datasets in the upper ten thousands and sensible to overfitting due to the number of free parameters in the model.

Having in mind that our main goal is to obtain the best classification performance with the least computational cost possible, we do not attempt to estimate auxiliary sets but rather to select a subset of the training data. The framework presented here, Predictive Active Set Selection (PASS-GP) uses the predictive distribution of a GP classifier in order to quantify the relative importance of each datapoint and then use it to iteratively update an active set. We call it active set because it is ultimately the one used to estimate the predictive distribution that produces the classification rule and active set updating scheme. In a nutshell, our framework consists on alternating between active set updates and hyperparameter optimization

*Corresponding author
Email addresses: `rhenao@binf.ku.dk` (Ricardo Henao), `owi@imm.dtu.ku.dk` (Ole Winther)

based upon the marginal likelihood of the active set. We provide two active set update schemes that target different practical scenarios. The first simply called PASS-GP builds the active set by including/removing points with small/large predictive probability until predictive convergence is reached, i.e. no more or too few data points are included in the active set. This means that the size of the active set is not known in advance so as the expected computational complexity. The second scheme is aware that in some applications is very important to keep the computational complexity and/or memory requirements on a budget, thus being able to specify the size of the active set beforehand is essential. In fixed PASS-GP (fPASS-GP) we keep the size of the active set constant by including and removing the same amount of data points in each update to achieve the desired behavior.

The remainder of the paper presents in Section 2 a concise description of expectation propagation based inference for GPC. Section 3 continues with our proposed framework for active set selection, followed by some theoretical insights based upon a 'representer theorem' for the predictive mean of a GP classifier in Section 4. Marginal likelihood approximations to the full GP classifier are introduced in section 5. Finally, experimental results and discussion appear in Sections 6 and 7, respectively.

2. Gaussian Processes for Classification

Given a set of input random variables $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top$, a Gaussian process is defined as a joint Gaussian distribution over functions in the input points $\mathbf{f} = [f_1, \ldots, f_N]^\top$ with mean vector $\mathbf{m}$ (taken to be zero in the following) and covariance matrix $\mathbf{K}$ with elements $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and hyperparameters $\boldsymbol{\theta}$. For classification, assuming independently observed binary ± 1 labels $\mathbf{y} = [y_1, \ldots, y_N]^\top$ and a probit (cumulative Gaussian) likelihood function $t(y_n|f_n) = \Phi(f_n y_n)$, we end up with an intractable posterior

$$p(\mathbf{f}|\mathbf{X}, \mathbf{y}) = Z^{-1} p(\mathbf{f}|\mathbf{X}) \prod_{n=1}^{N} t(y_n|f_n) \ ,$$

where the normalizing constant $Z = p(\mathbf{y}|\mathbf{X})$ is the marginal likelihood. If we want to perform inference we must resort to approximations. Here we use Expectation Propagation (EP) because it is the currently the most accurate deterministic approximation, see e.g. [2, 10]. In EP, the likelihood function is locally approximated by an un-normalized Gaussian distribution to obtain

$$\begin{aligned} q(\mathbf{f}|\mathbf{X}, \mathbf{y}) &= Z_{\text{EP}}^{-1} p(\mathbf{f}|\mathbf{X}) \prod_{n=1}^{N} z_n^{-1} \widetilde{t}(y_n|f_n) \\ &= Z_{\text{EP}}^{-1} p(\mathbf{f}|\mathbf{X}) \mathcal{N}(\mathbf{f}|\widetilde{\mathbf{m}}, \widetilde{\mathbf{C}}) \ , \\ &= \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{c}) \ , \end{aligned} \tag{1}$$

where $q(\mathbf{f}|\mathbf{X}, \mathbf{y}) \approx p(\mathbf{f}|\mathbf{X}, \mathbf{y})$, the z_n are the normalization coefficients, $\widetilde{t}(y_n|f_n)$ and $\mathcal{N}(\mathbf{f}|\widetilde{\mathbf{m}}, \widetilde{\mathbf{C}})$ conform the site Gaussian approximations to $t(y_n|f_n)$. In order to obtain $q(\mathbf{f}|\mathbf{X}, \mathbf{y})$, one starts from $q(\mathbf{f}|\mathbf{X}, \mathbf{y}) = p(\mathbf{f}|\mathbf{X})$ and update the individual $\widetilde{t}_n$ site approximations sequentially. For this purpose, we delete the site approximation $\widetilde{t}_n$ from the current posterior leading to the so called cavity distribution

$$q_{\setminus n}(\mathbf{f}|\mathbf{X}, \mathbf{y}_{\setminus n}) = p(\mathbf{f}|\mathbf{X}) \prod_{i \neq n} z_i^{-1} \widetilde{t}(y_i|f_i) \ ,$$

from which we can obtain a cavity predictive distribution

$$\begin{aligned} q_{\setminus n}(y_n|\mathbf{X}, \mathbf{y}_{\setminus n}) &= \int t(y_n|f_n) q_{\setminus n}(\mathbf{f}|\mathbf{X}, \mathbf{y}_{\setminus n}) d\mathbf{f} \ , \\ &= \Phi\left(\frac{y_n m_{\setminus n}}{\sqrt{1 + v_{\setminus n}}} \right) \ , \end{aligned} \tag{2}$$

where $m_{\setminus n} = v_{\setminus n}(C_{nn}^{-1} m_n - \widetilde{C}_{nn}^{-1} \widetilde{m}_n)$ and $v_{\setminus n} = (C_{nn}^{-1} - \widetilde{C}_{nn}^{-1})^{-1}$. We then combine the cavity distribution with the exact likelihood $t(y_n|f_n)$, to obtain the so called tilted distribution $q_n(\mathbf{f}|\mathbf{X}, \mathbf{y}) = z_n^{-1} t(y_n|f_n) q_{\setminus n}(\mathbf{f}|\mathbf{X}, \mathbf{y}_{\setminus n})$. Since we need to choose the parameters of the site approximations we must minimize some divergence measure. It is well known that when $q(\mathbf{f}|\mathbf{X}, \mathbf{y})$ is Gaussian, minimizing $\text{KL}(p(\mathbf{f})||q(\mathbf{f}))$ is equivalent to moment matching between those two distributions including zero-th moments for the normalizing constants. The EP algorithm iterates by updating each site approximation in turn and makes several passes over the training data until convergence is reached.

With the Gaussian approximation to the posterior distribution in equation (1), it is possible to calculate the predictive distribution of a new datapoint $\mathbf{x}^\star$ as

$$\begin{aligned} q(y^*|\mathbf{X}, \mathbf{y}, \mathbf{x}^\star) &= \int t(y^\star|f^\star) q(f^\star|\mathbf{X}, \mathbf{y}, \mathbf{x}^\star) df^\star \ , \\ &= \Phi\left(\frac{y^\star m^\star}{\sqrt{1 + v^\star}} \right) \ , \end{aligned} \tag{3}$$

where $q(f^\star|\mathbf{X}, \mathbf{y}, \mathbf{x}^\star)$ is the approximate predictive Gaussian distribution (the marginal of $q(\mathbf{f}, f^\star|\mathbf{X}, \mathbf{y}, \mathbf{x}^\star)$ w.r.t. $\mathbf{f}$) with mean $m^\star = \mathbf{k}^{\star\top}(\mathbf{K} + \widetilde{\mathbf{C}})^{-1} \widetilde{\mathbf{m}}$ and variance $v^\star = k^{\star\star} + \mathbf{k}^{\star\top}(\mathbf{K} + \widetilde{\mathbf{C}})^{-1} \mathbf{k}^\star$. In addition, the approximation to the marginal likelihood $p(\mathbf{y}|\mathbf{X})$ results in the normalization constant from equation (1), i.e. $q(\mathbf{y}|\mathbf{X}) = Z_{\text{EP}}$. The logarithm of $Z_{\text{EP}}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y})$ and its derivatives can be used jointly with conjugate gradient updates to perform model selection under the evidence maximization framework. For a detailed presentation of GP including its implementation details, consult [2, 10].

3. Predictive Active Set Selection

The EP algorithm is performed by iterative updates of each site approximation using the whole dataset $\{\mathbf{X}, \mathbf{y}\}$. In the active set scenario on the other hand, we only want to approximate the posterior distribution in equation (1) using a small subset, the active set $\{\mathbf{X}_A, \mathbf{y}_A\}$. Since exploring all possible active sets is obviously intractable even

for a fixed active set size M, the problem is how to select an active set that delivers a performance as good as possible within the available computing resources. The Informative Vector machine (IVM) for instance, computes in each iteration the differential entropy score for all data points not already part of the active set $\{\mathbf{X}_I, \mathbf{y}_I\}$ and perform updates by including the single point leading to a maximum score. Despite this greedy heuristic, IVM has proved to behave quite well in practice, giving the so far best reported GP performance on the USPS and MNIST tasks [7, 8]. We propose an iterative approach in the same spirit with two main conceptual changes:

- **Active set inclusion/deletion** based directly upon the data point weight in prediction. The 'representer theorem' for the mean prediction, discussed in Section 4, leads directly to the weight being expressed in terms of (a derivative of) the cavity predictive probability. This means that we can actually use the predictive distribution for a point in the inactive set to predict the weight it would have if it would be included in the active set. For classification we use the (cavity) predictive probability to decide upon deletion and inclusion because it is monotonically related the weight and a readily interpretable quantity.
- **Hyperparameter optimization** must be an integral part of algorithm, because the weights of the examples (and thus the active set) is conditioned on the hyperparameter values and vice versa. We therefore alternate between active set updates and hyperparameter optimization using several passes over the data set to allow for convergence.

Next we discuss the details of our (f)PASS-GP framework followed by a detailed comparison with the IVM. First we need to define rules for including and deleting points of the active set. As already mentioned, we use the predictive distribution in equation (3) for inclusions since data points with small predictive probability are more likely to contribute to improve the classifier performance and the quality of the active set. For deletions, we use the cavity predictive distribution in equation (2) because when examined carefully can be seen as a leave-one-out estimator [11]. This means that points with cavity probability close to one do not contribute to the decision rule thus can be discarded from the active set. With the two ranking measures set, i.e. equations (2) and (3), we have essentially two possibilities. The first is to set probability thresholds on the distributions and let the model to decide the size of the active set or we can rather specify directly the amount of inclusions/deletions. In PASS-GP, we include points in the active set with probability less than p_{inc} and remove them with probability greater than p_{del}. The appealing aspect of these thresholds is that they can be interpreted, for instance if we set $p_{\text{inc}} = 0.5$ we will include all misclassified observations in the current active set whereas if $p_{\text{inc}} = 0.6$ we will also include points near the decision boundary. We require two thresholds because we only want to remove points that as for the classifier are very easy to classify, so unlike p_{inc}, p_{del} must be close enough to one. In fPASS-GP, we want to keep the computational complexity of the classifier under control thereby we want the size of the active set to be fixed. For this purpose we only have to be sure that each active set update includes and removes the same amount of points. In practice we define p_{exc} as the exchange proportion w.r.t. M, meaning that each update replaces the fixed proportion of most hard to classify points in the inactive set with those more surely classified in the current active set. This update rule assumes that the active set is large enough to contain points in the active set with cavity probability close to one.

From a practical point of view, ranking every point the in inactive set at each iteration for inclusion could become prohibitive for large datasets. However we still want to be able to cover the whole dataset rather than selecting a random subset for ranking. We then split the data into N_{sub} non-overlapping subsets and process each one of them in each iteration, such that each batch has something between 100 and 1000 data points.

Hyperparameter selection is a very important feature and needs to be done jointly with the active set update procedure. Algorithm 1 starts from a fixed randomly selected active set of size N_{init} (that is M in fPASS-GP), large enough to provide a good initial hyperparameter set values. Next we alternate between active set and hyperparameter optimization updates. Having in mind that we only expect small changes of the hyperparameters from one iteration to another, we reuse current values of θ as initial values for the next iteration to speed-up the learning process. The addition and deletion rules in Algorithm 1 have parameters $\{p_{\text{inc}}, p_{\text{del}}\}$ and p_{exc} for PASS-GP and fPASS-GP, respectively.

3.1. Differences between (f)PASS-GP and IVM

Since IVM is the closest relative of our active set selection method, we briefly discuss the main differences between the two: (i) The active set and thus the computational complexity is usually fixed beforehand in IVM. PASS-GP works with inclusion and deletion thresholds instead. (ii) IVM does not allow for deletions from the active set which is a clear disadvantage as points often become irrelevant at a later stage, when more points have been included. In (f)PASS-GP we can make an (almost) unbiased common ranking of all training points active as well as inactive, using a quantity that is meaningful and directly related to the weight of the training point in predictions. Using both inclusions/deletions and several passes over the training set makes (f)PASS-GP quite insensitive to the initial choice of active set. (iii) When the dataset is considerably large, IVM randomly selects a subset of points to be ranked from the inactive set, meaning that is likely that some points of the dataset are never considered for

Algorithm 1:
Predictive active set selection algorithms

Input : $\{\mathbf{X}, \mathbf{y}\}$, $\boldsymbol{\theta}$ and $\{N_{\text{init}}, N_{\text{sub}}, N_{\text{pass}}\}$
Input : p_{inc} and p_{del} (PASS-GP)
Input : p_{exc} (fPASS-GP)
Output: $q(\mathbf{f}_A|\mathbf{X}_A, \mathbf{y}_A)$, $\boldsymbol{\theta}_{\text{new}}$ and A
begin
  $A \leftarrow \{1, \ldots, N_{\text{init}}\}$
  $\{\mathbf{X}, \mathbf{y}\}_{\text{sub}}^{(1)}, \ldots \{\mathbf{X}, \mathbf{y}\}_{\text{sub}}^{(N_{\text{sub}})} \leftarrow \{\mathbf{X}, \mathbf{y}\}$
  for $i = 1$ **to** N_{pass} **do**
    for $j = 1$ **to** N_{sub} **do**
      $\boldsymbol{\theta}_{\text{new}} = \text{argmax}_{\boldsymbol{\theta}} \log Z_{\text{EP}}(\boldsymbol{\theta}, \mathbf{X}_A, \mathbf{y}_A)$
      Get $q(\mathbf{f}_A|\mathbf{X}_A, \mathbf{y}_A)$ and $q(y^*|\mathbf{X}_A, \mathbf{y}_A, \mathbf{x}^\star)$
      forall the $\{\mathbf{x}_n, y_n\} \in \{\mathbf{X}_A, \mathbf{y}_A\}$ **do**
        if RemoveRule$(q_{\backslash n}(y_n|\mathbf{X}_A, \mathbf{y}_{A\backslash n}))$
        then $A \leftarrow A \backslash \{n\}$
      end
      forall the $\{\mathbf{x}_n, y_n\} \in \{\mathbf{X}, \mathbf{y}\}_{\text{sub}}^{(j)}$ **do**
        if AdditionRule$(q(y^*|\mathbf{X}_A, \mathbf{y}_A, \mathbf{x}^\star, \boldsymbol{\upsilon}))$
        then $A \leftarrow A \cup \{n\}$
      end
    end
  end
end

inclusion in the active set. (iv) The hyperparameter optimization is a part of the algorithm in (f)PASS-GP working on subsets of data between updates and iterating over the data set in principle until convergence. IVM makes a single inclusion per step and in principle stops when the limit for the active set is reached. (iv) In terms of complexity time per iteration IVM it is faster than (f)PASS-GP, $\mathcal{O}(NM)$ against $\mathcal{O}(M^2(2+N/N_{\text{sub}}))$ where M is the size of A, however storage requirements are considerable lower, $\mathcal{O}(M^2)$ compared to $\mathcal{O}(NM)$.

4. Representer for Mean Prediction

The 'representer theorem' for the posterior mean of $\mathbf{f}$ [11], connects the predictive probability and the weight of a data point. Using that $p(\mathbf{f}|\mathbf{X}) = -\mathbf{K}\frac{\partial}{\partial \mathbf{f}} p(\mathbf{f}|\mathbf{X})$, we get the exact relation for the posterior mean $\langle \mathbf{f} \rangle = \mathbf{K}\boldsymbol{\alpha}$ with the weight of element n being

$$\begin{aligned}\alpha_n &= \frac{1}{p(\mathbf{y}|\mathbf{X})} \int p(\mathbf{f}|\mathbf{X}) \frac{\partial}{\partial f_n} p(\mathbf{y}|\mathbf{f}) d\mathbf{f} \\ &= \frac{\langle\, p'(y_n|f_n) \,\rangle_{\backslash n}}{\langle\, p(y_n|f_n) \,\rangle_{\backslash n}} , \\ &= \frac{\partial}{\partial h} \log \langle\, p(y_n|f_n + h) \,\rangle_{\backslash n} \Big|_{h=0} ,\end{aligned}$$

where $\langle \cdot \rangle_{\backslash n}$ denotes an average over a posterior without the n-th data point and $p'(y_n|f_n) = \partial p(y_n|f_n)/\partial f_n$. The final expression implies that the weight is nothing but the log derivative of the cavity predictive probability $\langle\, p(y_n|f_n) \,\rangle_{\backslash n} = p(y_n|\mathbf{X}, \mathbf{y}_{\backslash n})$. For regression, $p(y_n|f_n) = \mathcal{N}(y_n|f_n, \sigma^2)$ and $\alpha_n = (y_n - \langle\, f_n \,\rangle_{\backslash n})(\sigma^2 + v_n)^{-1}$ with $v_n = \langle\, f_n^2 \,\rangle_{\backslash n} - \langle\, f_n \,\rangle_{\backslash n}^2$. The element α_n will therefore be small when the cavity mean has a small deviation from the target relative to the variance. For a new data point pair $\{\mathbf{x}^\star, y^\star\}$, we can calculate the weight of this point *exactly*, replacing the cavity average with the full average in the expression above. We can therefore predict without any EP rerunning, how much weight this new point will have. For classification we can calculate the weight using the current EP approximation. When $z_n = y_n \langle\, f_n \,\rangle_{\backslash n} / \sqrt{1 + v_n}$ is above ≈ 4, the cavity probability equation (2) approaches one and $\alpha_n \approx y_n \exp(-z_n^2/2)/\sqrt{2\pi(1+v_n)}$. This fast decay indicates that GPC in many cases will be effectively sparse even though $\boldsymbol{\alpha}$ strictly does not contain zeros.

In the inclusion/deletion steps we rank data points according to their weights. For classification we can indeed use the predictive probability directly, since it is a monotonic function of the weight. Including a new data point will of course affect the value of all other weights as well leading to a rearrangement of their rank. Including multiple data points will also invalidate the predicted value of the weights (e.g. think of the extreme of two new data points being identical). We therefore have to recalculate the weights by retraining with EP for classification or simply updating the posterior for regression before going to the next step. If we have already an active set covering the decision regions well enough, this rearrangement step will amount to minor adjustments and the approximation will work well.

In this work we have only used the representer theorem for active set selection. It is also possible, but not tested here, to use all training points for prediction while only calculating the posterior on the active set. The inactive set weights are then simply set to the predicted values from the active set posterior. To get the full predictive probability one also has to calculate the contribution to the predictive variances which can be obtained by a similar theorem but for the predictive variance, see [11].

5. Marginal Likelihood Approximations

In this section we decompose the marginal likelihood in their active and inactive set contributions. We will argue that the contribution from the active set will dominate, justifying why we can limit ourselves to optimizing the hyperparameters over this set. In the following section we will investigate this assumption empirically. The marginal likelihood can be decomposed via the chain rule as

$$p(\mathbf{y}|\mathbf{X}) = \; p(\mathbf{y}_I|\mathbf{y}_A, \mathbf{X}_A, \mathbf{X}_I) p(\mathbf{y}_A|\mathbf{X}_A) \; , \qquad (4)$$

where we have used the marginalization property of GPs,

$$p(\mathbf{y}_A|\mathbf{X}) = \int p(\mathbf{y}_A|\mathbf{f}_A) p(\mathbf{f}_A|\mathbf{X}_A) d\mathbf{f}_A = p(\mathbf{y}_A|\mathbf{X}_A) \; ,$$

that we approximate as $q(\mathbf{y}_A|\mathbf{X}_A) = Z_{\mathrm{EP},A}$ and we identify it as the marginal likelihood for the active set A. The conditional marginal likelihood term can be written as

$$p(\mathbf{y}_I|\mathbf{y}_A, \mathbf{X}_A, \mathbf{X}_I) = \int p(\mathbf{y}_I|\mathbf{f}_I)p(\mathbf{f}_I|\mathbf{X}_I, \mathbf{f}_A)p(\mathbf{f}_A|\mathbf{X}_A, \mathbf{y}_A)\, d\mathbf{f}_A d\mathbf{f}_I \ , \quad (5)$$

where we have used $p(\mathbf{f}|\mathbf{X}) = p(\mathbf{f}_I|\mathbf{X}_I, \mathbf{f}_A)p(\mathbf{f}_A|\mathbf{X}_A, \mathbf{y}_A)$. We can make an EP approximation here just like in equation (1) by replacing the posterior $p(\mathbf{f}_A|\mathbf{X}_A, \mathbf{y}_A)$ by the multivariate Gaussian $q(\mathbf{f}_A|\mathbf{X}_A, \mathbf{y}_A) = \mathcal{N}(\mathbf{f}_A|\mathbf{m}_A, \mathbf{C}_{AA})$ where active set specific means and variances are found by EP. Marginalizing over $\mathbf{f}_A$ in equation (5) makes it now tractable

$$q(\mathbf{y}_I|\mathbf{y}_A, \mathbf{X}_A, \mathbf{X}_I) \approx \int p(\mathbf{y}_I|\mathbf{f}_I)\mathcal{N}(\mathbf{f}_I|\mathbf{m}_{I|A}, \mathbf{C}_{II|A})d\mathbf{f}_I \ ,$$

with parameters

$$\mathbf{m}_{I|A} = \mathbf{K}_{IA}(\mathbf{K}_{AA} + \widetilde{\mathbf{C}}_{AA})^{-1}\widetilde{\mathbf{m}}_A \ ,$$
$$\mathbf{C}_{II|A} = \mathbf{K}_{II} - \mathbf{K}_{IA}(\mathbf{K}_{AA} + \widetilde{\mathbf{C}}_{AA})^{-1}\mathbf{K}_{AI} \ ,$$

where the tilted moments are as defined in Section 2. When the inactive set consists of a single example, we obtain the EP predictive distribution in equation (3), otherwise we have to solve for a new marginal likelihood. Denoting the marginal likelihood for a set $\{\mathbf{X}, \mathbf{y}\}$ with a non-zero mean GP prior by

$$Z(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}, \mathbf{m}) = \int p(\mathbf{y}|\mathbf{f})\mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{K})\, d\mathbf{f} \ ,$$

and its EP approximation by $Z_{\mathrm{EP}}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}, \mathbf{m})$, we can write the approximation to the marginal likelihood in equation (4) as

$$Z_{\mathrm{ACC}} \equiv Z_{\mathrm{EP}}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}_I, \mathbf{m}_{I|A})Z_{\mathrm{EP}}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}_A, \mathbf{0}) \ .$$

Using this approximate decomposition reduces the complexity of EP from $\mathcal{O}(N^3 N_{\mathrm{pass}})$ to $\mathcal{O}((|I|^3 + M^3)N_{\mathrm{pass}})$, where $|I|$ is the size of the inactive set. Unfortunately this is still too costly for large N.

We conclude this section with a few more qualitative comments that we will follow up upon in the empirical work. Since the inactive set I contains the well-classified patterns with predictive probability close to one, the marginal likelihood per example will be much smaller for the $I|A$-term than for the A-term. The values of the hyperparameters (length scales, etc.) will to a very large degree be determined by the active set examples lying close to the decision boundary. As a result, the product of marginals will be a lower bound to the marginal likelihood: $p(y_I|y_A, \mathbf{X}) > \prod_{i\in I} p(y_i|\mathbf{y}_A, \mathbf{X}_A, \mathbf{x}_i)$ because the easy well separated patterns in I will enforce each other. Using this lower bound we can thus compute a cheap approximation to $p(\mathbf{y}|\mathbf{X})$, denoted by Z_{APP}, which we illustrate in the next section (see Figure 3).

6. Experiments

The results presented in this section consists on several classification tasks performed with PASS-GP on USPS and MNIST, two databases of handwritten digits well known for being hard, very unbalanced, high dimensional and with considerably large training sets. We also present results for our approximation to the marginal likelihood of the full GP presented in the previous section and an empirical comparison of our fixed computational cost approach to the Reduced complexity SVM (RSVM) [3]. All experiments were performed on a regular 2.0GHz desktop machines with 2GB RAM.

6.1. USPS

The USPS digits database contains 9289 grayscale images of size 16×16 pixels, scaled and translated to fall within the range from -1 to 1. Here we adopt the traditional data splitting, i.e. 7291 observations for training and the remaining 2007 for testing. For each binary one-against-rest classifier we use the same model setup consisting on a squared exponential covariance matrix plus additive jitter

$$k(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\theta_2}\right) + \theta_3\delta_{ij} \quad (6)$$

where $\delta_{ij} = 1$ if $i = j$ and zero otherwise. We have three hyperparameters in $\boldsymbol{\theta}$, namely signal variance, characteristic length scale and jitter coefficient. The settings used for PASS-GP were $N_{\mathrm{init}} = 300$, $N_{\mathrm{sub}} = 10$, $N_{\mathrm{pass}} = 2$, $p_{inc} = 0.6$ and $p_{del} = 0.99$. For fPASS-GP we used $N_{\mathrm{init}} = 300$, $N_{\mathrm{sub}} = 10$, $N_{\mathrm{pass}} = 4$, $p_{exc} = 0.02$. We allow fPASS-GP to perform more passes through the data because fPASS-GP is slower due to p_{exc} being small. For RSVM, we set $M = 500$, $\boldsymbol{\theta} = [1\ 1/16\ 0]$, $C = 10$ and $\kappa = 10$. Specifically, $\boldsymbol{\theta}$ and the regularization parameter C were obtained by grid search cross-validation, and κ to the value suggested by the authors [3]. The methods considered may depend upon random initialization so we repeated each task 10 times.

Figure 1(a) shows mean test errors for every one-against-rest task using PASS-GP, fPASS-GP, RSVM and the full GPC with hyperparameter optimization. Besides, Figure 1(b) shows the active set sizes for each digit using PASS-GP. From the figure it can be seen that PASS-GP performs slightly better than the other three considered alternatives. Furthermore, compared to fPASS-GP ($M = 300$) and RSVM ($M = 500$), PASS-GP seems to require smaller active sets to achieve similar classification performance. It is important to mention that we also tried larger values of M for the fixed active set algorithms without any significant improvement in performance.

Figures 2(a) and 2(c) show classification errors for digits 3 and 4 against the others, respectively, as a function of the active set size. For fPASS-GP and RSVM we used $M = \{100, 200, \ldots, 600\}$ and for PASS-GP we used

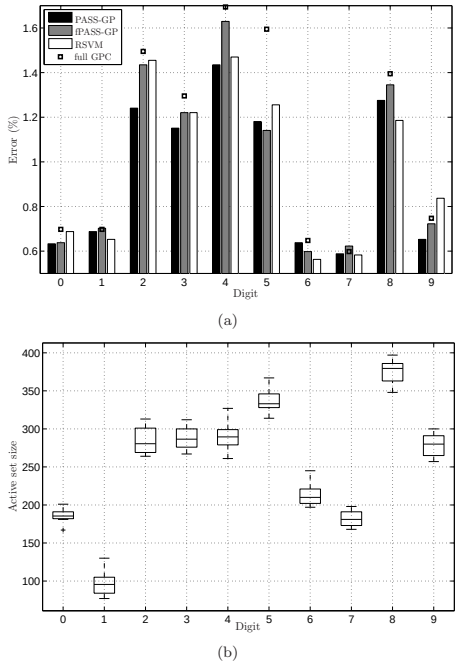


Figure 1: Error rates and active set sizes for USPS data. (a) mean classification errors for each digit task using PASS-GP, fPASS-GP, RSVM and the full GPC with hyperparameter optimization. (b) Active set sizes for PASS-GP. Note that fPASS-GP uses $M = 300$ and RSVM $M = 500$ for the results in (a).

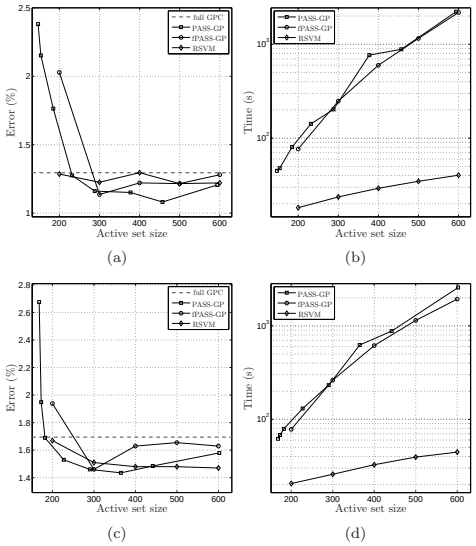


Figure 2: Results for individual digits of USPS data. (a) and (c) show classification errors as a function of the active set size for digits 3 and 4 vs the rest, respectively. (b) and (c) show corresponding running times for each case of panels (a) and (c). The horizontal line in (a) and (c) is the performance of a full GPC with hyperparameter selection. For both digits, the full GPC took 9.5e5 seconds approximately [6]. Each represents the average over ten independent repetitions.

$p_{inc} = \{0.2.0.3, \ldots, 0.9\}$. We also included the classification error obtained by a full GPC with hyperparameter optimization depicted as an horizontal line. See [6] for a more detailed comparison between PASS-GP and full GPCs. Several features from the Figures worth to be highlighted. (i) Both PASS-GP and fPASS-GP approach the full GP for large values of M, as expected. (ii) Similar to Figure 1(a), PASS-GP seems to consistently outperform fPASS-GP for similar sizes of M. (iii) For small values of M, RSVM is better than our active set methods, however further increasing M does not considerably improves its performance. When M is small enough, it is very likely that our approach is not able to obtain plausible estimates of the hyperparameters of the covariance function, thus its poor performance compared to RSVM that uses fixed values.

Figures 2(b) and 2(d) show computation times for each case of Figures 2(a) and 2(c). PASS-GP and fPASS-GP are approximately three and two orders of magnitude faster than a full GPC with and without hyperparameter optimization, respectively, see [6]. For similar active set sizes, PASS-GP and fPASS-GP have coinciding computational costs as one may expect. RSVM scales better than our active set selection methods, when looking at the slope in the running times plots. The difference in computational costs as seen in Figures 2(b) and 2(d) should not be considered as significant since we are not counting the time used to obtain the parameters used for the RSVM that unfortunately need to be selected by expensive grid search with cross-validation.

The results obtained on USPS show that (f)PASS-GP is performing slightly better than the full GPC. This could be due to numerical instability produced by the size of the problem, by the iterative nature of the EP algorithm and/or not enough iterations for the model selection procedure. However, it could also mean that optimizing on the active achieves a better "local" fit around the decision boundary region. A priori one cannot expect that one set of hyperparameters are able to describe all regions in input space and that might be what we see here. The same kind of local improvement observed here was also reported by [12] and [4] for GPC auxiliary set methods.

Combining the ten binary tasks into a one-against-rest multi-class classifier, PASS-GP obtained $4.51 \pm 0.17\%$

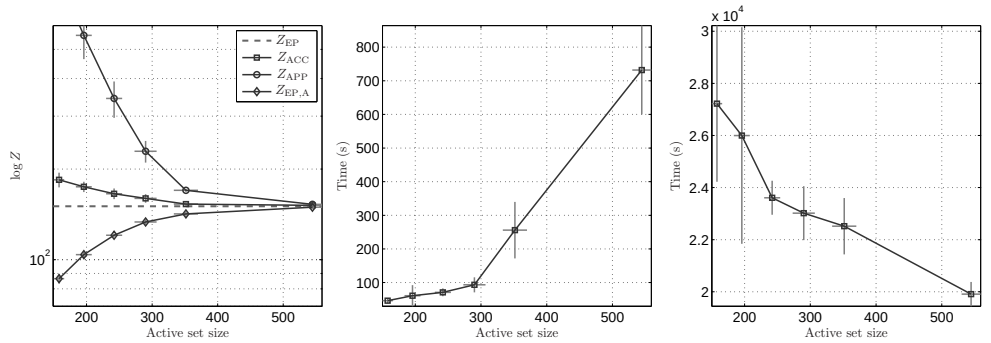


Figure 3: Marginal log-likelihood approximations as a function of the active set size for 3s vs the rest. The plots show means and standard deviations (error bars) for ten repetitions. Each marker corresponds to a different inclusion threshold $p_{\text{inc}} = \{0.5, 0.6, \ldots, 0.9, 0.99\}$. In the left panel, Z_{EP} is the full GPC ($p_{\text{inc}} = 1$) and the remaining three Z_{ACC}, Z_{APP} and $Z_{\text{EP,A}}$, the proposed approximations. The middle and right panels shows the computation times required to obtain $\{Z_{\text{APP}}, Z_{\text{EP,A}}\}$ and Z_{ACC}, respectively.

which is significantly better than[1], $4.65 \pm 0.10\%$ by fPASS-GP, $4.65 \pm 0.10\%$ by RSVM [3], 5.13% by GPC with hyperparameter optimization, 4.78% by GPC with fixed $\boldsymbol{\theta}$, 5.15% by online GP [13], 4.98% by IVM [8] and comparable with state-of-the-art techniques such as SVM, see [14]. As reference, it has been shown that the human error rate is approximately 2.5%.

Now we want to evaluate the two approximations to the marginal likelihood proposed in Section 5. We proceed by computing the accurate but expensive approximation Z_{ACC}, the less accurate but affordable Z_{APP} and the marginal likelihood of the full GPC, denoted simply as Z_{EP}. In order to show how the approximations depend on the size o the active set, we compute them for $p_{\text{inc}} = \{0.5, 0.6, \ldots, 0.9, 0.99, 1, \}$, $p_{\text{inc}} = 1$ being the full GPC. Figure 3 shows that the three approximations approach the marginal likelihood of the full GC as the inclusion threshold and so the active set increases. As expected, Z_{ACC} is the best approximation, however the computational effort needed to compute it is approximately two order of magnitude larger compared to the cost of computing Z_{APP} and $Z_{\text{EP,A}}$. It is very interesting that even with large values of $p_{\text{inc}} = 0.99$ the size of the active set is still below 10% of the training data and contribution to the log-marginal likelihood from the inactive $Z_{\text{EP}}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}_I, \mathbf{m}_{I|A})$ set basically vanishes, since Z_{APP} and $Z_{\text{EP,A}}$ are essentially the same.

6.2. MNIST

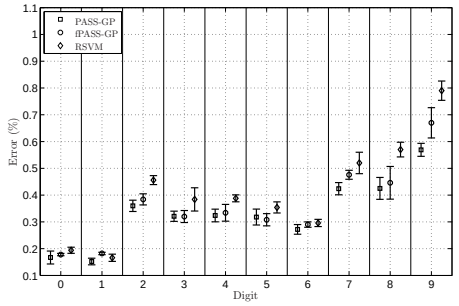
The MNIST digits database has 60000 and 10000 as training and testing examples respectively. Each example is a gray-scale image of 28×28 pixels. The estimated human test error is around 0.2%. The settings used for the algorithm are nearly the same as those for USPS with only two differences. N_{sub} is set 100 since the training set in MNIST is almost ten times larger than USPS and we are not updating the hyperparameter in each iteration but every 10-th, in order to make the training process faster. We also ran our algorithm with hyperparameter updates every single iteration without any noticeable improvement in performance (results not shown). Figure 4 shows test error rates, active set sizes, multi-class errors and running times for each binary classifier based on PASS-GP, fPASS-GP and RSVM using a 9-th degree polynomial covariance function

$$K(\mathbf{x}_i, \mathbf{x}_j) = \theta_1(\mathbf{x}_i \cdot \mathbf{x}_j + 1)^9 \ .$$

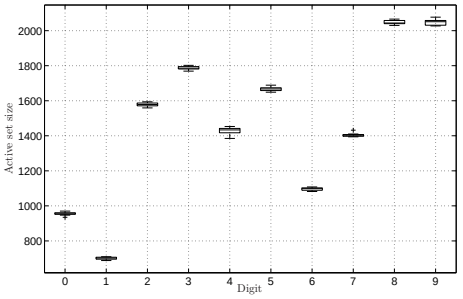
We are this instead for the standard squared exponential covariance function from equation (6), because a polynomial covariance is well known for providing optimal results for the MNIST dataset [15]. Results for the squared exponential covariance function can be found in [6] and confirm that the polynomial covariance behave slightly better for this dataset.

From Figure 4(b) it can be seen that in every case the size of the active set is less than 4% of the training set. The results for fPASS-GP and RSVM were obtained using $M = 2000$. We did try for larger values of M but the reduction in error was not significant compared to the overhead in computational cost. Figure 4(a) shows the classification error for each digit. The performance of the three approaches considered is comparable but letting PASS-GP with an edge over the other two, both in terms of error and variances. Figure 4(c) shows the results of combining the ten binary classifiers. Again, PASS-GP behaves slightly better than the others, however when looking at the run

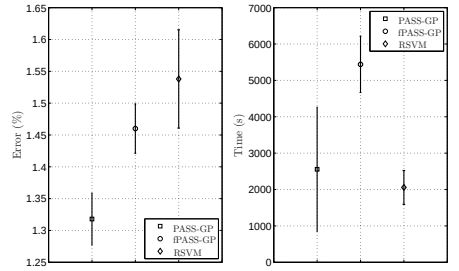
[1] Assuming independent errors the standard deviation on the performance is $\sqrt{\epsilon(1-\epsilon)/N_{\text{test}}}$ giving approximately 0.4% for USPS and 0.1% for MNIST



(a)



(b)



(c) (d)

Figure 4: Error rates, active set sizes and run times for MNIST data. (a) Mean classification errors for each digit task using PASS-GP, fPASS-GP and RSVM. (b) Active set sizes for PASS-GP. Note that fPASS-GP and RSVM use $M = 2000$. (c) Mean multi-class classification errors and (c) average timings over one-against-the-rest classifiers and repetitions. Error bars in (a), (c) and (d) are for standard deviations computed over 10 repetitions of the experiment.

times in Figure 4(d) we can see that RSVM is computationally more affordable than our approaches, even more considering that it uses $M = 2000$. Comparing PASS-GP to fPASS-GP, the former has a smaller mean run time but with larger variance compared to the more expensive fPASS-GP. fPASS-GP is more stable but takes more time because it uses a fixed $M = 2000$.

As far as the authors know these are the first GP results on MNIST using the whole database. IVM [7] has with sub-sampled images of size 13×13 been tested to produce a test error rate of $1.54 \pm 0.04\%$. Seeger [8] made additional tests on some digits (5, 8 and 9) on the full size images without any further improvement. On the other hand, PASS-GP is again comparable with state-of-the-art techniques not including preprocessing stages and/or data augmentation, for example SVM is 1.4% and 1.22% using RBF and a 9-th degree polynomial kernel, respectively. The reported sizes of support vector sets are approximately two times larger than our active sets [15].

6.3. Incorporating Invariances

It has been shown that a good way to improve the overall performance of a classifier is to incorporate additional prior knowledge in the training procedure particularly by means of externally handling invariances of the data. In [15], it is shown that instead of just dealing with the invariances by augmenting the original dataset — which turns out to be unfeasible in many cases, it is better to augment only the support vector set of a SVM. We therefore try the same procedure as suggested in [15] consisting on four 1-pixel translations (left, up, right and down directions) on each element of the active set for USPS and eight 1-pixel translations (including diagonals as well) for MNIST, resulting in new training sets of size $5 \times M$ and $9 \times M$, accordingly. In this case we have used the same settings as in the previous experiments with only two differences. First, the hyperparameters have been set to those found using the original dataset. Second, we made the important observation that in order to get a performance improvement a large active set was needed. For training on the augmented dataset we increased p_{inc} from 0.6 to 0.99 for USPS and 0.9 for MNIST. We conjecture that we can get even better performance — at the expense of a substantial increase in complexity, by increasing p_{inc} in the initial run to get a larger initial active set to work with.

Results in Table 1 show that performance-wise, PASS-GP reached $3.35 \pm 0.03\%$ for USPS and $0.86 \pm 0.02\%$ for MINST on the multi-class task, what is comparable to state-of-the-art techniques. For instance SVM obtained 3.2% on USPS and 0.68% on MNIST with an equivalent procedure. The difference in performance is probably due to our active set not being large enough, since support set sizes reported for SVMs are typically twice as large [15].

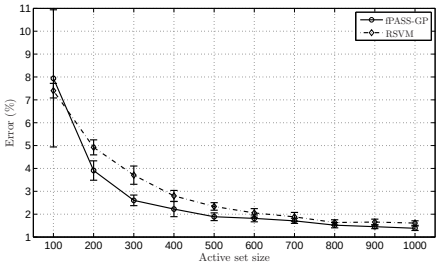
6.4. IJCNN

As final experiment, we want to compare fPASS-GP and RSVM on a common ground. For this purpose we use the IJCNN dataset which is widely used by the SVM research community. It consists of 49990 training examples, 91701 test examples and each observation counts with 22 features. We consider $M = \{100, 200, \ldots, 1000\}$ with squared

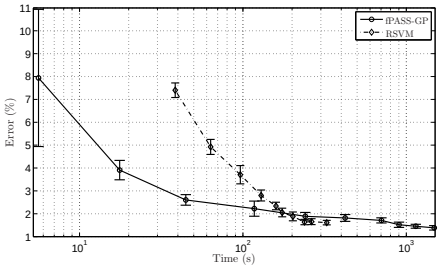
Digit	0	1	2	3	4	5	6	7	8	9
USPS (%)	0.63	0.38	1.01	0.69	0.93	1.16	0.51	0.37	0.59	0.65
Active set	870	442	1251	1316	1654	1425	1242	987	1532	1281
MNIST (%)	0.14	0.14	0.24	0.24	0.29	0.22	0.17	0.35	0.29	0.35
Active set	6505	4372	11401	12988	9776	11960	7360	9872	15194	14790

Table 1: Results for USPS and MNIST using PASS-GP and active set invariances. Figures are averages over 10 and 5 repetitions, respectively.

covariance function and fixed hyperparameters, the latter using the values suggested in [3], that is $\boldsymbol{\theta} = [1\ 1/8\ 1/16]$ for f-PASSGP and $\boldsymbol{\theta} = [1\ 1/8\ 0]$, $C = 16$ for RSVM. Besides, each setting was repeated 10 times to collect statistics. Figure 5 summarizes the results obtained. More specifically, Figure 5(a) shows the mean classification error as a function of the active set. We can see that fPASS-GP is slightly better than RSVM in the entire range of M, besides the former seems to be particularly good for small values of M. When we plot mean errors as a function of the run times — as a proxy for the computational cost, we see that there exist two regimes, one for small values of M where fPASS-GP outperforms RSVM and the other where the cubic complexity of the GPCs start hurting fPASS-GP thus letting RVM with a better error-cost trade-off.



(a)



(b)

Figure 5: Error rates and run times for IJCNN data. (a) Mean classification error as a function of the active set size using fPASS-GP and RSVM. (b) Mean classification error as a function of the run time. Error bars correspond to standard deviations computed over 10 repetitions of the experiment.

7. Discussion

We have proposed a framework for active set selection in GPC. The core of our active set update rule is that the predictive distribution of a GPC can be used to quantify the relative weight of points in the active set that can be marked for deletion or new points from the active set with low predictive probabilities, that make them ideal for inclusion. The algorithmic skeleton of our framework consists on two alternating steps, namely active set updates and hyperparameter optimization. We designed two active set update criteria that target two different practical scenarios. The first we called PASS-GP focuses on interpretability of the parameters of the update rule by thresholding the predictive distributions of GPC. The second acknowledges that in some applications having a fixed computational cost is key, thus fPASS-GP keeps the size of the active set fixed so the overall cost and memory requirements can be known beforehand.

We presented theoretical and practical support that our active set selection strategy is efficient while still retaining the most appealing benefits of GPC: prediction uncertainty, model selection, prior knowledge leverage and state-of-the-art performance. Compared to other approximative methods, although slower than IVM [7] and RSVM [3], PASS-GP provides better results. We did not consider any auxiliary set method like FITC [4] because for task of the size like for example MNIST or IJCNN, it is prohibitive. Additionally, we have noticed in practice that our approximation is quite insensitive to the initial active set selection and also that more than two or three passes through the data do not yield improved performance nor large active set sizes. The code used in this work is based on the Matlab toolbox provided with [2] and is publicly available at `http://cogsys.imm.dtu.dk/passgp`.

The not so satisfying feature of active set approximations, is that we are ignoring some of the training data. Although some of our findings on the USPS data set actually suggest that this can be beneficial for performance, it is of interest to make a modified version where the inactive set is used approximately in a cost efficient way. The representer theorem for the mean prediction and the approximations for marginal likelihood discussed in this paper might give inspiration for such methods. In conclusion, efficient methods for GPs are still much in need when the data is abundant such as in ordinal regression for collaborative filtering.

References

[1] J. Quiñonero-Candela, C. E. Rasmussen, A unifying view of sparse approximate Gaussian process regression, Journal of Machine Learning Research 6 (2005) 1939–1959.

[2] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning, The MIT Press, Cambridge, MA, 2006.

[3] S. S. Keerthi, O. Chappelle, D. DeCoste, Building support vector machines with reduced classifier complexity, Journal of Machine Learning Research 7 (2006) 1493–1515.

[4] A. Naish-Guzman, S. Holden, The generalized FITC approximation, in: J. C. Platt, D. Koller, Y. Singer, S. Roweis (Eds.), Advances in Neural Information Processing Systems 20, MIT Press, Cambridge, MA, 2008, pp. 1057–1064.

[5] T. Joachims, C.-N. J. Yu, Sparse kernel SVMs via cutting-plane training, Machine Learning 76 (2009) 179–193.

[6] R. Henao, O. Winther, PASS-GP: Predictive active set selection for gaussian processes, in: Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on, 2010, pp. 148–153.

[7] N. D. Lawrence, M. Seeger, R. Herbrich, Fast sparse Gaussian process methods: The informative vector machine, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems 15, The MIT Press, Cambridge, MA, 2003, pp. 600–616.

[8] M. Seeger, Bayesian Gaussian process models: PAC-Bayesian generalisation error bounds and sparse approximations, Ph.D. thesis, University of Edinburg (2003).

[9] N. D. Lawrence, J. C. Platt, M. I. Jordan, Extensions of the informative vector machine, in: J. Winkler, N. D. Lawrence, M. Niranjan (Eds.), Proceedings of the Sheffield Machine Learning Workshop, Springer-Verlag, Berlin, 2005.

[10] M. Kuss, C. E. Rasmussen, Assessing approximate inference for binary Gaussian process classification, Journal of Machine Learning Research 6 (2005) 1679–1704.

[11] M. Opper, O. Winther, Gaussian processes for classification: Mean-field algorithms, Neural Computation 12 (11) (2000) 2655–2684.

[12] E. Snelson, Z. Ghahramani, Sparse Gaussian processes using pseudo-inputs, in: Y. Weiss, B. Schölkopf, J. C. Platt (Eds.), Advances in Neural Information Processing Systems 18, The MIT Press, 2006.

[13] L. Csató, Gaussian processes - iterative sparse approximations, Ph.D. thesis, Aston University (2002).

[14] B. Schölkopf, A. J. Smola, Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond, The MIT Press, 2001.

[15] D. DeCoste, B. Schölkopf, Training invariant support vector machines, Machine Learning 46 (1-3) (2002) 161–190.

Appendix E

Latent Protein Trees

In preparation
Journal of the American Statistical Association

Available from DTU Informatics at
`http://imm.dtu.dk/~rh/pmics.pdf`

Latent Protein Trees

Ricardo Henao, J. Will Thompson, M. Arthur Moseley,
Geoffrey Ginsburg, Lawrence Carin and Joseph E. Lucas

February 22, 2011

Abstract

Unbiased, label-free proteomics is becoming a powerful technique for measuring protein expression in almost any biological sample. The output of these measurements are a collection of features (10's to 100's of thousands, only some of which are identified) and their associated intensities for each sample. Each of the features are each associated with a particular polypeptide having a particular number of Carbon-13 atoms and a particular charge state. Because we know that subsets of features are from the same polypeptide, subsets of polypeptides are from the same protein, and subsets of proteins are in the same biological pathways, we know that there is a very complex and informative correlational structure inherent in this data. However, attempts to model this data often focus on the identification of single features that are associated with a particular phenotype that is relevant to the experiment. These associations may be computed from hypothesis testing (with correction for multiple testing) or from various regression models. However, to date there have been no published approaches that appropriately model what we know to be multiple different levels of correlation structure. We present a hierarchical Bayesian model which is specifically designed to model the known correlation structure – both at the feature level and at the protein level – in unbiased, label-free proteomics. This model utilizes the partial identification information from peptide sequencing and database lookup as well as the observed correlation structure in the data set in order to appropriately compress features into metaproteins and to estimate the correlation structure of those identified metaproteins. We demonstrate the effectiveness of the model in the context of a series of proteomics measurements of serum plasma from a collection of volunteers who were infected with two different strains of viral influenza.

Keywords: Proteomics data analysis, H1N1, H3N2, hierarchical factor model, latent proteins, tree representations.

Ricardo Henao is PhD student at the DTU Informatics department, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark (e-mail: rhenao@binf.ku.dk). Joseph E. Lucas is Assistant Research Professor, Institute for Genome Sciences and Policy (IGSP), Duke University, Durham, NC 27710 (e-mail: joe@stat.duke.edu). This work was done when the first author was at the IGSP, Duke University. This work was supported by funding from the Defense Advanced Research Projects Agency (DARPA), number lN66001-07-C-0092 (G.S.G.)

1 INTRODUCTION

Unbiased, label-free, mass spectrometry proteomics data is fast becoming a popular assay in many medical and biological applications. The technique offers the ability to effectively measure large numbers of different proteins (potentially in addition to lipids and metabolites) in a biological samples. The technique works by measuring proteins or polypeptides as they travel down an electric field gradient. Because analytes that are either highly charged or very light weight travel faster, this electric field gradient acts as a technique for fractionating the sample. Ultimately, the constituents of the sample are measured as they hit a target at the end of the electric field gradient, leading to a series of peaks – associated with the intensity of material hitting the target – measured through time. As with any highly complex experimental technique, there are significant challenges, and early work in this field met with a number of notorious failures due to overlapping peaks, batch effect and systematic noise. However, with the newest, high mass accuracy mass spectrometers, and with multiple additional techniques for fractionation (such as liquid or gas chromatography and ion mobility), it is now possible to ensure that intensity peaks do not overlap.

In order to understand the correlation structure in this data, it is important to know a little bit about how the data is generated. First a biological sample is distilled to a solution containing those proteins that are of interest. This may involve simply centrifugation or may involve enrichment of some sort, such as binding to spiked-in proteins. This sample is then broken up via trypsin, which is a serine protease that cleaves proteins on the carboxyl side of arginine and lysine amino acid residues. This processed sample is then filtered through either liquid or gas chromatography, which separates the sample according to some physical property (such as hydrophobicity). The time at which a particular constituent of the sample passes out of the chromatography column is called the retention time. As the sample passes through the chromatograph, it is vaporized and an electric charge is induced on the molecules within (typically by adding hydrogen ions). These charged, short polypeptides they travel down an electric field gradient and are measured as they hit a target at the other end. The intensity of ions hitting the target are measured at regular intervals (called the sampling rate) and the resulting measurements form a trace with visible peaks, called features, that (hopefully) correspond to single polypeptides.

Because the sampling rates are high relative to the size of these features, each feature spans a range of mass-to-charge ratios and retention times. In nature, approximately 1% of all Carbon atoms are Carbon-13 (they contain an extra neutron). Because the accuracy of the mass-to-charge measurement in state of the art mass spectrometers is very high, it is possible to see distinct features for each version of a polypeptide that is present in high enough abundance. This leads to multiple features for each polypeptide, each representing a different, integer number of Carbon-13 atoms. Because they are relatively easy to recognize, these are often collected into a single "isotope group", and the intensity of this isotope group is estimated as the total volume under its associated features. In addition to multiple features due to differing mass, a particular polypeptide

may be present in the data set multiple times in different charge states. Due to the differing physical and chemical properties of different polypeptides, it is possible to attach anywhere from zero to 6 or 7 protons to a randomly chosen polypeptide, and it is common for a particular polypeptide to be present in the vaporized state, before traveling down the electric field gradient, in multiple different charge states. These different charge states can result in multiple isotope groups per polypeptide – depending on the overall abundance of the polypeptide and the relative abundance of each of the charge states. These are typically more difficult to identify than are features that differ by mass, and therefore they are often left as separate measurements in the summarized data set. In what follows, we utilize data that has been pre-summarized at the isotope group level for our statistical models. The process of this summarization is quite interesting, and is the result of significant computational modeling, but is beyond the scope of this paper.

Given a list of isotope group (between 30 and 40 thousand such isotope groups per sample in the infectious disease data set we will discuss later), and their associated intensities for each sample, we wish to understand the resulting correlation structure. This will allow us to substantially improve upon standard hypothesis testing coupled with controlling for multiple testing. There are inherently two different types of correlation present in label-free, unbiased proteomics data. First, each isotope group originates from a particular protein and there typically many isotope groups per protein in the data set – particularly for proteins that are highly abundant in the original sample. Second, some collections proteins are expected to behave similarly because they are in the same biological pathways. This will result in correlation between proteins (and therefore correlation between isotope groups) that is of distinct etiology. We would like to model these two sources of correlation separately.

Clearly, these two sources of correlation between isotope groups are confounding without some additional information allowing us to distinguish them. Luckily, there is an additional feature of the data that allows just this distinction. The mass spectrometers used to generate our infectious disease data operate alternately at two distinct energy levels. Thus, for each retention time, we have both a low energy trace, in which we measure the mass-to-charge ratio of the polypeptides that were generated by trypsin cleavage, and a high energy trace in which those polypeptides are additionally broken, at random locations, into even shorter polypeptides. This permits, for some isotope groups that are present at high concentrations, the identification of its specific amino acid sequence. These sequences are then associated, through sequence alignment to protein sequences in a public database, to particular proteins. Thus we have, for a limited subset of the isotope groups a (possibly imperfect) estimate of its originating protein.

The statistical model presented in this paper is essentially a factor model especially designed to deal with the particular challenges of proteomics data analysis and the problem of differential expression in mass spectrometry proteomics. Although there are many excellent approaches to factor modeling in the statistics literature that are similar to ours (West, 2003; Lucas et al., 2006; Carvalho et al., 2008; Lucas et al., 2009; Henao and Winther, 2011), they are mostly targeted to gene expression analysis. Our

approach has the following leading protein data oriented features: (i) Subtraction of large scale correlation structure very likely to arise from technical rather than biological variability, for instance due to batch effects. (ii) Uses identifications of isotope groups and proteins as prior knowledge, but allows for those identifications to change by admitting that miss annotations can occur. (iii) Admits that sections of a protein might be post-translationally modified and therefore resulting in expression profiles that are not representative of the protein in its whole integrity. (iv) Takes advantage of the correlation structure at the protein level to build a nested hierarchical representation of isotope group into proteins, then into groups of proteins we call *parent proteins*. (v) Can be used to build predictive models at a protein (or parent protein) level rather than isotope group expression based on condition specific protein enumeration. While some of the features mentioned can be found in already proposed models, however is the ability of simultaneously modeling all of them what makes our model so unique.

There is some work in the literature for proteomics specific data analysis although more oriented towards differential protein expression. For example Daly et al. (2008) describe a mixed effects model for estimating protein level differential expression, however each protein is handled independently discarding any possibility for protein correlation. Karpievitch et al. (2009) presents a statistical model for protein-level abundance that accounts for missing values in the data well because the peptide is not available independent of its abundance or because its abundance is too low to be detected by the machine. Both, Daly et al. (2008) and Karpievitch et al. (2009) assume that the peptide-protein associations are correct and employ maximum-likelihood estimation, failing at properly quantifying the inherent uncertainty of LC-MS based data. A factor model similar to the one proposed in this paper was elaborated in Lucas et al. (2011), but it does not consider separately the correlation structure between isotope groups and that between proteins.

From a factor modeling side, isotope group-protein interactions can be modeled in principle using the sparse factor modeling framework introduced by Lucas et al. (2006). In addition to sparsity, non-Gaussian and non-parametric modeling of proteins can be achieved using the approaches later introduced by Henao and Winther (2011) and Carvalho et al. (2008), respectively. Unfortunately, none of them considers the possibility of systematic effects subtraction or hierarchical representation for the estimated protein expression structure. From the hierarchical side of our model, most of the work in the literature is more concerned with hierarchical clustering instead for structured factor modeling. For instance, Neal (2003) introduces the so called Dirichlet diffusion tree, a family of prior distributions over multivariate distributions for hierarchical density modeling and clustering. Heller and Ghahramani (2005) proposes a probabilistic method for hierarchical clustering. Although, the method uses a probabilistic interpretation to build the tree, it does not attempt to model the data density nor places a prior over tree structures. Blundell et al. (2010) extends the work of Heller and Ghahramani (2005) by means of allowing the tree to have an arbitrary branching structure in what they call *rose trees*. Teh et al. (2008) propose an agglomerative hierarchical clustering method based on coalescents. Rai and Daume III (2009) use the method of Teh et al.

(2008) to provide the loading matrix of a factor model with a tree structure, however such a structure is only used for visualization and interpretation, meaning that it does not contribute to the density model produced by the factor model. Adams et al. (2010) introduces a model for hierarchical clustering based on nested stick-breaking processes. Its most interesting feature is that observed variables (proteins) can live at any node of the tree. Our model's protein interaction hierarchy is conceived in a three layer such that isotope sit in the bottom layer as leaves, subsequently they merge into an intermediate layer of latent proteins that we estimate from observed data with the aid of annotation obtained from standard proteomics analysis, and finally the top layer is a binary tree representation that summarize similarity of protein expression profiles into groups of parent proteins.

The remaining of this paper is organized as follows, Section 2 describes the H1N1/H3N2 based proteomics data used to illustrate the features of the model fully introduced in Section 3. The results are presented in Section 4 and we conclude with a discussion in Section 5.

2 DATA

We start from proteomics data obtained from 43 patients part of the DARPA H1N1/H3N2 plasma project (Zaas et al., 2009). From the entire pool, 24 patients were exposed to H1N1 whereas the remaining 17 were exposed to H3N2. For each patient, four samples were taken at different reference time points ($t = 0, 0.2, 0.8, 1$). Besides, each sample was categorized as symptomatic (SX) o asymptotic (ASX) based on self-reported symptom scores as well as viral culture. The samples of the H3N2 study could not be processed in a single run, thus two batches were produced, the first containing only samples from time points $t = 0$ and $t = 1$, i.e. two batches of 40 samples each. In summary, we have $N = 172$ samples of two studies (H1N1 and H3N2) divided in three batches (H1N1, H3N2_1 and H3N2_1) and two conditions (SX and ASX).

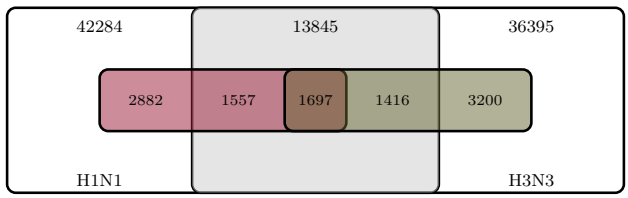


Figure 1: Data composition. Red and green areas represent annotated subsets of H1N1 and H3N3, respectively. The gray area is the subset of IGs aligned from the whole pool using the alignment model and the brown area is the subset of isotope groups with identical annotations after Mascot/PeptideProphet analysis.

After peptide quantitation and technical reproducibility assessment, several data dependent LC-MS analyses were performed as quality controls. The data to be analyzed

consists of a matrix of +40000 IG expressions (rows) per sample (columns). An Isotope group is defined as the total volume of the spectral peaks coming from the same peptide that differ only by the number of carbon 13' incorporated. This quantitation is monotonically related to the concentration of that peptide in the original sample, thus is what we finally use in our statistical analysis. Peptide identification and annotation was done using a combination of Mascot (`http://www.matrixscience.com/`) and the PeptideProphet algorithm (`http://peptideprophet.sourceforge.net/`, Keller et al., 2002). By identification and annotation we mean assigning peptides and parent proteins names to each isotope group available. In addition, IG sets from the three batches (H1N1, $H3N2_1$ and $H3N2_2$) were aligned using the statistical model described in Section S.1. From all available isotope groups, 13845 were found both in H1N1 and H3N2 but only 4670 were provided with annotation, see Figure 1 for a graphical summary. From the set of 4670 annotated IGs, only 1697 share the same annotation according to Mascot-PeptideProphet analysis. The remaining 2973 IGs consist of annotations transferred from H1N1 to H3N2 or vice versa, using the alignment model. The set of annotations itself, include 239 proteins from which 106 are assigned to more than a single IG.

The data has relatively low amount of missing values, most of them introduced because of resolution limitations of the LC-MS technology. The inconvenient is that the 2% of missing values in the dataset are very unevenly distributed, in particular, $H3N2_1$ has 10.3% of them, $H3N2_2$ 0.7% and H1N1 up to 2.5%. We removed one sample because it had more than 30% missing values in the set of annotated IGs.

3 MODEL DEFINITION

As mentioned in the previous section, we work at an isotope group level. We model the expression of a sample n from isotope group i and batch m, x_{in}^m as a linear combination of four different contributions, namely noise, protein expression, systematic and batch effects as follows

$$x_{in}^m = \mu_i^m + \sum_l a_{il} z_{ln} + b_{ik} w_{kn} + \epsilon_{in} \ , \tag{1}$$

where μ_i^m is the mean batch effect due to batch m, factors $z_{1n}, z_{2n}, \ldots$ are meant to capture systematic effects, w_{kn} is the expression value of protein k, a_{il} and b_{ik} are weights for the systematic effects and the protein expression, respectively, and ϵ_{in} is measurement uncorrelated noise. The model in equation (1) can be seen as a specialized version of a factor model having three levels of resolution: the coarse level composed by the batch means and the measurement noise, the middle level in the form of systematic effects and the in detail level dedicated to model the more specific protein expressions. The main motivation for having such an intermediate resolution level is to clean the observed data as much as possible aiming to obtain protein expression profiles that hopefully better reflect true biological rather than technical variability. Provided that proteins are not observed directly but need to be inferred, from now on we call them *latent proteins* to

avoid misinterpretations. For N samples grouped in N_B batches and having p isotope groups, N_F systematic factors and N_P proteins, we can write equation (1) in matrix form as

$$\mathbf{x}_n^m = \boldsymbol{\mu}^m + \mathbf{A}\mathbf{z}_n + \mathbf{B}\mathbf{w}_n + \boldsymbol{\epsilon}_n \ , \tag{2}$$

where $\mathbf{x}_n^m$, $\boldsymbol{\mu}^m$, $\mathbf{z}_n$, $\mathbf{w}_n$ and $\boldsymbol{\epsilon}_n$ are $p \times 1$ vectors, $\mathbf{A}$ is a $p \times N_F$ matrix and $\mathbf{B}$ is a $p \times N_P$ matrix. A-priori, we restrict isotope i to be associated to a single latent protein k, each row of $\mathbf{B}$ contains a single non-zero element b_{ik}. Auxiliary, we also define $\mathbf{u}$ as the p-dimensional vector of assignments with elements $u_i = k$ if $b_{ik} \neq 0$ and K_k as the subset of isotope groups associated to latent protein k, thus $K_k \subset \{x_1, \ldots, x_p\}$. We assume that $\boldsymbol{\mu}^m$, $\mathbf{z}_n$, $\mathbf{w}_n$ and $\boldsymbol{\epsilon}_n$ are mutually uncorrelated as well as the systematic factors with each other. We cannot assume zero correlation of the latent proteins because we know they tend to co-express in groups conforming pathways, thus we are also interested in capturing and interpreting such a covariation structure.

Our goal is then first to separate technical from biological variability through $\mathbf{z}_n$ and $\mathbf{w}_n$, respectively. Second, we want to estimate relative protein concentration using $\mathbf{B}$, i.e. the expression pattern of an isotope group is a scaled version of the expression pattern of a latent protein. Third, from annotation we have protein assignments for a subset of IGs, thus a latent protein is assigned after inference to a collection of IGs which may be dominated by a particular protein, but containing also IGs from other proteins as well. As a result, we want to characterize the latent proteins. This is another reason to differentiate between proteins and latent proteins.

3.1 Prior Distributions

We need to specify prior distributions for each one of the elements in the right hand side of equation (2). The noise component is zero-mean Gaussian with diagonal covariance matrix $\boldsymbol{\Psi}$ to allow for different noise variances for each isotope group, similar to the protein level aggregation of (Clough et al., 2009). An element specific prior for $\boldsymbol{\Psi}$ is set to flat gamma hyperpriors with shape $t_s = 1.1$ and rate $t_r = 0.001$, to keep the variance bounded away from zero. The mean batch effects have a Gaussian priors with mean $t_m = 8$ and small precision $t_p = 0.01$ to accommodate for the overall mean expression profile of the data.

Systematic effects The main purpose of the systematic effect factors is to capture variability of large groups of isotope groups that cannot be regarded as non-specific measurement noise and are most likely due to technical rather than biological variability, thus non-representative of protein specific expression profiles. This can be achieved by letting $\mathbf{z}_n$ to have a heavy tailed distribution to allow some of its elements to disagree with the substructure defined by loading matrix $\mathbf{A}$. In practice we use Laplace distributions parameterized as scale mixtures of Gaussians with exponential mixing densities to facilitate inference (Andrews and Mallows, 1974). We further place a conjugate gamma hyperprior on the rate of the Laplace distribution with parameters $\ell_s = 4$ and $\ell_r = 2$.

The number of systematics factors N_F is not critical because (i) we are not concerned about the interpretability of $\mathbf{A}$ and (ii) because we have observed empirically that the variance explained by the systematic effect factors saturates as N_F increases, see Section 4.4. In the experiments we use $N_F = 5$. For the elements of $\mathbf{A}$, we assume for independent standard Gaussian priors to reflect that systematic effects can in principle span all isotope groups. This setting also minimizes identifiability issues between $\mathbf{B}$ which is very sparse and $\mathbf{A}$ that is in essence dense.

Protein profiles We can identify two main features in the protein model, one is that each isotope group can be assigned to only one latent protein and the other that latent proteins correlate each other. For the first desideratum we set a prior hierarchy as follows

$$b_{i,u_i} \sim \mathcal{N}(0,1) , \quad u_i \sim \text{Discrete}(\mathbf{v}_i) , \quad \mathbf{v}_i \sim \text{Dirichlet}(\boldsymbol{\rho}_i + \boldsymbol{\kappa}) ,$$

where isotope group i is associated with latent protein u_i with probability $\mathbf{v}_i$. The prior for the vector of N_P probabilities $\mathbf{v}_i$ is set using the information obtained from annotation (Mascot-PeptideProphet-alignment model). In particular, κ_k is the total number of isotope groups identified to protein k and ρ_{ik} enforces annotation by having a large value ρ_0 if isotope i comes from protein k or zero otherwise. When ρ_0 is set to a large value, we increase the importance of the annotation while decreasing the relevance of the correlation structure of the data. We found empirically that our default $\rho_0 = 1000$ offers a good trade-off between prior-likelihood relevance, see Section 4.4. If an isotope group happens to be unannotated then $\boldsymbol{\rho}_i = \mathbf{0}$. This setting allows us to prefer annotated and abundant proteins for which we can most likely obtain plausible expression profiles. Conversely, it will certainly discourage a rare protein with say only one or two isotope groups associated to it. We know the chances of capturing a signal from a rare protein from a typical study (N in the lower hundreds and p in the thousands) is very limited unless its signal turns out to be considerably strong, in which case the proposed prior must be able to readily pick it.

The easiest way to introduce correlations between latent proteins is to provide $\mathbf{w}_n$ with a multivariate Gaussian distribution and then infer the correlation structure by placing an inverse Wishart prior on its covariance matrix. We know that groups of proteins might have similar expression profiles for different reasons, for example because they are structurally similar, mediate similar biological processes, share a pathway, etc. In that sense, it could be interesting to model such similarities in a more interpretable way. This is why we choose to place a prior over binary trees on the N_P latent proteins, so we can both model correlations and have an interpretable representation of isotope groups, latent proteins and their interactions. Figure 2 shows the concept for a particular problem with $p = 15$ isotope groups distributed in $N_P = 5$ proteins. We can see a hierarchical clustering structure in which for instance w_4 and w_5 are more similar than w_3 and w_4, thus more correlated. The *pseudo time* t serves as similarity measure so that more alike proteins merge sooner in time, allowing us to directly quantify their pairwise or group-wise distances. The proposed hierarchy also reflects the fact that isotope groups and latent proteins must lay in different levels and that parent proteins are proxies for

the average profile of groups of proteins. It is clear that allowing proteins to be placed anywhere in the tree is more in line with the notion of functional pathway, however we will not try to pursue this in here.

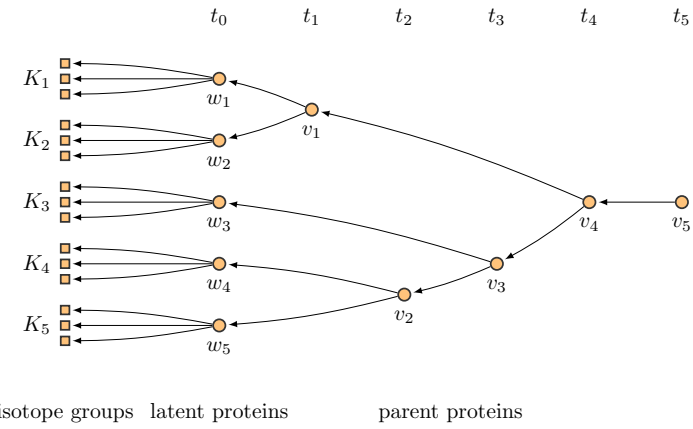


Figure 2: Latent protein tree structure. Particular tree with $N_P = 5$ and three isotope groups assigned to each latent protein. The pseudo time variable t defines the merging points.

Assuming that the tree structure π is given, we can easily compute the marginal distribution of each node in the tree using *belief propagation* (Pearl, 1988). To do so, we use *message passing* by first propagating messages from the leaves to the root and then in the opposite way. This is advantageous because the marginal for node j can be computed efficiently as the product of messages going into it. For the leaves to root pass we can identify two kinds of messages from Figure 2, namely those going from isotope groups to latent proteins and those from latent proteins to parent proteins. We can write then

$$\begin{aligned} \mu_{v_{s,j}\to v_j} &= \int p(v_j|v_{s,j},\Delta_{s,j})p(v_{s,j})dv_{s,j} \ , \\ \mu_{K_k\to w_k} &= \int p(w_k|K_k,\mathbf{b}_k)p(K_k)dK_k \ , \end{aligned} \tag{3}$$

where $s = \{l, r\}$ (left,right) denote siblings locations w.r.t. v_j, $\Delta_k = t_k - t_{k-1}$ and the tree π is the set of all sibling assignments $\{s, j\}$ and merging times $t_0, \ldots, t_{N_P}$. Note that $\mu_{K_k\to w_k}$ is the conditional posterior of latent protein w_k, $p(K_k)$ is the prior distribution of K_k and $\mathbf{b}_k$ is a column of $\mathbf{B}$. Using equation (3) we can partially update nodes' marginals as

$$p(v_j|\boldsymbol{\theta}) \ = \ \mu_{v_{l,j}\to v_j}\mu_{v_{l,j}\to v_j} \ , \quad p(w_k|\boldsymbol{\theta}) \ = \ \mu_{K_k\to w_k} \ , \tag{4}$$

where $\boldsymbol{\theta}$ is a shorthand for all the hyperparameters, i.e. $\boldsymbol{\theta} = \{\pi, \mathbf{B}, \ldots\}$. Besides, assuming that we can marginalize out v_j we can then compute

$$Z_{v_j|\boldsymbol{\theta}} = \int p(v_j|\boldsymbol{\theta}) p(v_j) dv_j \ . \tag{5}$$

For the root to leaves pass we just have to complete the update of the marginals in equation (4) using

$$p(v_j|\boldsymbol{\theta}) \ = \ \mu_{v_{l,j}\to v_j}\mu_{v_{l,j}\to v_j}\mu_{v_p\to v_j} \ , \ \ p(w_k|\boldsymbol{\theta}) \ = \ \mu_{K_k\to w_k}\mu_{v_p\to w_k} \ , \tag{6}$$

where v_p is the parent variable of v_j or w_k. Now, we need to specify distributions for $p(K_k)$, $k(v_j|v_{s,j}, \Delta_{s,j})$ and $p(\pi)$. We set them as follows

$$p(v_j|v_{s,j}, \Delta_{s,j}) = \mathcal{N}(v_j|v_{s,j}, (t_{s,j} - t_j)\boldsymbol{\Phi}) \ , \tag{7}$$

$$p(K_k) = \delta_{K_k} \ , \tag{8}$$

$$p(\pi) = \prod_{j=1}^{k-1} \exp\left(-\binom{n-j+1}{2}\Delta_j\right) \ , \tag{9}$$

where $p(K_k)$ is a point mass at K_k that reflects that the conditional posterior of w_k is computed based on N observations of the variables in K_k. In a slight abuse of notation, this means that both w_k and v_j represent variables and N-dimensional vectors interchangeably. The transition distribution for the tree is Gaussian with diagonal covariance matrix $\boldsymbol{\Phi}$, scaled by pseudo time differences $\Delta_{s,j}$. The covariance matrix is set to diagonal to accommodate for observations having different levels of variance, outliers and missing values. The prior for the tree structure is an Kingman's coalescent, a convenient and powerful method for describing the ancestral tree of a set of individuals (Kingman, 1982a). The process is referred as the coalescent because it describes the probability of coalescent events, i.e. the time point in the genealogy where two individuals merge. For n individuals (proteins), Kingman's n-coalescent is nothing but a distribution over genealogies of those n individuals considered or equivalently, over binary trees (genealogies) with n leaves. In particular, the n-coalescent is a continuous-time Markov process that starts at time $t = 0$ with all $\{1, \ldots, n\}$ individuals and evolves in time, merging pairs of elements until only one is left. Every pair of individuals coalesce independently with rate 1., thus the time between events j and $j-1$ is $\Delta_j \sim \exp\left(\frac{n-j+1}{2}\right)$ and the pair to be merged is chosen uniformly from those available at time $j-1$. Among the interesting properties of this prior distribution we have that the marginal distribution over tree structures is uniform and independent of the merging times and it is infinitely exchangeable, see Kingman (1982a) and Kingman (1982b) for further details.

3.2 INFERENCE

Bayesian analysis is performed using Markov chain Monte Carlo (MCMC) to produce samples from the posterior of all parameters in the model, namely $\boldsymbol{\mu}^m$, $\mathbf{A}$, $\mathbf{z}_n$, $\mathbf{B}$, $\mathbf{w}_n$, $\boldsymbol{\Psi}$,

v_j and π. The most relevant summaries involve posterior samples from the latent, parent proteins, and the hierarchical structure encoded by $\mathbf{B}$ and π. Nearly all quantities of interest are updated using Gibbs sampling except for the tree components that require sequential Monte Carlo (SMC) updates with resampling. In all the experiments we collected 1500 samples for posterior computations after a burn-in period of 3000 iterations. For problems similar to the one described in Section 2 we can expect the sampler to take a couple of hours in a desktop machine. All the details of inference can be found in Appendix A.

Summaries for most of the important quantities of the model can be computed in the usual way by means of histograms and empirical quantiles. Summarizing trees on the other hand is not an easy task because tree averaging is not a well defined operation. We could in principle use the pseudo time variable to build a pairwise distance matrix between latent proteins and then attempt to build a tree from a summary of such a *similarity* matrix. The problem is that we do not have any guarantee that this average of binary trees will produce another binary tree as well. In fact, we tried this approach empirically both with artificial and real data, and found that the tree built using the mean or median of the similarity matrices collected during inference oftentimes produce trees with non-binary branches thus not matching our prior assumptions. In view of this, we decided to select a single tree from all the available samples using as criterion the tree marginal likelihood based on equation (5).

4 RESULTS

The results of the case study based on data described in Section 2 can be divided in four parts. The first addresses the general more features of the model. The second focuses on latent proteins that are discriminant of the symptomatic/asymptomatic status of the samples. The third highlights the features of the latent protein tree representation of the model. Finally, the fourth explores the sensitivity of the model to its most critical parameters, namely the number of systematic factors N_F and the annotation enforcing parameter ρ_0.

4.1 Resolution, integrity and identity of the model

First, we want to highlight some of the most relevant features of our factor model using the data previously described. We collected 1500 posterior samples after a burn-in period of 3000 MCMC iterations with hyperparameters as explained in Section 3.1. Figure 3 shows in a first place the data matrix as a 4670×171 heatmap with observations (columns) grouped in batches $\{\text{H1N1,H3N2}_1\text{,H3N2}_2\}$ and missing values represented as black spots. The remaining three heatmaps in the figure also with columns grouped in batches, picture respectively the batch means $\{\boldsymbol{\mu}^m\}_{m=1}^{N_B}$, systematic factors $\mathbf{Z}$ and latent protein expressions $\mathbf{W}$. The residuals $\{\boldsymbol{\epsilon}_n\}_{n=1}^{N}$ are presented as a histogram to display its heavy-tailed behavior that is mostly product of data heterogeneity, e.g. outliers and missing values. From the heatmaps it is easy to recognize the different levels of resolution

captured by the model. The batch means explain the data in a broad scale by catching the more salient batch effect features. The systematic factors while still correlate with the batch grouping, capture more detailed elements of variability at an observation-wise level, thus outliers and internal batch features are picked. Finally, latent protein expressions represent the most fine variability in the data. Note that in contrast to $\{\boldsymbol{\mu}^m\}_{m=1}^{N_B}$ and $\mathbf{Z}$, $\mathbf{W}$ no longer shows evidence of those obvious batch effects exhibited by the other two. Furthermore, we compared each of 106 the latent proteins to the batch set indicators and found no significant correlation for any of them. The posterior summaries of every element of Figure 3 were obtained using medians and the histogram of the residuals was smoothed using kernel density estimation.

Next we want to provide some details about the integrity of the estimated latent proteins and by integrity we mean how different are IG-protein assignments a-priori and a-posteriori. Figure 3(b) shows two bar plots: the orange bars represent sorted a-priori counts for each one of the 106 unique proteins obtained from annotation, so that each count is the number of IGs associated to a given protein. The green bars are posterior summary counts computed as the mode of the vector of assignments $\mathbf{u}$ after inference. Although different, both bar groups have a similar decreasing trend. As an attempt to quantify the stability of $\mathbf{u}$ we computed the number of IGs for which the number of MCMC posterior draws targeting the mode of u_i is less than a threshold, 50% in this case (see Figure S1). The resulting set of 134 IGs (3%) is regarded as non-stable in the sense that each of its IG members have been assigned to a variety of latent proteins during inference, thus substantially decreasing the frequency of the modes. Figure 3(b) also shows the names of 7 proteins: ATRN-H, FA9-H, HBA-H, PLEC1-H, PLGA-H, ECM1-H and TLN1-H. These happened to be empty, meaning that according to the posterior summary those 7 proteins do not have any IGs associated to them. Interestingly, all empty proteins only have two IGs a-priori associated to them. The most likely scenario for this to happen is (i) the IGs associated to the proteins being emptied do not have signals strong/consistent enough for the model to keep them or (ii) the IGs are wrongly annotated, hence their expression profile correlated better with another protein in the model. Figure S2 shows that the histogram of the number of empty proteins has a very clear peak at 7 proteins and a very small standard deviation, thus empty proteins do not change much during inference. As shown in Figure 1, from the 4670 IGs included in the model, 1697 are assigned to the same protein in both H1N1 and H3N2 datasets — HxNx from now on, 1557 are annotated in H1N1 and matched to H3N2 and 1416 are annotated in H3N2 and matched to H1N1. In numbers, integrity is here defined as the proportion of IGs that keep their original annotation names. In practice, we do not expect integrities to be close to one because we know that proteins are prone to miss-annotations or could be not representative of the expression pattern of its parent protein. The latter can happen for several reasons for instance post-translational modifications. Besides, only 36% of the IGs at hand share annotations in HxNx whereas the remaining 64% of the annotations were transferred from one set to the other using the alignment model, thus increasing the risk for miss-annotation. Figure 3(a) shows that in average the total integrity is 49%, hence the remaining 51% of the annotated IGs changed their

original protein assignments. More specifically, 26% of the integrity is for IGs annotated in HxNx, corresponding to 68% of that subset of 1697 IGs. The remaining 11% and 12% are the total integrities for datasets H3N2 and H1N1, respectively. The most important element of Figure 3(a) is that IGs annotated in both sets tend to be more reliable than those matched using the alignment model, since the relative integrity of HxNx is almost twice the values of H1N1 and H3Ns, that is 35% and 38% respectively. Figures 3(c), 3(d), 3(e) and 3(f) show relative integrity histograms for H1N1, H3N2 and HxNx, together with total integrities computed during inference. In every case we report in the x-axis the number of IGs instead of the proportion and in the y-axis the frequencies rather than counts. We consider the model has a stable IG-protein assignment state because each histogram has a shape with a well defined mode and reasonable standard deviations (12 IGs at the most). We ran several independent chains and we found similar results also supporting that the model has stable integrity summaries (results not shown).

Lets examine now the composition of the estimated latent proteins. Since we have a model with 106 latent proteins and we know from Figure 3(a) that 51% of the IGs were reassigned from their original annotations, it is reasonable to assume that each estimated latent protein is at its best, a collection of IGs with a large proportion of them having the same protein assignment as the latent protein label. When the latter occurs we say that the latent protein is identified because the data and the model both agree with the label we provided it a-priori. Figure 4(a) presents a histogram of the number of identified proteins in which we can see that in average there are 77 (72%) identified proteins. Taking a step further, we compute the largest proportion of IGs having a particular protein label and we call it purity. If a latent protein happen to have 100% purity it means that all their IG members have the same protein label then it is fully identified, conversely if the purity is low it means that there is too much diversity among the members of the latent protein, thus making it hard to label. Figure 4(b) and Table S1 show purity values for each of the 99 nonempty proteins. Circles correspond to identified proteins and crosses to proteins that need to be relabeled because their initial assignment does not agree with the majority of their IG members. We found that 21 latent proteins were relabeled as also indicated in Figure 4(b) and Table S1. Figure 4(c) shows some examples of latent protein compositions. For instance CRP-H has all its members labeled as CRP-H, thus 100% purity. Proteins like APOB-H, LPB-H, CO9-H and A2GL-H have purities larger than 60%. For FHR1-H nearly half of its members are also FHR1-H. In cases like TETN-H, the leading protein is not TETN-H, it has been relabeled to ANT3-H. However there is still an identified protein under the name of ANT3-H having a purity close to 50%, interestingly with a low content of TETN-H labeled proteins. This can happen for several reasons for example two proteins with similar expression profiles but large variability, making them easy to confound or simply because there are subgroups of IGs from the same protein with different expression patterns. We observed that such large variability could affect even proteins with a large sets of a-priori assigned IGs like TETN-H and CLUS-H (+100 IGs each).

4.2 Discriminant latent proteins

Provided with the symptomatic/asymptomatic status of each observation in the dataset, we decided to find whether there are latent proteins correlating with such status. For this purpose we fit individual linear discriminant classifier for each latent protein at each MCMC draw and estimate the classification accuracy as the area under the ROC curve (AUC, Receiver Operating Characteristic, Fawcett, 2006). Figure 6 shows results for five of the most discriminant latent proteins: FHR1-H, CRP-H, LBP-H. A2GL-H and CO9-H. Each panel shows AUC values with corresponding 50% credible intervals for two different partitions of the whole dataset, (i) by disease: H1N1, H3N2 and HxNx, and (ii) by normalized time: $t = \{0, 0.2, 0.8, 1\}$ and where ALL just means all time points. Figure 5(a) shows that FHR1-H has a decent performance when using the entire dataset HxNx and all time points. It is particularly good at time points $t = \{0, 0.8, 1\}$ when H3N2 is used and particularly bad for $t = 0.2$ and H1N1. Figures 5(b) and 5(c) are very similar and correspond to latent proteins CRP-H and LBP-H, respectively. Their performance on the full dataset is not any better than using FHR1-H however when H3N2 is used alone at time points $t = \{0.8, 1\}$ the performance is very good. Latent proteins A2GL-H and CO9-H in Figures 5(d) and 5(e), respectively, produce similar results to those of CRP-H and LBP-H but with less variance and some reduced performance when the subsets H3Ns and $t = 0.8, 1$ are used.

In general terms, we can say that the main source of classification error is the H1N1 subset, suggesting that not only that H3N2 is somewhat an easier task but that H1N1 and H3N2 definitely do not share the same discriminant features. It is worth noting that we could not find any set of latent proteins proteins that perform well on the H1N1 subset and that we could not find any suitable explanation at least from a biological point of view for this to happen. Focusing on the set producing good results, i.e. H3N2, time points $t = 0, 0.2$ can be addressed with FHR1-H, $t = 0.8$ with FHR1-H or CRP-H, and $t = 1$ with CRP-H or LBP-H. Figure S3 show ROC curve samples for FHR1-H using all time points and CRP-H and LBP-H restricted to $t = 0.8$.

4.3 Latent protein tree

As already described in Section 3, the prior distribution for the set of latent proteins allows to to build tree representation of its elements in a hierarchical clustering fashion. Figure 7 shows the latent protein structure corresponding to the MCMC draw producing the largest tree marginal likelihood based upon equation (5). There is some straightforward groupings in the tree mostly corresponding to protein variants like APOC2-H and APOC3-H, CO8G-H and CO8B-H, FIBG-H and FIBB-H, F13A-H and F13B-H, all of them having similar profiles when looking at their estimated signatures (results not shown). In other cases like the COx family, they show great diversity in their profiles then they turn out to be quite spread in the structure. In red, we show proteins that have been relabeled, for example the triplet ITIH1-H, ITIH2-H and C1S-H, where the latter two were labeled during inference to ITIH1-H, see Table S1. The latent proteins colored in orange correspond to four of the most discriminant variables of the model as shown in Figure 6.

As a result, not only they have similar classification performance but they also conform a subtree that hierarchically link them in terms of their correlation similarities. Note also that similar to Figure 6, the pair LBP-H and CRP-H is more alike than the pair CO9-H and A2GL-H, however they still merge at some point (in pseudo time). Figure 8 shows the subtree structure along with a scatter of the expression values of each latent protein. Each panel in the figure shows expression in the y-axis and data grouping in the x-axis. Data to the left hand side of the dashed vertical line corresponds to the asymptotic set whereas the other side contain symptomatic observations. Each side is further grouped according to time, so points closer to the dashed vertical line are for $t = 0$ (green), then $t = 0.2$ (yellow), $t = 0.8$ (red) and the farthest to $t = 1$ (purple). The good separation of observations from times $t = \{0.8, 1\}$ agrees with the classification results shown in Figure 6.

4.4 Sensitivity to N_F and ρ_0

There are two parameters in the model that seem to be more critical than the others, they are the number of systematic factors N_F and the pseudo count that enforces annotation ρ_0. The former is selected so the systematic factors capture enough variability without washing out IG specific variability. This means that if N_F, is too small the latent proteins will be affected by batch effects for example, whereas if N_F is too large $\mathbf{Z}$ will start reflecting protein expression. Figure 9 shows HxNx integrity, total integrity, number of identified and empty proteins for five different values of N_F, namely $\{2, 4, 6, 8, 10\}$. Specific integrities are included in the supplementary material as Figures 3(a) and 3(b). Also in Figure 9, 95% credible intervals for each setting. From Figures 8(a) and 8(b) we can observe that increasing N_F always increases integrity. This is reasonable because increasing the variance explained by $\mathbf{Z}$ will decrease the contribution of $\mathbf{W}$, thus turning the contribution of the prior to the posterior of the IG-proteins assignments stronger. Note however that the behavior of the curves in Figures 8(a) and 8(b) is not linear but tends to *saturate*. Specifically, the integrity change from $N_F = 2$ to $N_F = 6$ is greater than 600 IGs whereas between 6 and 10 systematic factors is nearly 150 IGs. The number of identified proteins shown in Figure 8(c) does not change dramatically but decreases when $N_F = 10$ due to $\mathbf{Z}$ taking too much variability from $\mathbf{W}$. Figure 8(d) shows the number of empty proteins behaving quite stable between $N_F = 4$ to $N_F = 8$.

Changing ρ_0 it is more critical because it controls the informativeness of the annotation based prior. In particular, if $\rho_0 = 0$ the prior does not care for annotation but if $\rho_0 \to \infty$ the model is not allowed to relabel IGs. It is clear then that any of these two extreme cases is a bad choice for a prior. Figure 10 shows HxNx integrity, total integrity, number of identified and empty proteins for five different values of ρ_0, namely $10^{\{1,2,3,4,5\}}$. The specific integrities are shown in Figure S4. The integrities and number of identified proteins depicted in Figures 9(a), 9(b) and 9(c) increase approximately linear with the power of ρ_0, however the effect of ρ_0 on the number of identified proteins it is not as strong as compared to the integrities. For example, a two orders of magnitude change in ρ_0 only increases the number of identified proteins by 10. The number of empty proteins

on the other hand, decreases with ρ_0, nevertheless such a change does not seems to be significative.

Figures 9 and 10 lead us to conclude that $N_F = [5, 8]$ and $\rho_0 = [1e3, 1e4]$ are reasonable choices in our case. This statement also considers that: (i) for smaller values of N_F it will be possible to see traces of batch effects in the protein expressions. (ii) A large N_F decreases the number of identified proteins. (iii) The number of empty proteins directly increases with N_F and ρ_0. (iv) A very large value puts too much weight on the annotation that we now is not flawless. (v) When looking at the tree structures and classification performances with the ranges provided above the main structural features of Figures 7 and 6 persist. See Figure S5 as an example, where we show that the change in AUC using FHR1-H, CRP-H and LBP-H, for different values of N_F and ρ_0 is not significant wide around $N_F = 6$ and $\rho_0 = 1e3$.

5 DISCUSSION - CONCLUDING REMARKS

We have presented a factor model specifically designed for proteomics data anaa lysis. It successfully handles variability coming from broad scale variability that is known to come from several sources of technical variability such as batch effects and isotope group specific noise, hence enabling us to estimate latent protein profiles that better describe biological variability. Our hierarchical representation of isotope groups, latent proteins and parent proteins provide us with detailed annotation uncertainty assessment, detection of possibly miss annotated isotope groups or post-translational modified proteins and clustering of proteins with similar expression profiles that hopefully reflect biologically related interaction mechanisms. We also showed that features of our model can be used to define predictive models based either on latent proteins or groups of latent proteins.

Particular to the H1N1/H3N2 study case, we found that we can subtract the evident systematic and batch effects from the estimated latent protein expression profiles. Annotation-wise, we built a model that achieved 49% an 68% total and HxNx specific integrities, respectively. This is an indication that it is possible that some of the isotope groups are miss annotated and some of the proteins post-translational modified. The difference between specific and total identity, tell us that the approach used to align the datasets has room for improvement. We also found that 3% of the isotope groups are not stable in terms of protein assignments thus can be regarded as highly noisy or of poor quality. At a latent protein level, we encountered that 72% of them are representative of their annotation label, thus graded as *identified*. Finally, we found that a group of 5 proteins are discriminative of the symptomatic/asymptomatic status and that samples coming from H3N2 study are easier to classifier, particularly at latter stages of the disease.

The experiments in Section 4 omitted the fact that the data at hand is a time series. We did tried to incorporate such an information by providing the latent proteins with Gaussian process priors. In practice, this amounts to replace the covariance of the

Gaussian transition density of the tree from a diagonal matrix to a function encoding the time dependence of the samples. The result will be that samples of such a covariance function will have a block diagonal structure by noting that we can only have time correlations in groups of four samples corresponding to the four time points available per patient. Unfortunately we could not make any significant improvement compared to the results already obtained, thus not included here. We conjecture that this may be due to insufficiency of time points compared to the size of the problem or the time scale being large enough to render the time correlations just "too weak" to be detected by our model.

There are still some challenges and features that can be integrated into the model to increase its applicability, for instance:

Non-annotated isotope groups Including non annotated isotope groups in our current model has essentially two difficulties. The first is that if we run our model using all the data, we will accept that annotated and annotated isotope groups belong to the same set of proteins. This is unrealistic because such a set of proteins is taken entirely from the annotation. The second has to do with latent protein identification. Lets assume we can let the model decide upon the number of latent proteins, this will allow the model to accommodate proteins beyond the initial set created from the annotation. After inference, we will have to label latent proteins according to the concentration of its components, however this can be done only with annotated isotope groups for which we have protein assignments. Preliminary results using beta processes suggest that the model becomes harder to interpret because we tent to end up with latent proteins in which the concentration of annotated proteins is too low or too heterogeneous to be able to label them.

Introducing side information in the tree It is possible to affect the way in which latent proteins merge in the tree. One possibility is to introduce a conditioning observed variable y on each parent protein in the tree. Then we only need to replace $p(v_{s,j})$ with $p(v_{s,j}, y)$ so equations (3) and (4) become

$$\mu_{v_{s,j}\to v_j} = \int p(v_j|v_{s,j}, \Delta_{s,j})p(y|v_{s,k})p(v_{s,j})dv_{s,j} \ ,$$
$$p(v_j|\boldsymbol{\theta}) = \mu_{v_{l,j}\to v_j}\mu_{v_{l,j}\to v_j}\mu_{y\to v_j} \ ,$$

where $\mu_{y\to v_j} = p(y|v_{s,k})$ assuming that $p(y) = \delta_y$. This representation can be very convenient because depending on y, the so called parent proteins can be seen for example as regressors or predictors providing the tree with more possibilities from an interpretation point of view.

A MCMC INFERENCE DETAILS

We describe next the MCMC analysis mostly based on Gibbs sampling. We provide then the relevant conditional posteriors and SMC based updates details for the tree structure. To simplify notations, we use the following shorthands $\widetilde{\mathbf{X}} = [\mathbf{x}_1^m - \boldsymbol{\mu}^m \ \ldots \mathbf{x}_N^m - \boldsymbol{\mu}^m]$, $\mathbf{Z} = [\mathbf{z}_1 \ \ldots \mathbf{z}_N]$, $\mathbf{W} = [\mathbf{w}_1 \ \ldots \mathbf{w}_N]$, $\mathbf{X}_{i:}$ as the i-th row of $\mathbf{X}$.

Noise variance Sample each element of the diagonal of $\boldsymbol{\Psi}$ using

$$\psi_i^{-1}|s_s, s_r \ \sim \text{Gamma}\left(\psi_i^{-1}\left|s_s + \tfrac{N}{2}, s_r + c\right.\right) \ ,$$

where t_s and t_r are respectively prior shape and rate and

$$c = \tfrac{1}{2}(\widetilde{\mathbf{X}}_{i:} - \mathbf{A}_{i:}\mathbf{Z} - \mathbf{B}_{i:}\mathbf{W})(\widetilde{\mathbf{X}}_{i:} - \mathbf{A}_{i:}\mathbf{Z} - \mathbf{B}_{i:}\mathbf{W})^\top \ .$$

Batch means Sample mean vector for batch m from

$$\boldsymbol{\mu}^m|t_m, t_p \ \sim \ \mathcal{N}\left(\boldsymbol{\mu}^m\middle|\mathbf{C}\left(t_m t_p + \boldsymbol{\Psi}^{-1}\sum_{n\in B_m}\mathbf{x}_n - \mathbf{A}\mathbf{z}_n - \mathbf{B}\mathbf{w}_n\right), \mathbf{C}\right) \ ,$$

where $\mathbf{C} = (t_p + N_{B_m}\boldsymbol{\Psi}^{-1})^{-1}$, B_m is the set of observations in batch m, N_{B_m} in the size of B_m, t_m is the prior mean and t_p is the prior precision.

Systematic effect factors The conditional posterior of $\mathbf{Z}$ using scale mixtures of Gaussians representations can be computed independently for element of the matrix using

$$z_{ln}|\tau_{ln} \ \sim \ \mathcal{N}(z_{ln}|c_{ln}\mathbf{A}_{:l}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{\epsilon}_{\setminus ln}, c_{ln}) \ ,$$

where $c_{ln} = (\mathbf{A}_{:l}^\top\boldsymbol{\Psi}^{-1}\mathbf{A}_{:l} + \tau_{ln}^{-1})^{-1}$ and $\boldsymbol{\epsilon}_{\setminus ln} = \mathbf{x}_n - \mathbf{A}\mathbf{z}_n - \mathbf{B}\mathbf{w}_n - \boldsymbol{\mu}^m|_{z_{ln}=0}$. For a Laplace distribution, the mixing variances τ_{jn} are exponentially distributed $\tau_{jn} \sim \text{Exponential}(\tau_{jn}|\lambda^2)$, hence the resulting conditional is

$$\begin{aligned}
\tau_{ln}^{-1}|\lambda^2 \ &\sim \text{IG}\left(\upsilon_{ln}^{-1}\middle|\sqrt{\frac{\lambda^2}{z_{ln}}}, \lambda^2\right) \ , \\
\lambda^2|\ell_s, \ell_r \ &\sim \text{Gamma}\left(\lambda^2\middle|\ell_s + \tfrac{1}{2}, \ell_r + \tfrac{1}{2}\sum_{l,n}\tau_{ln}\right) \ ,
\end{aligned}$$

where ℓ_s and ℓ_r are shape and rate priors, respectively. $\text{IG}(\cdot|\mu, \lambda)$ is the inverse Gaussian distribution with mean μ and scale parameter λ (Chhikara and Folks, 1989). Finally, each element a_{il} from the factor loading matrix is sampled from

$$a_{il} \ \sim \mathcal{N}(a_{il}|c_{il}\boldsymbol{\epsilon}_{\setminus il}\mathbf{Z}_{l:}^\top, c_{il}\psi_i) \ ,$$

where $c_{il} = (\mathbf{Z}_{l:}\mathbf{Z}_{l:}^\top + \psi_i)^{-1}$ and $\boldsymbol{\epsilon}_{\setminus il} = \widetilde{\mathbf{X}}_{i:} - \mathbf{A}_{i:}\mathbf{Z} - \mathbf{B}_{i:}\mathbf{W}|_{a_{il}=0}$.

Protein profiles The conditional posterior for latent proteins w_k in equation (3) can be updated using

$$\mathbf{W}_{k:}|K_k \;\sim\; \mathcal{N}(\mathbf{W}_{k:}|c\mathbf{b}_k^\top \boldsymbol{\Psi}^{-1}(\widetilde{\mathbf{X}} - \mathbf{AZ}), c\mathbf{I}) \;,$$

where $c = (\mathbf{b}_k^\top \boldsymbol{\Psi}^{-1}\mathbf{b}_k + 1)^{-1}$. The coefficients $b_{ik} \neq 0$ only if $x_k \in K_k$ and we can sample them from

$$\mathbf{b}_k|K_k \;\sim\; \mathcal{N}(\mathbf{b}_k|\mathbf{C}(\widetilde{\mathbf{X}} - \mathbf{AZ})\mathbf{W}_{k:}^\top, \mathbf{C}\boldsymbol{\Psi}) \;,$$

where $\mathbf{C} = (\mathbf{W}_{k:}\mathbf{W}_{k:}^\top + \boldsymbol{\Psi})^{-1}$. Now we can sample the isotope group-latent protein assignments K_k using

$$\begin{aligned} u_i|\boldsymbol{\rho}_i, \boldsymbol{\kappa}, t_s, t_r \;&\sim \text{Discrete}(u_i|\mathbf{v}_i) \;, \\ v_{ki} \;&\propto (\rho_{ki} + \kappa_k)c^{-\frac{1}{2}} \left(t_s + \tfrac{1}{2}c^{-1}\widetilde{\mathbf{X}}_{i:}\mathbf{W}_{k:}^\top \mathbf{W}_{k:}\widetilde{\mathbf{X}}_{i:}^\top\right)^{-t_s - \frac{N}{2}} \;, \end{aligned}$$

where $c = \mathbf{W}_{k:}\mathbf{W}_{k:}^\top$, v_{ki} is element of $\mathbf{v}_i$ and $K_k = \{x_i | u_i = k\}$.

Starting from equations (7) and (8), we can obtain explicit expressions for the parent protein distributions in equations (3), (4) and (5) as

$$\mu_{v_{s,j} \to v_j} = \mathcal{N}(v_j|c_j\mathbf{m}_{s,j}, c_j\boldsymbol{\Phi}) \;, \quad p(v_j|\boldsymbol{\theta}) = \mathcal{N}(v_j|\mathbf{m}_j, s_j\boldsymbol{\Phi}) \;,$$

where $c_j = c_{s,j} + \Delta_{s,j}$, $s_j = (s_{l,j}^{-1} + s_{r,j}^{-1})^{-1}$, $\mathbf{m}_j = s_j(s_{l,j}^{-1}\mathbf{m}_{l,j} + s_{r,j}^{-1}\mathbf{m}_{r,j})$ and $p(v_{s,j}) = \mathcal{N}(v_{s,j}|\mathbf{m}_{s,j}, s_{s,j}\boldsymbol{\Phi})$. The marginal in equation (6) can be easily obtained from the second equation above. We still need to approximate the conditional posterior of the tree π. We use SMC with multinomial resampling (Doucet et al., 2001) during the leaves to root pass together with equations (5) and (9) to generate L tree configurations with weights $h_j^{(l)}$ sequentially updated as

$$h_j^{(l)} = h_{j-1}^{(l)} Z_{v_j|\boldsymbol{\theta}} \exp\left(-\tbinom{n-j+1}{2}\Delta_j\right) q(\Delta_j, v_{l,j}, v_{r,j}|\ldots)^{-1} \;, \tag{10}$$

where $q(\Delta_j, v_{l,j}, v_{r,j}|\ldots)$ is the proposal distribution for merging $v_{l,j}$ and $v_{r,j}$ at time Δ_j given the current state of the tree $(\ldots)$. As proposal distribution we set $q(\Delta_j, v_{l,j}, v_{r,j}|\ldots) = p(v_{l,j}, v_{r,j}|\Delta_j)p(\Delta_j)$, i.e. we draw Δ_j from its prior and the pair of variables to merge using their conditional posterior, thus equation (10) reduces to multiply the current weight $h_{j-1}^{(l)}$ by the sum of $Z_{v_j|\boldsymbol{\theta}}$ for every possible choice of $v_{l,j}$ and $v_{r,j}$. Note that inference for π depends entirely on marginals from equation (5) which only has $\boldsymbol{\Phi}$ as hyperparameter. We can avoid alternating between updates for π and $\boldsymbol{\Phi}$ by marginalizing the latter out, hence

$$Z_{v_j|\boldsymbol{\theta}} \propto \prod_{n=1}^{N} s_j^{-\frac{1}{2}} \left(t_r + \frac{1}{2}\frac{m_n^2}{s_j}\right)^{-t_s - \frac{1}{2}} \;,$$

where m_n is the n-th component of $\mathbf{m}_{l,j} - \mathbf{m}_{r,j}$, we used a gamma prior for $\boldsymbol{\Phi}$ with shape t_s and rate t_r, and $\boldsymbol{\theta}$ denotes the remaining parameters. After the leaves to root pass is done we can use the weights to sample a tree from the resulting configurations, i.e. $\pi = \text{Discrete}(\pi|h_j^{(1)}, \ldots, h_j^{(L)})$. To complete the procedure we just have to update the marginals from equation (6) with the tree structure π fixed.

Missing values The treatment of the missing values is rather simple, for each missing value x_{in}^m corresponding to isotope group i, sample n and batch m, we use independent standardized Gaussian distributions to exploit conjugacy.

Initialization We start from maximum likelihood estimates for the less critical quantities, i.e. batch means $\{\boldsymbol{\mu}^m\}_{m=1}^{N_B}$ and noise variances $\boldsymbol{\Psi}$. The systematic factors $\mathbf{Z}$ and latent proteins $\mathbf{W}$ are initialized using standardized Gaussian distributions. The loading matrices $\mathbf{A}$ and $\mathbf{B}$ (non-zero elements only) were set to ordinary least squares estimates based upon random initialization of $\mathbf{Z}$ and $\mathbf{W}$, respectively. The vector of associations $\mathbf{u}$ was provided with the information obtained from annotation about isotope group-protein assignments.

References

R. P. Adams, Z. Ghahramani, and M. I. Jordan. Tree-structured stick breaking for hierarchical data. In *Advances in Neural Information Processing Systems 24*. MIT Press, 2010.

D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society: Series B (Methodology)*, 36(1):99–102, 1974.

C. Blundell, Y. W. Teh, and K. A. Heller. Bayesian rose trees. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 2010.

C. M. Carvalho, J. Chang, J. E. Lucas, J. R. Nevins, Q. Wang, and M. West. High-dimensional sparse factor modeling: Applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484):1438–1456, 2008.

R. S. Chhikara and L. Folks. *The inverse Gaussian distribution: theory, methodology, and applications*. M. Dekker, New York, 1989.

T. Clough, M. Key, I. Ott, S. Ragg, G. Schadow, and O. Vitek. Protein quantitation in label-free LC-MS experiments. *Journal of Proteome Research*, 8:5275–5284, 2009.

D. S. Daly, K. K. Anderson, E. A. Panisko, S. O. Purvine, R. Fang, M. E. Monroe, and S. E. Baker. Mixed-effects statistical model for comparative LCMS proteomics studies. *Proteomics Research*, 7(3):1209–1217, 2008.

A. Doucet, N. de Freitas, N. Gordon, and A. Smith. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

K. A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the 22nd international conference on Machine learning*, pages 297–304. ACM, 2005.

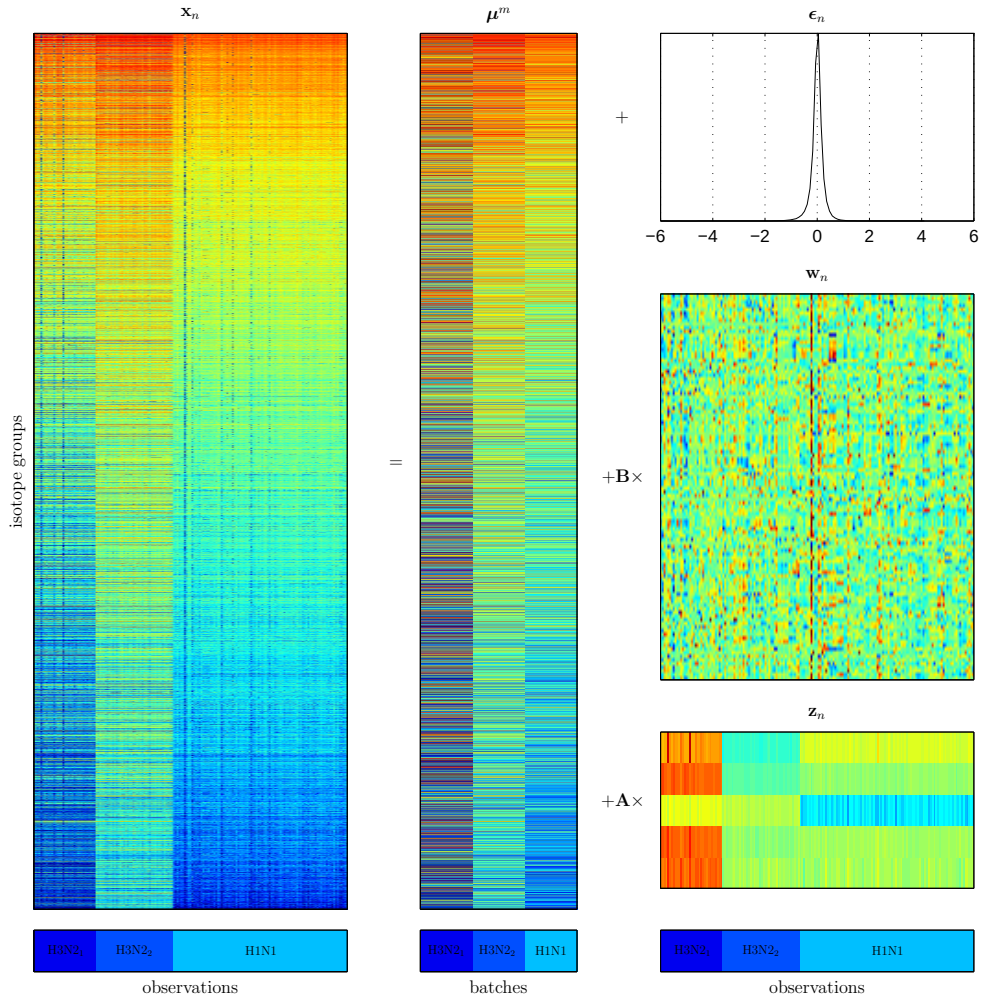


Figure 3: Data and model estimates as in equation (2). Samples are grouped into batches {H1N1,H3N2$_1$,H3N2$_2$}. Rows of $\mathbf{w}_n$ are latent proteins and rows of $\mathbf{z}_n$ are systematic factors. Missing values are black spots in $\mathbf{x}_n$.

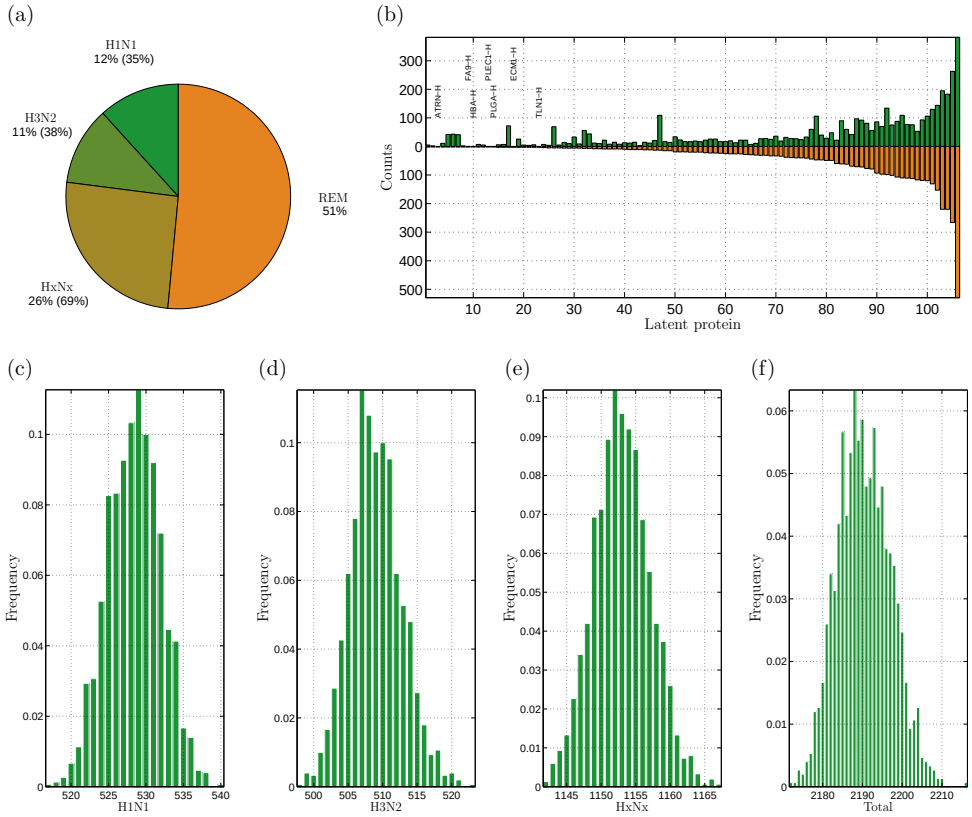


Figure 4: Integrities and protein counts. (a) Distribution of the 49% total integrity obtained after inference and subset specific integrities in parenthesis. (b) A-priori and a-posterior counts for latent proteins. Count is the number of IGs associated to a particular latent protein. The summary of **B** produced 7 empty proteins that are indicated with names as appear in annotation. (c), (d), (e) and (f) are posterior histograms of specific and total integrities.

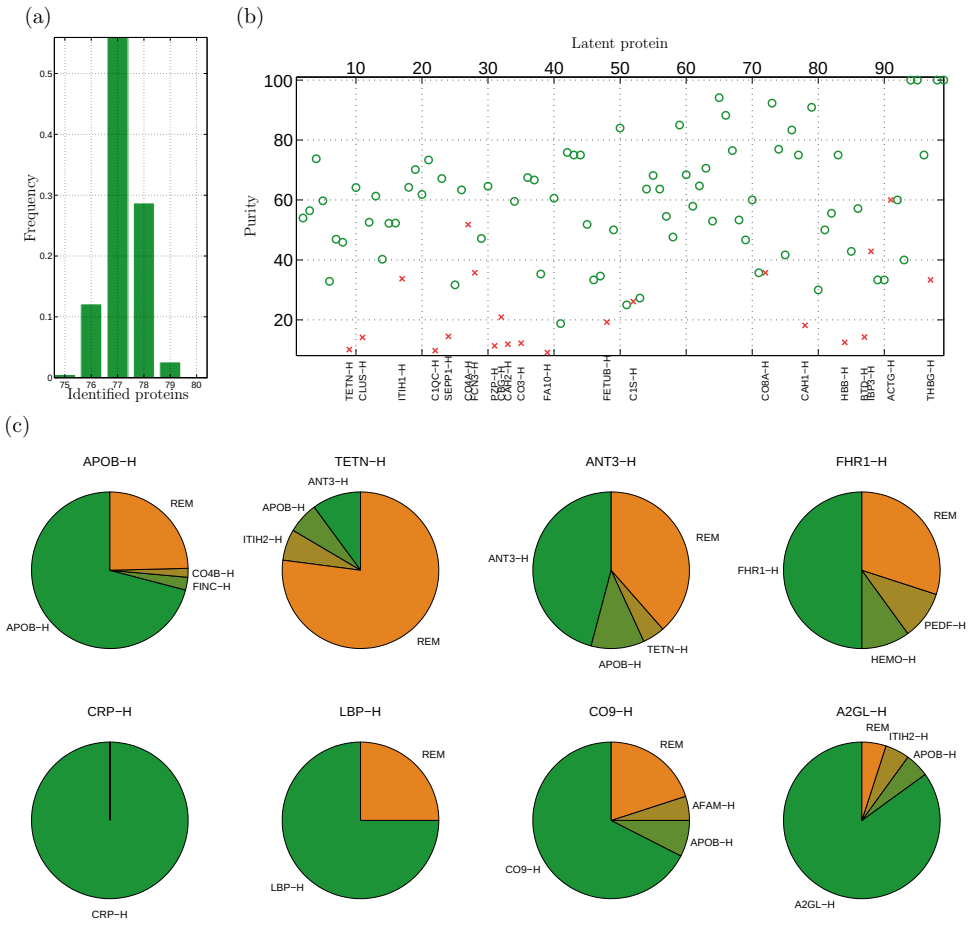


Figure 5: Identifications and latent protein compositions. (a) Posterior histogram of the number identified proteins. (b) In average 77 latent proteins out of 106 were identified (circles). 21 latent proteins were relabeled (crosses) due to disagreement between latent protein a-priori assigned label and its summary. A-priori names of the relabeled latent proteins are provided. (c) Selected latent protein compositions: APOB-H, the most abundant in annotation, TETN-H was relabeled as ANT-3 however ANT-3 has its own latent protein with high purity, FHR1-H, CRP-H, LBP-H, CO9-H, A2GL-H, the most discriminant proteins. REM means other proteins.

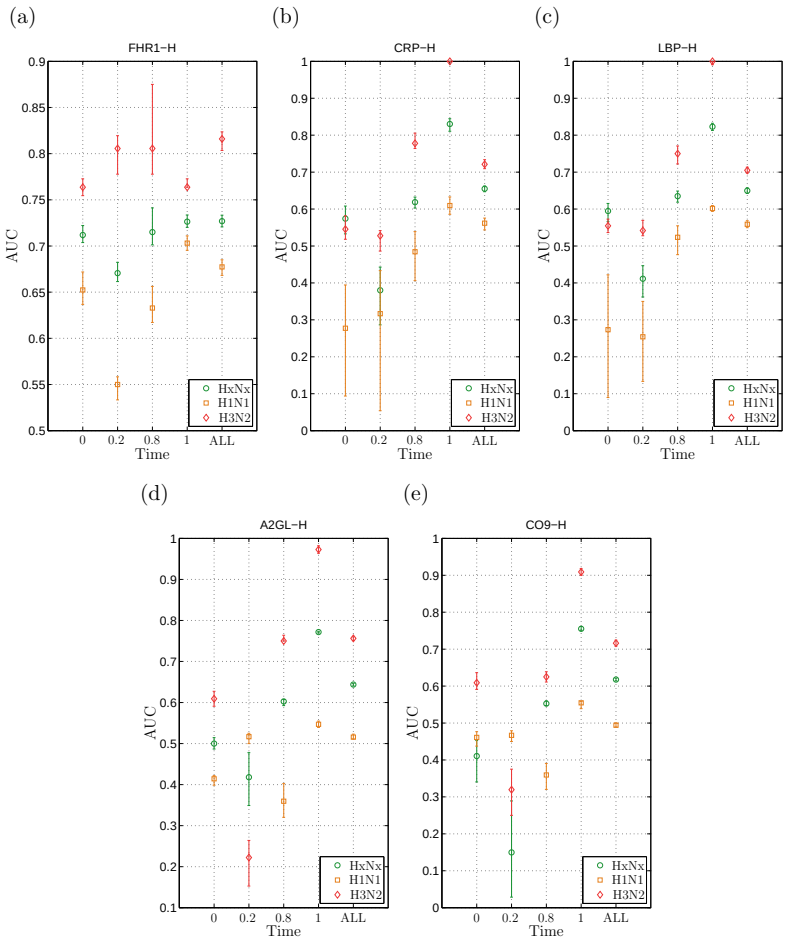


Figure 6: Discriminant latent proteins. The classification accuracy is presented as AUC values estimated using leave-one-out. The data is separated into different time points (x-axis) and studies, where HxNx means both studies. Markers indicate median values and error bars 50% credible intervals. overall, H3N2 is easier to classify than H1N1, and CRP-H and LBP-H are particularly good at classifying samples from time point $t = 1$.

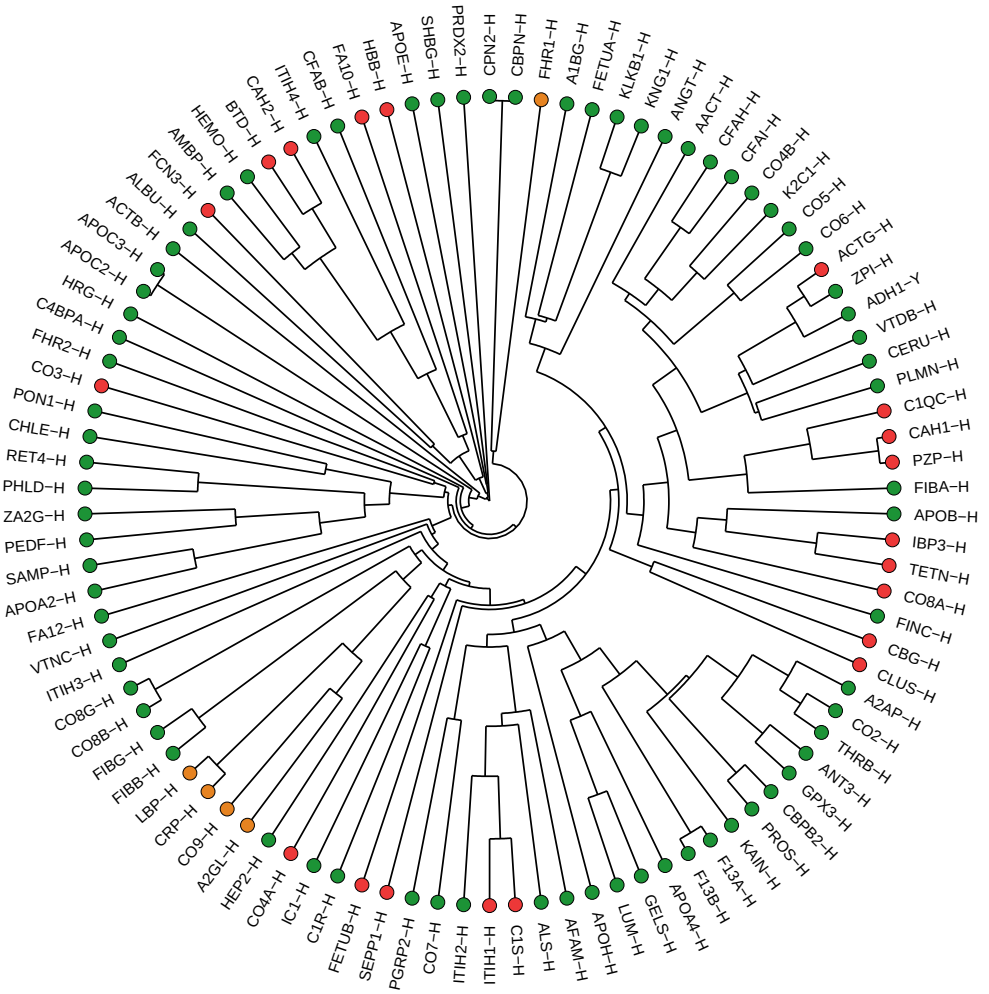


Figure 7: Latent protein tree. Latent proteins in orange are discriminant of the symptomatic/asymptomatic status of the data. Those in red were the ones relabeled after inference.

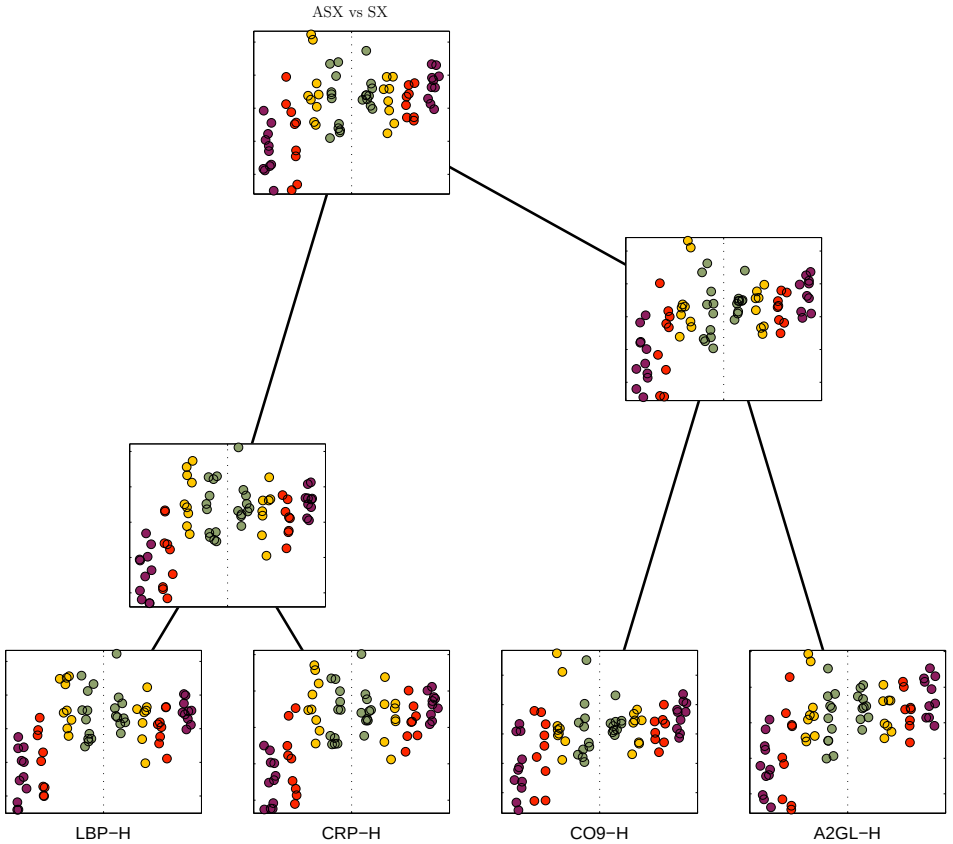


Figure 8: Discriminant subtree. Each node is represented as a scatter of the samples from study H3N2 only. The dotted line separates samples labeled asymptomatic (left) and symptomatic (right). The x-axis groups samples according to time: red for $t = 0$, yellow for $t = 0.2$, red for $t = 0.8$ and purple for $t = 1$. The y-axis is the estimated latent/parent protein expression. Note that time points $t = 0.8$ and $t = 1$ are nicely separated and the similarities/changes as the merge in the tree structure.

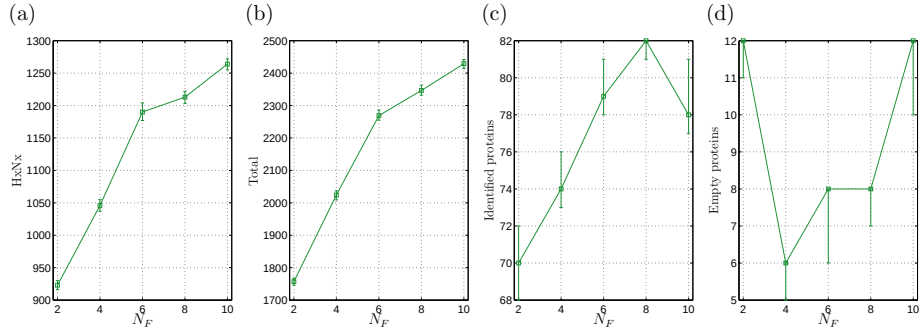


Figure 9: Behavior of N_F for different values (x-axis). In particular, (a) HxNx integrity, (b) total integrity, (c) number of identified latent proteins and (c) number of empty latent proteins. Error bars represent 95% credible intervals.

R. Henao and O. Winther. Sparse linear identifiable multivariate modeling. *Journal of Machine Learning Research*, To appear, 2011.

Y. V. Karpievitch, J. Stanley, T. Taverner, J. Huang, J. N. Adkins, C. Ansong, F. Heffron, T. O. Metz, W.-J. Qian, H. Yoon, R. D. Smith, and A. R. Dabney. A statistical framework for protein quantitation in bottom-up MS-based proteomics. *Bioinformatics*, 25:2028–2034, 2009.

A. Keller, A. I. Nesvizhskii, E. Kolker, and R. Aebersold. Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Analytica Chemistry*, 74(20):5384–5392, 2002.

J. F. C. Kingman. The coalescent. *Stochastic processes and their applications*, 13(3): 235–248, 1982a.

J. F. C. Kingman. On the genealogy of large populations. *Journal of Applied Probability*, 19:27–43, 1982b.

J. Lucas, C. Carvalho, Q. Wang, A. Bild, J. R. Nevins, and M. West. *Bayesian Inference for Gene Expression and Proteomics*, chapter Sparse Statistical Modeling in Gene Expression Genomics, pages 155–176. Cambridge University Press, 2006.

J. Lucas, C. Carvalho, and M. West. A Bayesian analysis strategy for cross-study translation of gene expression biomarkers. *Statistical Applications in Genetics and Molecular Biology*, 8(1):1–26, 2009.

J. E. Lucas, J. W. Thompson, L. G. Dubois, J. McCarthy, K. Patel, H. Tillman, A. Thompson, J. McHutchison, and M. A. Moseley. Metaprotein expression modeling for label-free quantitative proteomics. Technical report, Duke University, 2011.

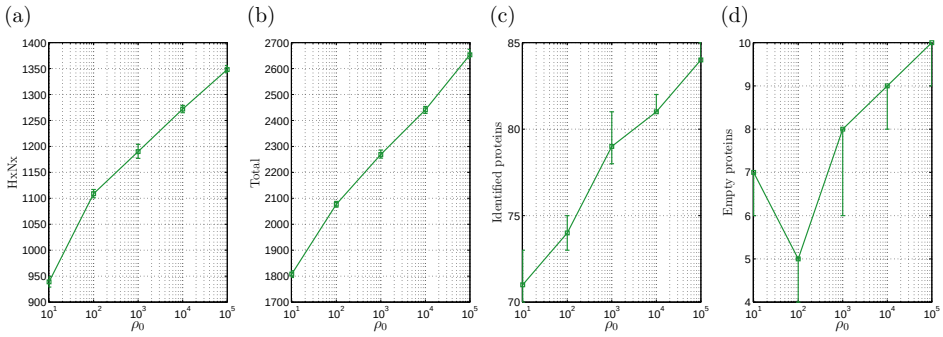


Figure 10: Behavior of ρ_0 for different values (x-axis). In particular, (a) HxNx integrity, (b) total integrity, (c) number of identified latent proteins and (c) number of empty latent proteins. Error bars represent 95% credible intervals.

R. Neal. Density modeling and clustering using Dirichlet diffusion trees. In J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, editors, *Bayesian Statistics 7*, pages 619–629. Oxford University Press, 2003.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

P. Rai and H. Daume III. The infinite hierarchical factor regression model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1321–1328. The MIT Press, 2009.

Y. W. Teh, H. Daume III, and D. Roy. Bayesian agglomerative clustering with coalescents. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1473–1480. MIT Press, 2008.

M. West. Bayesian factor regression models in the "large p, small n" paradigm. In J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, editors, *Bayesian Statistics 7*, pages 723–732. Oxford University Press, 2003.

A. K. Zaas, M. Chen, J. Varkey, T. Veldman, A. O. Hero, J. Lucas, Y. Huang, R. Turner, A. Gilbert, R. Lambkin-Williams, N. C. Øien, B. Nicholson, S. Kingsmore, L. Carin, C. W. Woods, and G. S. Ginsburg. Gene expression signatures diagnose influenza and other symptomatic respiratory viral infections in humans. *Cell*, 6(3):207–217, 2009.

SUPPLEMENTAL MATERIALS

S.1 Alignment of data sets

The data from a single sample consists of a list of features along with their associated mass-to-charge ratios and retention times. Because there is some level of randomness to all of these measurements, there is some uncertainty in the identification of which feature from one sample should be associated with a given feature in another sample. The process of matching features across samples is termed data alignment. For matching features within a single experiment, we utilize Rosetta Elucidator$^{\text{TM}}$, which is a commercial package for the processing and analysis of proteomics data. However, there were sufficient differences between the H3N2 and H1N1 experiments (which were run months apart) to make the Rosetta algorithm inadequate for the task of alignment across datasets. For data alignment across the different batches, we utilized the following construction.

Let i be an index over the set of all peptides measurable in our experiment. Further, define γ_i to be 1 if the ith peptide was measured in the experiment. Let x_i be a vector containing the "true" retention time and mass-to-charge ratio associated with the ith peptide. Then, if $\gamma_i = 1$, we assume our measured values, x_i^* are normally distributed around x_i with some shift and scale along with an unknown covariance, $x_i^* \sim N(\mu + \delta x_i, \Phi^{-1})$. There is a small subset of isotope groups that have been identified in both data sets (approximately 600). We initialize all of our parameters to maximize the likelihood of this small subset, then use a greedy algorithm to select matches for the remainder of the isotope groups. The algorithm stops assigning matches based on the prior probability that $\gamma_i = 1$ (we have used 0.5).

There is substantial information available that is not being used in this algorithm. First, it would be possible to assign prior distributions to the model parameters and iteratively fit this model. This would lead to better estimates of model parameters along with full posterior distributions. However, because the distribution is extremely spiky, the estimation of uncertainty from this algorithm is somewhat uninteresting and uninformative. Second, there is information available in the high energy mass spectrometry trace even when that trace is insufficient to fully identify the peptide which one might include in our model as additional dimensions to x_i. Third, we are not making use of the intensity of the measured isotope groups across the samples. This allows for the possibility that there may be drastic changes in peptide concentrations between the two experiments. Even so, it is likely possible to obtain and use reasonable and informative distributions on these intensities for the purposes of alignment. However, because all of our results are based on factors, which are the aggregate expression of multiple isotope groups, a low but non-zero level of inaccurate alignments may lead to a mild increase in noise, but not a drastic change in our overall results.

a-priori	a-posteriori	purity	a-priori	a-posteriori	Purity
APOB-H	APOB-H	70.94	CO4B-H	CO4B-H	53.99
CERU-H	CERU-H	56.41	FINC-H	FINC-H	73.77
CFAH-H	CFAH-H	59.72	THRB-H	THRB-H	32.84
HEMO-H	HEMO-H	46.92	ANT3-H	ANT3-H	45.87
TETN-H	ANT3-H	**10.09**	CFAB-H	CFAB-H	64.15
CLUS-H	A1BG-H	**14.15**	APOH-H	APOH-H	52.58
PLMN-H	PLMN-H	61.29	AACT-H	AACT-H	40.22
FETUA-H	FETUA-H	52.22	VTDB-H	VTDB-H	52.27
ITIH1-H	ITIH2-H	**33.72**	ANGT-H	ANGT-H	64.20
CO5-H	CO5-H	70.13	ITIH4-H	ITIH4-H	61.84
APOA4-H	APOA4-H	73.33	C1QC-H	THRB-H	**9.72**
A1BG-H	A1BG-H	67.14	SEPP1-H	HEMO-H	**14.49**
FIBG-H	FIBG-H	31.67	GELS-H	GELS-H	63.33
CO4A-H	CO4B-H	**51.79**	FCN3-H	APOB-H	**35.71**
ITIH2-H	ITIH2-H	47.17	PGRP2-H	PGRP2-H	64.58
PZP-H	APOB-H	**11.36**	CBG-H	CERU-H	**20.93**
CAH2-H	HEMO-H	**11.90**	KNG1-H	KNG1-H	59.52
CO3-H	CO4B-H	**12.20**	CO9-H	CO9-H	67.50
PEDF-H	PEDF-H	66.67	C1R-H	C1R-H	35.29
FA10-H	CO4B-H	**9.09**	HEP2-H	HEP2-H	60.61
FIBA-H	FIBA-H	18.75	AFAM-H	AFAM-H	75.86
ALS-H	ALS-H	75.00	IC1-H	IC1-H	75.00
A2AP-H	A2AP-H	51.85	FIBB-H	FIBB-H	33.33
CO8G-H	CO8G-H	34.62	FETUB-H	CERU-H	**19.23**
RET4-H	RET4-H	50.00	HRG-H	HRG-H	84.00
VTNC-H	VTNC-H	25.00	C1S-H	ITIH2-H	**26.09**
ALBU-H	ALBU-H	27.27	CFAI-H	CFAI-H	63.64
CO6-H	CO6-H	68.18	CO7-H	CO7-H	63.64
SAMP-H	SAMP-H	54.55	ITIH3-H	ITIH3-H	47.62
A2GL-H	A2GL-H	85.00	C4BPA-H	C4BPA-H	68.42
ZA2G-H	ZA2G-H	57.89	APOE-H	APOE-H	64.71
CO2-H	CO2-H	70.59	CO8B-H	CO8B-H	52.94
FA12-H	FA12-H	94.12	KAIN-H	KAIN-H	88.24
LUM-H	LUM-H	76.47	F13A-H	F13A-H	53.33
KLKB1-H	KLKB1-H	46.67	SHBG-H	SHBG-H	60.00
CBPN-H	CBPN-H	35.71	CO8A-H	CERU-H	**35.71**
ADH1-Y	ADH1-Y	92.31	PHLD-H	PHLD-H	76.92
ACTB-H	ACTB-H	41.67	CBPB2-H	CBPB2-H	83.33
PON1-H	PON1-H	75.00	CAH1-H	AACT-H	**18.18**
CPN2-H	CPN2-H	90.91	F13B-H	F13B-H	30.00
FHR1-H	FHR1-H	50.00	PROS-H	PROS-H	55.56
CHLE-H	CHLE-H	75.00	HBB-H	CO4B-H	**12.50**
AMBP-H	AMBP-H	42.86	APOC2-H	APOC2-H	57.14
BTD-H	ANT3-H	**14.29**	IBP3-H	APOA4-H	**42.86**
APOC3-H	APOC3-H	33.33	PRDX2-H	PRDX2-H	33.33
ACTG-H	FINC-H	**60.00**	FHR2-H	FHR2-H	60.00
K2C1-H	K2C1-H	40.00	ZPI-H	ZPI-H	100.00
GPX3-H	GPX3-H	100.00	LBP-H	LBP-H	75.00
THBG-H	FETUA-H	**33.33**	APOA2-H	APOA2-H	100.00
CRP-H	CRP-H	100.00			

Table S1: Protein Labeling summary. Bold purities indicate relabeled proteins.

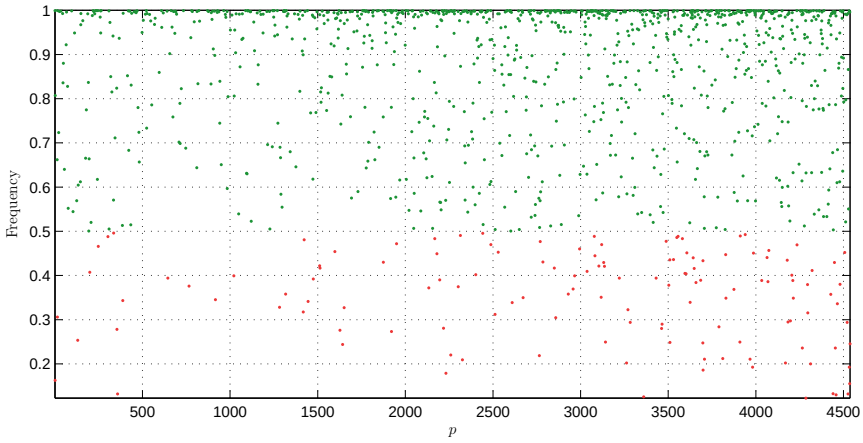


Figure S1: Summary of **u**. The y-axis represents the frequency of the mode of **u** for each isotope group. Small values indicate high variability in the IG-protein assignment, thus unstable/noisy IGs not suitable for interpretation. Dots in red indicate IG with frequencies less than 50% (154) that can be considered as unstable.

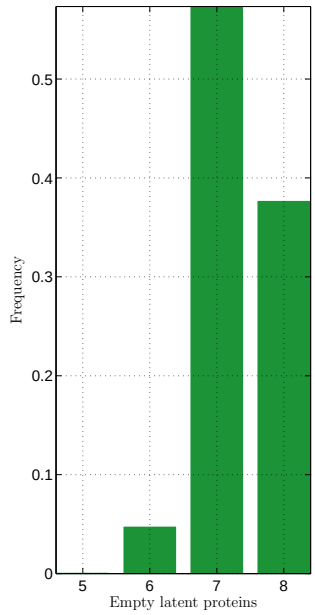


Figure S2: Posterior histogram of the number of empty proteins indicating that more than 50% of the posterior samples led to 7 empty proteins.

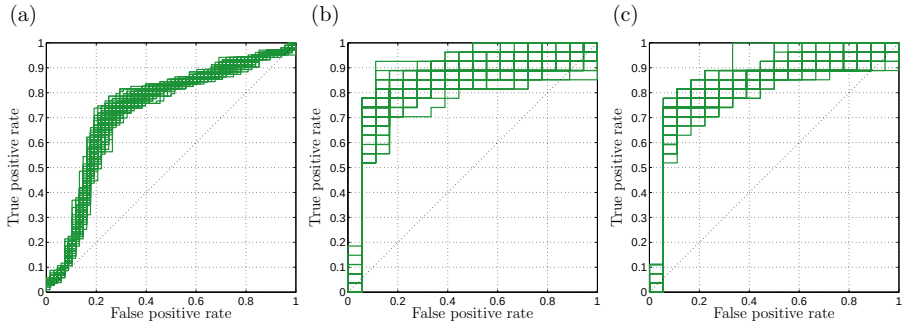


Figure S3: ROC curves for (a) FHR1-H, (b) CRP-H and (c) LBP-H. Each curve represents a posterior sample (3000) and the diagonal dotted line indicates a random classification performance.

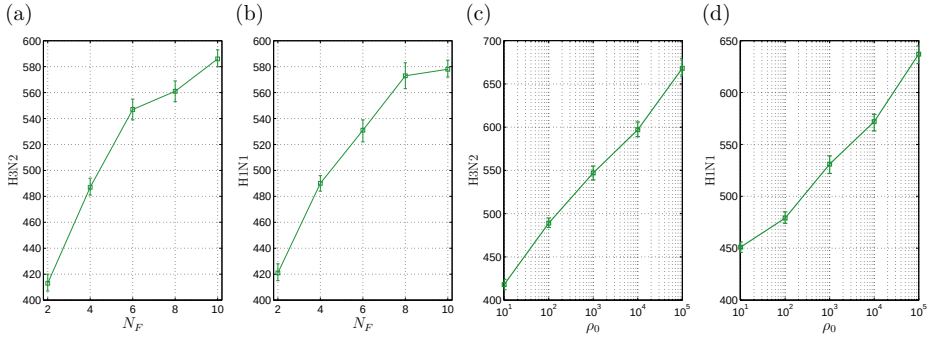


Figure S4: Specific integrities for different values of N_F in (a) and (b) and ρ_0 in (c) and (d). Error bars represent 95% posterior credible intervals.

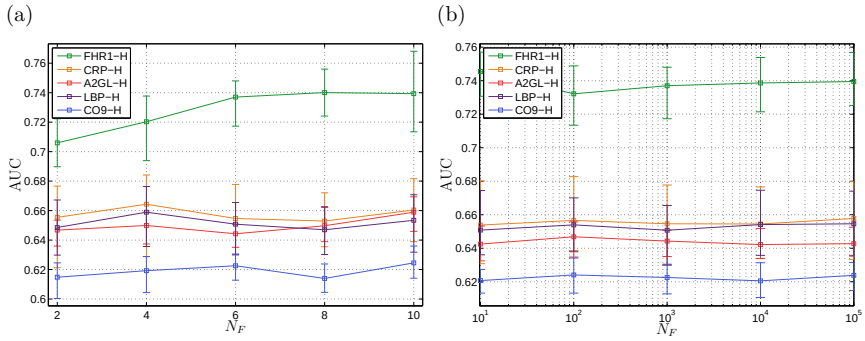


Figure S5: Discriminant latent proteins performance for different values of N_F in (a) and ρ_0 in (b). each curve is a different latent protein and the error bars indicate 50% posterior credible intervals.

Appendix F

Sparse Bayesian multi-category classification

Work in progress

Available from DTU Informatics at
`http://imm.dtu.dk/~rh/sbmc.pdf`

BIOINFORMATICS

Vol. 00 no. 00 2011
Pages 1–9

Sparse Bayesian multi-category classification: Multivariate probe selection for gene expression data

Ricardo Henao [A,B]*, Bogumil Kaczkowski [A] and Ole Winther [A,B]

[A]Bioinformatics Centre, University of Copenhagen, 2200 Copenhagen, Denmark
[B]DTU Informatics, Technical University of Denmark, 2800 Lyngby, Denmark

Received on XXXX; revised on XXXX; accepted on XXXX

Associate Editor:

ABSTRACT

Motivation: Classification for the extreme ill-posed case of many more covariates than examples is still an open question despite much recent research within machine learning and statistics. Robust and high performance classifiers for this scenario is a key ingredient in diagnosis based upon gene expression profiling. Current practice often involves using (single gene) univariate tests as a feature extraction step prior to classification. This is in general suboptimal as it can miss important features for instance when linear combinations of inputs are required to get discrimination.

Results: In this contribution we propose a sparse hierarchical Bayesian multi-category linear classifier especially well-suited for the more covariates than examples scenario arising prominently in classification of gene expression profiles. The prior over parameters is the so-called slab and spike prior: a two-component mixture of a point-mass at zero (spike) and continuous (slab) which expresses that a priori there is a large probability that the contribution of a covariate is zero. We use Gibbs sampling for inference and demonstrate empirically that the model is robust, easy to set up and viable even for large problems in the expression profiling scenario. We consider up to 5.000 covariates and 1.500 samples. We outperform state-of-the-art linear classification methods when they are used with univariate tests for probe selection. Our results are at least as good as state-of-the-art non-linear classification methods with and without covariate selection preprocessing. Our findings indicate that the success of our approach can be attributed to the fact that we use an explicit model for sparsity and get accurate assessment of distribution of the parameters. The strong covariation between genes seen in expression profiles implies that many possible subsets of genes may solve the discriminatory task. This effect seriously affects normal classification strategies ability to correctly assess the importance of individual genes. We thus also observe that genes that are found to be important by the Bayesian approach are very different from those extracted by univariate tests.

Availability: Software (in R and C) and data sets are available from `http://www.binf.ku.dk/sbmc`.

Contact: rhenao,bok,winther@binf.ku.dk

1 INTRODUCTION

Classification for the extreme ill-posed case of many more covariates than examples, sometimes referred to as "large p, small n" (West, 2003), has attracted much recent attention in bioinformatics especially in the context of gene expression profiling (Statnikov et al., 2005) . Current practice often involves a combination of supervised and unsupervised single covariate filtering prior to classification. Genes with a deviation of the intensity small or large relative to the mean may be excluded and univariate t- and F-tests may be used to further reduce the number of input features (Dudoit and van der Laan, 2008). Using univariate supervised techniques, for example t- and F-tests, may in general suboptimal as it can miss important features when the separating of the classes is not aligned with expression of the single genes by rather linear combinations of the genes. Also, when there is co-variation between the genes then univariate tests give a misleading picture of the significance of the genes. The motivation for working with a reduced set is both computational and predictive, i.e. working directly with a high number of non-informative covariates increases the risk for overfitting for most standard classifiers. Therefore, two step covariate selection/classification procedures with nested validation loops, e.g. using leave-one-out (LOO) cross-validation, must be used in order to obtain reliable unbiased results (Statnikov et al., 2005).

Classifiers with built-in multi-category capabilities and probability estimates of the output category labels are highly desirable in bioinformatics applications. It has been shown empirically that support vector machines (SVMs) have a better performance when used with an *ad-hoc* multi-category approach known as *one vs. rest* (Hsu and Lin, 2002, Statnikov et al., 2005), in which C different binary classifiers are trained by grouping the observations for each of the C categories in turn. The drawback of this approach is usually that the covariate selection procedure is performed on the multi-category setting (using F-tests for example) even when the classifier might not be multi-category by itself. Another possibility will be to select relevant covariates in an *one vs. the rest* fashion, which as a result will make the summarization and interpretation of the results rather convoluted. The second desirable property, probabilistic outputs, is readily available in model-based classifiers like Linear Discriminant Analysis (LDA) or Gaussian Process (GP) classification or can also be emulated in for instance SVMs (Platt, 1999). Classifiers with probabilistic outputs add a new layer of interpretability to the results, not only because is a way of measure uncertainty but also because allows us to evaluate relatedness between categories and the possibility of reject or reevaluate predictions with high uncertainty levels.

In this paper we propose a fully Bayesian approach to the problem of multi-category classification for the ill-posed setting. The key ingredient we want to capture in the model is sparsity, i.e. that

*To whom correspondence should be addressed.

only a small fraction of the genes (probes) can be expected to give discriminatory information. We explicitly model this by having a prior over the weights that put a large point probability mass at zero. The prior distribution over the weights is therefore a so-called two layer slab (continuous) and spike (δ-function at zero) mixture (West, 2003, Lucas et al., 2006). We limit ourselves to linear models because inference is much easier because of standard Gibbs sampling can be used. The straightforward extension to non-linearity can be done using GPs (sparse but binary, classifier Liang et al., 2005-09) or Dirichlet process mixtures (multi-category but not sparse, Shahbaba and Neal, 2009). However, the need for Metropolis-Hastings sampling of the GP's covariance function parameters (with spike and slab prior) makes inference much more complicated in this case. In general, the model complexity should match the relative size of the training set so one can expect that often a linear model is the right choice for the ill-posed scenario. In fact, Statnikov et al. (2005) observed this empirically through a series of benchmark experiments.

Some related work include sparse multinomial logistic regression using Laplace distributions to achieve sparsity as in the models introduced by Krishnapuram et al. (2005) and Cawley et al. (2007). Since the only way to obtain sparsity using Laplace distributions without using thresholding is to perform Maximum a-posteriori (MAP) inference. In the latter, it is hard to asses the uncertainty of the selected variables or ranking them using a criterion other than the magnitude associated to the their weights. Girolami and Rogers (2006) proposed a multi-category GP classifier in which variable selection is possible through Automatic Relevance Determination (ARD) covariance functions and thresholding. Lastly, Claeskens et al. (2008) introduce a new information criterion for variable selection using SVMs and also offers a review of existing methods in the same spirit. These approaches are also point estimates as MAP so they have the same issues associated to uncertainty assessment. In addition, similar to the other parameters in the SVM, the level of sparsity must be selected by cross-validation, making the computational cost an issue and the interpretation more difficult.

Related fully Bayesian approaches for the binary classification scenario include Bae and Mallick (2004) using Laplace priors. Lee et al. (2003), Zhou et al. (2004a) and Hernandez-Lobato et al. (2010) employ one layer slab and spike priors as defined by George and McCulloch (1993, 1997). For multi-category classification, we are aware of two approaches also using one layer spike and slab priors (Zhou et al., 2004b, Sha et al., 2004). The main difference between the latter two and our approach is the more elaborated sparsity prior and that we compute proper predictive distributions to obtain fully probabilistic outputs.

The rest of the paper is organized as follows. In Sections 2 and 3, we describe the model and inference procedure in detail. Section 4 is the most important for practitioners. In this section we outline how to use our model and perform extensive benchmark comparisons on both artificial and real gene expression profiling data. Finally, we conclude with a discussion in Section 5.

2 SPARSE BAYESIAN CLASSIFICATION

Suppose we count with a set of N independent observations $\mathbf{x}_1,\ldots,\mathbf{x}_n,\ldots,\mathbf{x}_N$, where $\mathbf{x}_n$ is a vector with d covariates assigned to one of C different classes. The class labels y_n are assumed to have a discrete distributions with parameters $p_{1n},\ldots,p_{cn},\ldots,p_{Cn}$. We define stochastic regression functions $f_{cn} = f_c(\mathbf{x}_n)$ for all classes $c = 1,\ldots,C$ and observations $n = 1,\ldots,N$ available, similar to Albert and Chib (1993). Here we will focus on model on the form $f_c(\mathbf{x}_n) = h_c(\mathbf{x}_n) + \varepsilon_{cn}$. The independently distributed additive term ε_{cn} determines the link function of the model. For instance, a zero mean-unit variance Gaussian leads to the probit link, however we will also consider a more general class of Student's t-distributed links. Effective Gibbs sampling inference can be implemented for a linear model $h_c(\mathbf{x}_n) = \mathbf{w}_c^\top \mathbf{x}_n$, where $\mathbf{w}_c$ is the weights vector for class c. Although, in principle this function could be made non-linear by using for examples Gaussian process priors (see Girolami and Rogers, 2006, Liang et al., 2005-09).

2.1 Multi-class likelihood

In classification, the likelihood function links the regression model with the probabilistic model for the output labels. The simplest model assigns probability one to class c when f_{cn} is larger than the other f_{jn} for $j \neq c$, so

$$p_{cn} = p(y_n = c|\mathbf{f}_n) = \prod_{j\neq c} \Theta(f_{cn} - f_{jn}) \,,$$

where we have used $\mathbf{f}_n = [f_1,\ldots,f_C]^\top$ and $\Theta(\cdot)$ is the Heaviside step-function. From the expression above we can see directly only the difference between regression functions is identifiable for the model. This will in some cases play a role for the interpretation of the inferred model parameters (see first artificial data example in Section 4). As shown by Albert and Chib (1993), when $p_{\text{link}}(\varepsilon_{cn}|\cdot)$ is Gaussian, i.e. $\varepsilon_{cn} \sim \mathcal{N}(\varepsilon_{cn}|0,1)$, we may marginalize over f_{cn}, $\mathbf{f} \sim \mathcal{N}(\mathbf{f}|\mathbf{h},\mathbf{I})$, to obtain a soft decision boundary model expressed in terms of $\mathbf{h} = [h_1(\mathbf{x}),\ldots,h_C(\mathbf{x})]^\top$ and the parameters of the link distribution as

$$\begin{aligned} p(y=c|\mathbf{h},\cdot) = &\int \prod_{j\neq c} \Theta(f_c - f_j) \prod_k p_{\text{link}}(f_k|h_k(\mathbf{x}),\cdot) d\mathbf{f} \\ &\int \prod_{j\neq c} \Phi(f_c - h_c(\mathbf{x})) \mathcal{N}(f_c|h_c(\mathbf{x}),1) d f_c \,, \end{aligned} \quad (1)$$

where $\Phi(\cdot)$ is the probit link (cumulative Gaussian) function. For the particular case of $C = 2$, this formulation reduces to standard probit regression since $p(y_n = 1|h(\mathbf{x}_n)) = \Phi(h(\mathbf{x}))$ with $h(\mathbf{x}) = (h_1(\mathbf{x}) - h_2(\mathbf{x}))/\sqrt{2}$ and in the linear model $h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ and $\mathbf{w} = (\mathbf{w}_1 - \mathbf{w}_2)/\sqrt{2}$.

2.2 Beyond probit link

We can use infinite scale mixture of Gaussian representations (Andrews and Mallows, 1974) to easily provide ε_{cn} with a more general class of distributions, namely the Student's t family, while still being able to sample from the posterior distribution in an efficient way. We only have to note that

$$t(\varepsilon_{cn}|\sigma^2,\theta) = \int \mathcal{N}(\varepsilon_{cn}|0,\psi_{cn}\sigma^2) \text{Gamma}\left(\psi_{cn}^{-1}|\tfrac{\theta}{2},\tfrac{\theta}{2}\right) d\psi_{cn} \,, \quad (2)$$

where σ^2 is the variance and θ is the degrees of freedom. For example, the t link contains as special cases, probit when $\sigma^2 = 1$ and $\theta \to \infty$, Cauchy when $\sigma^2 = \theta = 1$, and it is known to be a

good approximation of the logit link when $\sigma^2 = 0.401$ and $\theta = 8$ (Albert and Chib, 1993) or similarly $\sigma^2 = 0.413$ and $\theta = 7.581$ obtained in (Chen and Dey, 1998).

2.3 Sparse parameters with slab and spike prior

The use of sparse models is based on the assumption that the data contains irrelevant covariates, i.e. there is a number of covariates $d' < d$ such that the model with d' (relevant) covariates is at least equally supported by the data as the full model. This is specially true for instance in gene expression classification where the expected number of genes involved in a particular condition (class) is known to be small compared to the size of a commercial microarray. Sparsity may also be motivated by the need to control the complexity in the data poor regime $N \ll d$, i.e. we simply have too little data to learn the model in all its complexity but we can learn at least some features of it from limited data. The ideal complexity of a model is thus closely related with the number of observations used to fit its parameters, which in principle means that we cannot expect to use more covariates than observations if we want to prevent overfitting, unless we regularize the model in some way. One possibility for restricting the model is to turn off some of its parameters, which in the linear case correspond to variable selection.

Here we use a prior distribution over the weight matrix that is able to perform variable selection and produce interpretable results. We construct a hierarchical slab and spike prior in the following way (Lempers, 1971, Mitchell and Beauchamp, 1988, West, 2003, Lucas et al., 2006), (i) we assume that the weight w_{ci} corresponding to the c-th class and the i-th covariate is non-zero with probability η_{ci}. (ii) *A-priori*, η_{ci} has a probability $1-\nu_c$ of being zero and non-zero elements are drawn from a beta distribution with mean α_m and precision α_p. (iii) Finally, ν_c is specific to class c so each class has it own overall sparsity level. This means that an element w_{ci} can be turned off in the model even if the remaining $w_{c'i}$, $c' \neq c$ support it. As a result of (ii) and (iii), η_{ci} has in general a bimodal distribution with one of its modes at zero and the other at $\alpha_m > 0$, making it more informative and easy to interpret than a "one-level model" with η_{ci} being beta distributed with shared parameters. In that case, there is a higher risk of getting hard to interpret results with distributions for η_{ci} too spread over the unit interval. See Lucas et al. (2006) for a thorough discussion on one- and two-level slab and spike sparsity priors. In equations we can write

$$\begin{aligned} w_{ci}|r_{ci},\tau_{ci} &\sim (1-r_{ci})\delta(w_{ci}) + r_{ci}\,p(w_{ci}|\tau_{ci}) \;, \\ r_{ci}|\eta_{ci} &\sim \text{Bernoulli}(r_{ci}|\eta_{ci}) \;, \\ \eta_{ci}|q_{ci},\alpha_v,\alpha_m &\sim (1-q_{ci})\delta(\eta_{ci}) + q_{ci}\text{Beta}(\eta_{ci}|\alpha_v\alpha_m,\alpha_v(1-\alpha_m)) \\ q_{ci}|\nu_c &\sim \text{Bernoulli}(q_{ci}|\nu_c) \;, \\ \nu_c|\beta_m,\beta_v &\sim \text{Beta}(\nu_c|\beta_v\beta_m,\beta_v(1-\beta_m)) \;, \end{aligned} \tag{3}$$

where $p(w_{ci}|\cdot)$ is the distribution of those elements w_{ci} that turn out to be non-zero. Here we assume a Gaussian distribution with variance $\tau_{ci} \sim \text{Gamma}(\tau_{ci}|t_s,t_r)$ with shape t_s and rate t_r. The precision parameter α_p reflects the relative uncertainty in the choice of the mean probability α_m. Besides, the column $w_{1i},\ldots,w_{Ci}$ can be switched off with mean probability β_m and precision β_p. Prior specification is completed by assigning values to all hyperparameters, specifically we want *non-zero weights* to be non-zero with high probability and relatively large variance. This is done to obtain a more clear the separation between zero/non-zero weights as expressed by their distribution of being non-zero through η_{ci}, see Figure 2(c) for an example of this. For the shared parameter ν_c we want to promote highly sparse models with high precision. Parameters β_m and β_p are thus mainly set to reduce the false discovery rate. For the magnitude of non-zero weights we use a rather diffuse prior, say $t_s,t_r = \{2, 0.02\}$.

2.4 Making predictions

To make predictions we need to compute a predictive distribution of the form

$$p(\mathbf{y}^\star = c|\mathbf{X}^\star,\mathbf{y},\mathbf{X},\cdot) = \int p(\mathbf{y}^\star = c|\mathbf{f}^\star)p(\mathbf{f}^\star|\mathbf{W},\mathbf{U},\mathbf{X}^\star)p(\mathbf{W},\mathbf{U}|\mathbf{y},\mathbf{X})d\mathbf{f}^\star d\mathbf{W} d\mathbf{U} \;, \tag{4}$$

where $\mathbf{X} = [\mathbf{x}_1,\ldots,\mathbf{x}_N]$ and $\mathbf{y} = [y_1,\ldots,y_N]$ are respectively observations and labels used during inference, $\mathbf{X}^\star$ is a new set of observations to test and $\mathbf{y}^\star$ are their predicted classes. The distribution involved in equation (4) are, the posterior of the weight matrix $\mathbf{W} = [\mathbf{w}_1,\ldots.\mathbf{w}_C]^\top$, the posterior of $\mathbf{f}^\star$ in equation (2), the variances of the link function $\mathbf{U} = \{\psi_{cn}\}$ and the link function itself. It turns out that even when the integral above is intractable, is not difficult to sample from this posterior distribution as we will show in the next section.

3 INFERENCE

Bayesian analysis is performed using Markov chain Monte Carlo (MCMC) to produce samples from the posterior of all the parameters of the model, namely $\mathbf{W}$, $\mathbf{U}$, $\{\eta_{ci}\}$ and $\mathbf{v} = [\nu_1,\ldots,\nu_d]$. The most important summaries involve posterior samples from $p(w_{ci}|w_{ci} \neq 0,\mathbf{y},\mathbf{X})$, $p(w_{ci} \neq 0|\mathbf{y},\mathbf{X})$ and $p(\mathbf{y}^\star = c|,\mathbf{X}^\star,\mathbf{y},\mathbf{X})$. MCMC analysis is implemented using Gibbs sampling, that amounts to sequentially draw samples from *conditional posterior distributions*. For improved mixing of the Markov chain we also use collapsed Gibbs sampling where applicable. The latter consists in sampling from conditionals where a subset of the parameters have been marginalized out. We list these distributions in the remainder of the section. The graphical model in Figure 1 is provided to show all the dependencies between variables. Obtaining the conditional distribution for a specific variables amounts to collect all terms that either point to (prior) and out of (likelihood) the variable of interest and then normalizing. We use the following shorthands, $\mathbf{\Psi}_n$ is a diagonal matrix with entries $\{\psi_{1n},\ldots,\psi_{Cn}\}$, $\mathbf{\Psi}_c$ is a diagonal matrix with entries $\{\psi_{c1},\ldots,\psi_{cN}\}$, $\mathbf{X}_{i:}$ and $\mathbf{X}_{:n}$ are rows and column of $\mathbf{X}$, respectively.

1. Sample each non-zero weight w_{ci} from its conditional posterior $p(w_{ci}|\cdot) \propto p(\mathbf{F}|\mathbf{W},\mathbf{\Psi}_c)p(w_{ci}|\tau_{ci})$,

$$w_{ci}|\mathbf{F}_{c:},\mathbf{W}_{\setminus ci},\mathbf{X}_{i:},\tau_{ci},\mathbf{\Psi}_c \;\sim\; \mathcal{N}(w_{ci}|v\boldsymbol{\varepsilon}_{\setminus ci}\mathbf{\Psi}_c^{-1}\mathbf{X}_{i:}^\top, v) \;,$$

where

$$v_c = (\mathbf{X}_{i:}\mathbf{\Psi}_c^{-1}\mathbf{X}_{i:}^\top + \tau_{ci}^{-1})^{-1} \;,$$

and $\boldsymbol{\varepsilon}_{\setminus ci} = \mathbf{F}_{c:} - \mathbf{w}_c\mathbf{X}|_{w_{ci}=0}$, $\mathbf{F} = [\mathbf{f}_1,\ldots,\mathbf{f}_N]$ and only if $r_{ci} = 1$, otherwise set $w_{ci} = 0$.

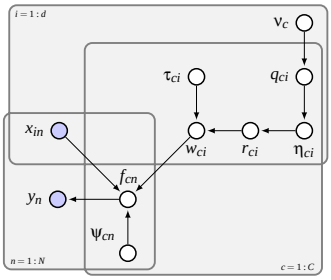


Fig. 1. Graphical model for the multi-category classifier. There are two observed nodes $\mathbf{X}$ and $\mathbf{y}$ corresponding to the covariates and class labels, respectively. The latent variables in $\mathbf{F}$ are connected to the variances $\mathbf{U}$ of the t link and the hierarchical prior for $\mathbf{W}$. The spike and slab prior consists on a binary variable r_{ci} indicating whether w_{ci} is non-zero with mean probability η_{ci} and a shared sparsity parameter ν_c with indicator variable q_{ci}. The non-zero elements in $\mathbf{W}$ are Gaussian distributed with variance τ_{ci}.

2. Sample the latent variables f_{cn} independently from $p(\mathbf{f}_n|\cdot) \propto p(y_n|\mathbf{f}_n)p(\mathbf{f}_n|\mathbf{w}_c, \mathbf{x}_n, \psi_{cn})$. This step factorizes over samples and it is important to distinguish $f_{y_n n}$ from f_{cn} for $c \neq y_n$. In both cases we have to sample from truncated Gaussians. We define $\mathcal{TN}(\cdot|\mu, \sigma^2, a, b)$ to be $\mathcal{N}(\cdot|\mu, \sigma^2)$ with mean μ, variance σ^2 and truncated within the interval (a, b), where $-\infty \leq a < b \leq \infty$. For $c \neq y_n$ we must sample from a truncated Gaussian below $f_{y_n n}$

$$f_{cn}|f_{y_n n}, \mathbf{w}_c, \mathbf{x}_n, \psi_{cn} \sim \mathcal{TN}(f_{cn}|\mathbf{w}_c\mathbf{x}_n, \psi_{cn}, -\infty, f_{y_n n}) ,$$

and for $f_{y_n n}$, i.e. $c = y_n$ we sample from a truncated Gaussian above the largest element of $\mathbf{f}_n$ without $f_{y_n n}$

$$f_{y_n n}|\mathbf{f}_{n\setminus y_n}, \mathbf{w}_c, \mathbf{x}_n, \psi_{cn} \sim \mathcal{TN}(f_{y_n n}|\mathbf{w}_c\mathbf{x}_n, \psi_{cn}, \max \mathbf{f}_{n\setminus y_n}, \infty) ,$$

where $\mathbf{f}_{n\setminus y_n}$ is the vector $\mathbf{f}$ for observation n without $f_{y_n n}$.

3. Sample the variances of $\varepsilon_{cn} = \mathbf{f}_{cn} - \mathbf{w}_c\mathbf{x}_n$ from its conditionals $p(\psi_{cn}^{-1}|\cdot) \propto p(\varepsilon_{cn}|0, \psi_{cn}\sigma^2)p(\psi_{cn}^{-1}|\frac{\theta}{2}, \frac{\theta}{2})$, which corresponds to a gamma distribution with shape $\frac{\theta}{2} + \frac{1}{2}$ and rate $\frac{\theta}{2} + \frac{1}{2\sigma^2}\varepsilon_{cn}^2$.

4. Sample from the degrees of freedom θ of the t link from $p(\theta|\cdot) \propto p(\mathbf{U}^{-1}|\theta)p(\theta|\cdot)$ using

$$\theta|\boldsymbol{\Psi} \sim p(\theta)\prod_{c,n} c(\theta)\psi_{cn}^{-\left(\frac{\theta}{2}-1\right)} \exp\left(-\frac{\theta\psi_{cn}^{-1}}{2}\right) ,$$

where $p(\theta)$ is the prior distribution (uniform in this case) for θ and $c(\theta)$ is the posterior normalizing constant. Since we are only interested on a finite set of θ, drawing samples from the posterior requires computing the probabilities over this set. We consider $\theta \in \{1, 2, 4, 8, \infty\}$, which roughly interpolates between extremely heavy-tailed (Cauchy-link) over logit to probit.

5. Sample from the spike and slab hierarchy in equation (3). We sample the binary variables, r_{ci} and q_{ci}, and their mean parameters, η_{ci} and ν_c, as follows. For r_{ci} we collapse η_{ci} and q_{ci} to get

$$r_{ci}|\mathbf{F}_{c:}, \mathbf{W}_{\setminus ci}, \mathbf{X}_{i:}, \tau_{ci}, \boldsymbol{\Psi}_c, \alpha_m, \nu_c \sim \text{Bernoulli}\left(r_{ci}\middle|\frac{\xi_{ci}}{1+\xi_{ci}}\right) ,$$

where

$$\xi_{ci} = \frac{\alpha_m\nu_c}{1-\alpha_m\nu_c}v_c^{1/2}\exp\left(\frac{(\boldsymbol{\varepsilon}_{\setminus ci}\boldsymbol{\Psi}_c^{-1}\mathbf{X}_{i:}^\top)^2}{2v_c^{-1}}\right) .$$

For η_{ci} we sample from

$$p(\eta_{ci}|r_{ci}, q_{ci}, \alpha_v, \alpha_m) \propto p(r_{ci}|\eta_{ci})p(\eta_{ci}|q_{ci}, \alpha_v, \alpha_m) ,$$

using

$$\eta_{ci}|r_{ci}, q_{ci}, \alpha_v, \alpha_m \sim (1-q_{ci})\delta(\eta_{ci}) + q_{ci}\text{Beta}(\eta_{ci}|\alpha_v\alpha_m + r_{ci}, \alpha_v(1-\alpha_m) + 1 - r_{ci}) .$$

Similarly for the q_{ci}, we collapse η_{ci} to get

$$q_{ci}|r_{ci}, \alpha_m, \nu_c \sim \text{Bernoulli}\left(q_{ci}\middle|r_{ci} + (1-r_{ci})\frac{\nu_c(1-\alpha_m)}{1+\nu_c\alpha_m}\right) ,$$

i.e. we set $q_{ci} = 1$ if $r_{ci} = 1$. Finally

$$\nu_c|q_1, q_0, \beta_v, \beta_m \sim \text{Beta}(\nu_c|\beta_v\beta_m + q_1, \beta_v(1-\beta_m) - q_0) ,$$

where $q_1 = \sum_{i=1}^d q_{ci}$ and $q_0 = d - q_1$.

6. Sample from the weight variances τ_{ci} independently from $p(\tau_{ci}^{-1}|\cdot) \propto p(w_{ci}|\tau_{ci})p(\tau_{ci}^{-1}|t_s, t_r)$, i.e. a gamma distribution with shape $t_s + \frac{1}{2}$ and rate $t_r + \frac{w_{ci}^2}{2}$.

7. To make predictions on a test set $\mathbf{X}^\star$, we draw samples for $\mathbf{W}$ and $\mathbf{U}$ and use them to compute $p(\mathbf{f}^\star|\mathbf{W}, \mathbf{U}, \mathbf{X}^\star)$ in equation (4). The unidimensional intractable integral in equation (1) can be quite precisely approximated using Gauss-Kronrod quadrature.

The complexity of the proposed method is calculated as follows: the conditional posterior sequence is computed N_b times as burn-in period followed by N_s times to collect the samples needed to summarize the quantities of interest. Each iteration takes CdN elementary operations as the computation $\mathbf{WX}$ is the most expensive step. This leads to an asymptotic computational complexity of $O((N_b + Ns)CdN)$.

4 SIMULATION RESULTS

This section starts with a description of the practicalities of our model, meaning parameter settings, computational complexity considerations and procedures used to summarize and visualize the relevant posteriors of the classifier. Next we present two artificially generated examples designed to highlight the key features of our model. In particular, ability to select informative features, solution multiplicity in multi-category classification and how standard univariate techniques fail at selecting important variables when linear combinations of them are required to render them discriminant. The last experiment addresses a real study for large scale leukemia classification (Haferlach et al., 2010).

4.1 Parameter settings

The standard parameter settings described below were used in all experiments and should be applicable for most purposes. Here we only describe the most critical hyperparameters, the remaining ones are set to values as described above.

• *Number of sampling steps:* we collect $N_s = 2000$ posterior samples to compute summaries after a burn-in period of $N_b = 1000$ iterations.
• *Non-zero weights:* we want high probabilities for these weights, we set then $\alpha_m = 0.85$ and $\alpha_p = 10$ to be able to get a clear differentiation between zero and non-zero weights we allow for large variance.
• *Over-all sparsity level:* we allow only for small global effects by setting β_m to a small value, $10/d$ for example with a high precision $\beta_p = d$ to make it more informative, i.e. to promote highly sparse models. The dependence on the number of dimensions d simply indicates that we expect sparser models for larger values of d.

We consider that the settings above must be in general good choices towards reducing the false discovery rate. If a more or less sparse model is required, it can be achieved by increasing α_m for denser models or decreasing it otherwise. The precision parameter, α_p associated to α_m may be changed accordingly to obtain a reasonable separation of zeros and non-zeros when looking to the resulting credible intervals of $\{\eta_{ci}\}$.

Having in mind that the memory requirements of the sampler are rather high, i.e. $o(N_s C d)$ just for $\mathbf{W}$, it is a good idea to still select N_s to satisfy convergence but to collect only a fraction of them for posterior summaries, say 10% ($0.1N_s$). This practice called *thinning*, it is achieved by defining an additional stride parameter so we keep $N_s = 2000$ but with a stride set to 10.

4.2 Posterior summaries and visualization

The following posterior distributions are important for prediction and interpretation.

• *Predictions:* $p(y^\star = c|\mathbf{x}^\star, \mathbf{y}, \mathbf{X})$, the probability of a test observation $x^\star$ of being labeled as category c.
• *Weight matrix:* $p(w_{ci}|w_{ci} \neq 0, \mathbf{y}, \mathbf{X})$, the distribution of non-zero weights in $\mathbf{W}$.
• *Sparsity pattern:* $p(w_{ci} \neq 0|\mathbf{y}, \mathbf{X}) = p(\eta_{ci}|\mathbf{y}, \mathbf{X})$, the probability of an element of $\mathbf{W}$ of being non-zero.

Inference provides us with distributions for all relevant quantities, however we also want to compute some point estimate summaries to ease interpretation. The predictions are summarized using the sample mean. The weights are summarized as medians conditioned on the weight being non-zero. Sparsity matrices are summarized using modes, this is the most frequent sparsity pattern. Non-zero element probabilities are summarized using medians. We use medians because most of the posterior distributions of our model are either skewed or bimodal. Another useful summary is the amount of sparsity patterns produced by the inference procedure and their corresponding frequencies.

The parameters of the classifier relating categories and covariates, namely $\mathbf{W}$ and $\{\eta_{ci}\}$ can be visually summarized using a Hinton diagram as follows, each element w_{ci} conditioned on a specific sparsity pattern is represented as a square with color encoding its sign (+:green and –:red) and size proportional to its magnitude. In addition, each w_{ci} is provided with its median probability of being non-zero using η_{ci} and highlighted in bold in case that $\eta_{ci} \geq \alpha_m$, since $\mathbb{E}(w_{ci} \neq 0|\mathbf{X}, \cdot) = 1 - \beta_m(1 - \alpha_m)$. The entire plot defines a grid with categories in the rows and covariates in the columns relating how covariates affect different categories and admits one of the following interpretations (see Figure 3(a) as an example):

• An empty column (not plotted) means that covariate is not used, at least in average.
• A column with non-zero elements of the same sign means that the covariate is informative at separating the categories with non-zero coefficients from the remaining ones.
• A column with non-zero elements of different signs means that the covariate is informative at separating the categories with positive coefficients from the ones with negative coefficients. This means also that the covariate is specific to the categories used.
• An empty row does not mean that the classifier is not taking into account the category but that there are no covariates specific to that category as seen by the data.

It is important to bear in mind that it could be more than one sparsity pattern producing the same separating boundaries for different categories, hence the interpretation of the Hinton diagram could be one of many stories being told by the data, hence it should be taken into account when formulating hypothesis about the covariates being used by the classifier. Unfortunately, it is not possible in practice to know how many equivalent sparsity patterns are there given a dataset.

4.3 Finding informative covariates

First we want evaluate our model using artificially generated data to illustrate its main features and make the initial comparison with two well known methods, namely Linear Discriminant Analysis (LDA) and Support Vector Machines (SVMs) with polynomial kernel (Vapnik, 1998), with and without using univariate feature selection based upon F-tests. The SVM has two parameters namely C controlling the complexity of the model and p, the degree of the polynomial used to introduce non linearities to the model.

The artificial dataset consists on 120 covariates and 4 linearly separable categories. We generated each of the samples in the following way: the first two covariates were generated from Uniform$(-1,1)$ whereas the remaining ones were drawn from $\mathcal{N}(0,1)$, then we assigned the class labels to match the quadrant in which the sample lies using only the first two covariates, see Figure 2(a) (circles) for an illustration of the class assignments. The training and test sets have 60 and 900 observations, respectively. Each square in Figure 2(a) represent a test point and its size the class-conditional posterior probability, smaller markers indicate more uncertainty. Figure 3(a) shows a Hinton map that can be used to highlight two aspects of our model. (i) Individual weights w_{ci} are interpretable, e.g. $\mathbf{x}_1$ separates $c = 1,2$ from $c = 3,4$ and x_2 separates $c = 1,3$ from $c = 2,4$, which indeed can produce the two ideal boundaries located at the axes in the plane. (ii) There is not necessarily a unique representation for the non-empty columns of $\mathbf{W}$, i.e. it is possible that there is more than one solution for $\mathbf{W}$ producing the same separating boundary. In total, inference produced almost 2000 different sparsity patterns, however after dropping patterns with non-zero weights supported by less than 10% of the $N_s = 2000$ samples collected, we ended up with

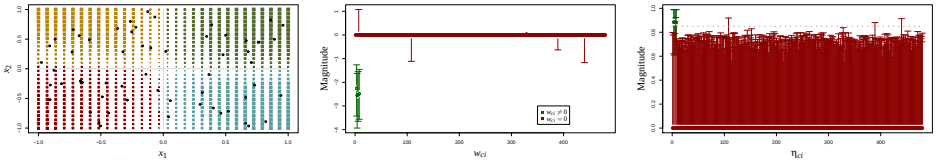


Fig. 2. Results for artificial data. (a) Training (circles) and test (dots) sets, the resulting test error is 1.5%. (b) Medians and 95% credible intervals for $\mathbf{W}$. (c) Probability of the elements of $\mathbf{W}$ of being non-zero with 95% credible intervals, the dotted line is the non-zero mean prior probability, α_m. Note the correspondence between error bars in (b) and those in (c) crossing the α_m threshold.

a single leading sparsity pattern, i.e. the mode of $\mathbf{W}$. Figure 2(b) shows the summary of all elements of $\mathbf{W}$ in the form of medians and 95% credible intervals depicted as error bars. Note that for none of the non-zero elements w_{ci} its credible interval cross zero as should be expected. Figure 2(c) shows $p(w_{ci} \neq 0|\mathbf{y},\mathbf{X})$ where the threshold (dotted line) is roughly α_m and the variance is inversely related to α_p, i.e. larger variances are due to small values of α_p. The predictions on the test set are shown as dots in Figure 2(a) superimposed to the training set (circles). Classification errors summarized in Figure 3(b) show the importance of selecting relevant covariates when it is known in advance that only a few (two in this case) are discriminant. Note that even a sophisticated classifier like SVM is not able to perform satisfactorily when there is a large amount of non-informative covariates and the number of observations is small.

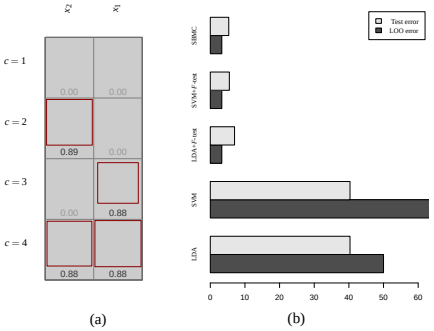


Fig. 3. Results for artificial data. (a) Hinton map summarizing the mode of $\mathbf{W}$ and $\{\eta_{ci}\}$, empty columns (118) are omitted. (b) LOO and test errors using our method (SBMC), LDA and SVM, with and without univariate t-tests based covariate selection.

It is important to mention that the time required to run LOO for SVM is comparable to our approach due to the inner-loop that needs to be done to select for the SVM parameters. In particular, to give an idea of empirical computational complexity, it took approximately 8 minutes to run the entire leave-one-out loop in a standard desktop machine (2.4GHz processor with 4GB RAM), which means that it takes about 8 seconds just to build the classifier.

4.4 Failure of univariate feature selection

In this experiment we consider a simple yet very likely binary classification scenario. Due to visualization purposes, we assume only two informative covariates out of $d = 120$ as in our previous experiment. Those two covariate follow a bi-variate Gaussian distribution as follows

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \middle| m_c\sqrt{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \frac{10}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \right),$$

where m_c is the sign of the category, particularly $m_1 = 1$ and $m_2 = -1$. The remaining $x_3,\ldots,x_{120}$ covariates follow univariate Gaussian distributions with zero mean and unit variance. This setting implies that in the plane, the two groups are completely separable by a line crossing the origin with negative unitary slope, see Figure 4. However, each covariate is not discriminative by itself due to the high correlation between x_1 and x_2, meaning that a univariate test will not be able to select either of them out of the total pool of 120 with probability greater than by chance. The predictions made by of our classifier using LOO validation on a dataset with $N = 60$ observations is shown in Figure 4, where the size and color of the circles is proportional to the predictive probability. The LOO error in this case is zero as must be expected since the task is linearly separable in $\mathbb{R}^2$.

We also tried LDA and SVM with and without univariate feature selection (t-tests), to obtain LOO errors between 36% and 50%, agreeing with the idea that t-tests are unable to select for the two relevant covariates. The classifiers without univariate feature selection are not able to correctly separate the two categories because the levels of noise from the non-informative dimensions relative to size of the training set just as in our previous experiment.

4.5 Pima data

Pima Indians Diabetes dataset (Asuncion and Newman, 2007) consist of 768 observations in 7 dimensions (features), divided in two classes. We want to use this small dataset to illustrate the kind of conclusions we want to obtain of the summaries of the model when used on real gene expression tasks. We ran our model using the traditional splitting, 300 observations for fitting the model and the remaining for testing. The mean test accuracy achieved by our model, 78.8% is slightly better than the results for LDA, 77.2%. Figure 5 the summary results. The central panel shows 10 sparsity patterns (rows) produced by the model during inference. The last column correspond to the bias term of the classifier that is non-zero a-priori. Inference produced a total of 50 sparsity patterns. The left panel shows the cumulative probability of the sparsity

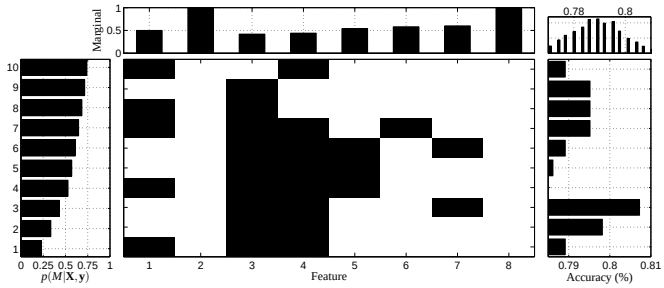


Fig. 5. Results for Pima Indian data. Central panel: top 10 sparsity patterns. Left panel: cumulative distribution of the sparsity patterns $M_1,\ldots,M_{10}$. Top panel: marginal probability of each feature. Right panel: test accuracy computed for each sparsity pattern. Top-right panel: test accuracy histogram.

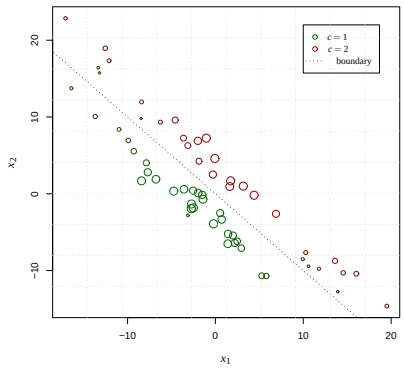


Fig. 4. Results for highly correlated data. The task is linearly separable with ideal separating boundary depicted as a dotted line. The size and color of the points is proportional to the predictive probability of the categories, i.e. $p(\mathbf{y}^\star = c|\mathbf{X}^\star,\mathbf{y},\mathbf{X},\cdot)$. Each point is a LOO based prediction.

patterns computed as their frequency during sampling. We can see that the first sparsity pattern dominates inference with probability close to 0.25. The remaining 10 patterns amount to 0.75 of the probability mass, meaning that the other 40 sparsity patterns are used very scarcely, i.e. approximately 25% of the time. The top panel represents the marginals for each feature. It shows for instance that features 2 and 8 (bias) are used in almost every sparsity pattern whereas features 3 and 4 are used in less than half of the them. The right panel shows the test accuracies computed for each sparsity pattern, where we can see that patterns 1 and 4 produce the best results. Examining the test errors and all sparsity patterns we can say that features 1 and 4 are jointly responsible for the good performance of the classifier when sparsity patterns 1 and 4 are selected. Finally, the top-right panel shows the distribution of the marginal test accuracy. Notice that there is almost no probability mass below 0.78 that is still better than the result obtained by LDA.

4.6 Leukemia data

This dataset is the result of a large study performed by 11 laboratories across three continents including 2096 patients (after quality control filtering) divided in 18 different leukemia and myelodysplastic syndromes including healthy specimens. The results presented by Haferlach et al. (2010) report a classification accuracy of 92.2% based on three repetitions of 30-fold cross-validation. Variable selection was done with t statistics by selecting the top 100 differentially expressed probes for each one of the 153 *one vs. one* SVM classifiers built.

We ran our model on a reduced set of probes selected by non-specific variance filtering. After removing all probes with standard deviations less than 1.5 we ended up with $d = 1637$. To give an idea of the computational cost, running the sampler took approximately 2 hours on a standard 2.4GHz desktop machine with 4GB RAM. When d is large, the number of possible sparsity patterns in $\mathbf{W}$ is very large — actually 2^{Cd}, thus summarizing all sparsity patterns produced by the model during sparsity is not feasible because N_s will be too small to be able to collect reasonable statistics. We proceed by only summarizing sparsity patterns for elements of $\mathbf{W}$ being non-zero at least 20% of the collected samples, i.e. $0.2N_s$. After this procedure, only 3411 non-empty weights (11.6%) were considered for sparsity pattern summarization, still the total number of patterns was 1223 out of $N_s = 2000$. Figure 6 shows the top 500 sparsity patterns sorted according to usage during inference. The left panel in Figure 6 presents the cumulative distribution of the sparsity patterns, from which we can see that the 500 patterns correspond to roughly 60% of the total produced by the model. A closer look to Figure 6 reveals that the top 6 patterns explain approximately 40% of the occurrences in the model and that those 6 patterns are indeed 99% identical. The top panel in Figure 6 shows the marginal of each element of $\mathbf{W}$ of being non-zero, $p(\eta_{ci} > 0|\cdot)$. We see that using marginals will be far more sparse than the summary of the 18 patterns that explain 20% of the model. In fact from the marginals alone, the averaged sparsity pattern is 92% sparse whereas the top used sparsity pattern is 88% sparse.

From Figure 6 we can think of several difficulties. First of all the total number of sparsity patterns is comparable to the number of collected posterior samples, meaning that most of the sparsity patterns are being used too little during inference making any

statistics computed on them rather pointless. Summarizing *unique* sparsity patterns is not feasible, we need to group sparsity patterns to increase the coverage, for instance by taking those 6 99% identical patterns and treating them as one. In other words, we need a clustering mechanism for summarizing our results if we want to handle real problems.

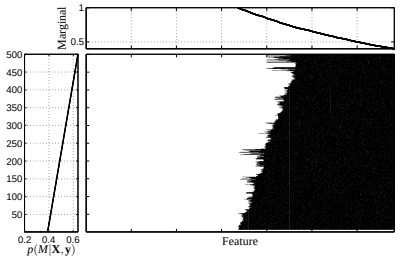


Fig. 6. Results for Leukemia data. The left panel shows the cumulative probability of the patterns and the top panel the marginals of the probability of each element of **W** of being non-zero. The main panel shows the top 500 sparsity patterns (rows, 4381 non-empty weights only) produced by our model during inference, sorted according to usage.

We computed cross-validation performances in the same way as in Haferlach et al. (2010) to obtain 91.1% accuracy, that is comparable to the results previously reported. We attribute the difference in performance to the fact we are pre-filtering the data to reduce the cost of fitting our model, so we are only considering 1637 probes. Our plan is to run our model on a larger version of the dataset once we have solved our summarization issues. The idea is to obtain a summary similar to the one of Figure 5 but with thousands of variables in a suitable way.

5 DISCUSSION AND PENDING WORK

We have proposed a new sparse hierarchical Bayesian multi-category linear classifier. The key element in our model is the spike and slab prior for the weight matrix which allows the classifier to perform variable selection automatically. In this way we consider a large number of potential hypotheses about the data by averaging. This procedure is not only computationally attractive but also avoids the need for multiple comparison correction arising in frequentist approaches to model selection. We successfully tested our model on artificial and real benchmark data to show that not only the classifier performs better than other state-of-the-art classifiers, but also produce robust and interpretable results.

The artificial classification tasks are set up to demonstrate the model features needed for successful classification of datasets with many more covariates than examples, many non-informative or redundant features and datasets where univariate feature selection (t-tests) will not help. A classifier that work on such datasets are likely to also work in gene expression profiling classification tasks. The success of our approach in these difficult scenario can be explain by the use of priors that explicitly model sparsity, Bayesian averaging to consider multiple hypotheses without overfitting and posterior summaries for model interpretation.

Pending work

There are two critical aspects of the model that need to be addressed in order to make our approach entirely useful in practice.

Reducing ambiguities. It is well known that in multi-category classification there is not a unique way to represent classification boundaries. This means that several weight matrices can produce the same classifier, so the model is unfortunately unidentifiable. As far as we know, there is nothing we can do to turn a linear multi-category classifier into a fully identifiable model, however we can at least try to remove as many redundancies as possible, mainly to improve mixing and to ease interpretation. For instance, we can share sparsity indicators across columns of **W**, meaning that we have to remove category specific sparsities, i.e. $r_{1i},\ldots,r_{Ci} = r_i$. This will not only make inference faster and reduce the number of discrete variables to be inferred but it will make interpretation easier in the sense that we will not have to think about category specific variables (genes). The latter is a nice feature to have in a model because we can for example relate genes to particular conditions/diseases, however we cannot be entirely sure if we are observing a truly biological interaction or a byproduct of the ambiguities of the model.

Covariation by combination. It is very common in microarray data that some groups of genes tend to have similar expression profiles. This kind of covariation is very important because it may indicate groups of genes conforming functional pathways. In sparse classification this phenomenon can be seen from two different angles. (i) Two highly correlated variables must be kept in the model if they as separate entities help to the classification problem. (ii) If the two variables are discriminant but highly correlated, there is not a reason to keep both of them in the model since we will have to increase the complexity of the model without need. In theory, our model must be able to handle the two angles successfully by indicating that the probability for the two variables of being in the model is 0.5, i.e. there are two modes, one of them using one of the variables and the other with the alternative. The problem is that in practice with thousands of variables and limited computational resources, such situations are hard to detect, thus it is very likely we end up with a model using one of the variables and discarding the other one. On way to avoid this effects is to group variables with similar profiles. For this purpose, we can borrow the ideas from our Latent Protein Tree model and perform classification on a set of *latent gene* variables instead of the original input space. This has the benefit of reducing the complexity of the sparse classification model while grouping variables with similar gene expression profiles, thus making interpretation more easy to handle.

More efficient inference. We need to improve inference if we want to scale our approach to datasets with say more than $d > 5000$. Current limitations include, mixing speed, memory requirements and cross-validation. Possible strategies include, model simplification, micro sampling and parallel computing.

Open questions

Possible extensions to our model include skewed distributions for unbalanced datasets, non-linear classifiers using Gaussian process priors Liang et al. (2005-09) and tree-like priors over **W** to infer hierarchical structures in the labeling space Kingman (1982), Chipman et al. (2002).

REFERENCES

J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422): 669–679, 1993.

D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society: Series B (Methodology)*, 36(1): 99–102, 1974.

A. Asuncion and D .J. Newman. UCI machine learning repository, 2007.

K. Bae and B. K. Mallick. Gene selection using a two-level hierarchical Bayesian model. *Bioinformatics*, 20(18):3423–3430, 2004.

G. C. Cawley, N. L.C. Talbot, and M. Girolami. Sparse multinomial logistic regression via Bayesian L1 regularisation. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 209–216. The MIT Press, Cambridge, MA, 2007.

M-H. Chen and D. K. Dey. Bayesian modeling of correlated binary responses via scale mixture of multivariate normal link functions. *Sankhya Series A*, 60(3):322–343, 1998.

H. A. Chipman, E. I. George, and R. E. McCulloch. Bayesian treed models. *Machine Learning*, 48(1-3):299–320, 2002.

G. Claeskens, C. Croux, and J. Van Kerckhoven. An information criterion for variable selection in support vector machines. *Journal of Machine Learning Research*, 9:541–558, 2008.

S. Dudoit and M. J. van der Laan. *Multiple testing procedures with applications to genomics*. Springer, New York, 2008.

E. I. George and R. E. McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

E. I. George and R. E. McCulloch. Approaches for Bayesian variable selection. *Statistica Sinica*, 7:339–373, 1997.

M. Girolami and S. Rogers. Variational Bayesian multinomial probit regression with Gaussian process priors. *Neural Computation*, 18(8): 1790–1817, 2006.

T. Haferlach, A. Kohlmann, L. Wieczorek, G. Basso, G. T. Kronnie, M.-C. Béné, John De Vos, J. M. Hernández, W.-K. Hofmann, K. I. Mills, A. Gilkes, S. Chiaretti, S. A. Shurtleff, T. J. Kipps, L. Z. Rassenti, A. E. Yeoh, P. R. Papenhausen, W.-M. Liu, P. M. Williams, and R. Foà. Clinical utility of microarray-based gene expression profiling in the diagnosis and subclassification of leukemia: Report from the international microarray innovations in leukemia study group. *Journal of Clinical Oncology*, 28 (15):2529–2537, 2010.

D. Hernandez-Lobato, J. M. Hernandez-Lobato, and A. Suarez. Expectation propagation for microarray data classificatio. *Pattern Recognition Letters*, 31(12):1618–1626, 2010.

C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

J. F. C. Kingman. The coalescent. *Stochastic processes and their applications*, 13(3):235–248, 1982.

Balaji Krishnapuram, Lawrence Carin, Mario A.T Figuereido, and Alexander J. Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Transactions on Pattern Aanalysis and Machine Intelligence*, 27(6):957–968, 2005.

K. E. Lee, N. Sha, E. R. Dougherty, M. Vanucci, and B. K. Mallick. Gene selection: a Bayesian variable selection approach. *Bioinformatics*, 19(1): 90–97, 2003.

F. B. Lempers. *Posterior Probabilities of Alternative Linear Models*. Rotterdam University Press, 1971.

F. Liang, K. Mao, M. Liao, S. Mukherjee, and M. West. Nonparametric bayesian kernel models. Discussion paper, Duke University ISDS, 2005-09.

J. Lucas, C. Carvalho, Q. Wang, A. Bild, J. R. Nevins, and M. West. *Bayesian Inference for Gene Expression and Proteomics*, chapter Sparse Statistical Modeling in Gene Expression Genomics, pages 155–176. Cambridge University Press, 2006.

T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404): 1023–1032, 1988.

J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. The MIT Press, 1999.

N. Sha, M. Vanucci, M. G. Tadesse, P. Brown, I. Dragoni, N. Davies, T. C. Roberts, A. Contestabile, M. Salmon, C. Buckley, and F. Falciani. Bayesian variable selection in multinomial probit models to identify molecular signatures of disease stage. *Biometrics*, 60:812–819, 2004.

B. Shahbaba and R. Neal. Nonlinear models using Dirichlet process mixtures. *Journal of Machine Learning Research*, 10:1829–1850, 2009.

A. Statnikov, C F. Alferis, I. Tsamardinos, D. Harris, and S. Levy. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21(5): 631–643, 2005.

V .N. Vapnik. *Statistical learning theory*. Wiley, New York, 1998.

M. West. Bayesian factor regression models in the "large *p*, small *n*" paradigm. In J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, editors, *Bayesian Statistics 7*, pages 723–732. Oxford University Press, 2003.

X. Zhou, K.-Y. Liu, and S. T. C. Wong. Cancer classification and prediction using logistic regression with Bayesian gene selection. *Journal of Biomedical Informatics*, 37:249–259, 2004a.

Xiaobo Zhou, Xiaodong Wang, and Edward R. Dougherty. Gene prediction using multinomial probit regression with Bayesian gene selection. *EURASIP Journal on Advances in Signal Processing*, 2004(1):115–124, 2004b.

Bibliography

R. P. Adams, Z. Ghahramani, and M. I. Jordan. Tree-structured stick breaking for hierarchical data. In *Advances in Neural Information Processing Systems 24*. MIT Press, 2010.

J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society: Series B (Methodology)*, 36(1):99–102, 1974.

K. Bae and B. K. Mallick. Gene selection using a two-level hierarchical Bayesian model. *Bioinformatics*, 20(18):3423–3430, 2004.

J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, second edition, 1985.

J. M. Bernardo and A. F. M. Smith. *Bayesian theory*. Wiley, 1994.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

C. Blundell, Y. W. Teh, and K. A. Heller. Bayesian rose trees. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 2010.

M. Branco and D. K. Dey. A general class of multivariate skew-elliptical distributions. *Journal of Multivariate Analysis*, 79(1):99–113, 2001.

F. Caron and A. Doucet. Sparse Bayesian nonparametric regression. In *Proceedings of the 25-th International Conference on Machine Learning*, pages 88–95, 2008.

C. M. Carvalho, J. Chang, J. E. Lucas, J. R. Nevins, Q. Wang, and M. West. High-dimensional sparse factor modeling: Applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484):1438–1456, 2008.

C. M. Carvalho, N. G. Polson, and J. G. Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

G. Casella. An introduction to empirical Bayes data analysis. *American Statistician*, 39(2):83–87, 1985.

G. C. Cawley, N. L.C. Talbot, and M. Girolami. Sparse multinomial logistic regression via Bayesian L1 regularisation. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 209–216. The MIT Press, Cambridge, MA, 2007.

M-H. Chen and D. K. Dey. Bayesian modeling of correlated binary responses via scale mixture of multivariate normal link functions. *Sankhya Series A*, 60 (3):322–343, 1998.

R. S. Chhikara and L. Folks. *The Inverse Gaussian Distribution: Theory, Methodology, and Applications*. M. Dekker, New York, 1989.

S. Chib. Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association*, 90(732):1313–1321, 1995.

G. Claeskens, C. Croux, and J. Van Kerckhoven. An information criterion for variable selection in support vector machines. *Journal of Machine Learning Research*, 9:541–558, 2008.

L. Csató. *Gaussian Processes - Iterative Sparse Approximations*. PhD thesis, Aston University, 2002.

D. S. Daly, K. K. Anderson, E. A. Panisko, S. O. Purvine, R. Fang, M. E. Monroe, and S. E. Baker. Mixed-effects statistical model for comparative LCMS proteomics studies. *Proteomics Research*, 7(3):1209–1217, 2008.

P. Daniusis, J. Janzing, J. Mooij, J. Zscheischler, B. Steudel, K. Zhang, and B. Schölkopf. Inferring deterministic causal relations. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 2010.

D. DeCoste and B. Schölkopf. Training invariant support vector machines. *Machine Learning*, 46(1-3):161–190, 2002.

A. Doucet, N. de Freitas, N. Gordon, and A. Smith. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

S. Dudoit and M. J. van der Laan. *Multiple testing procedures with applications to genomics*. Springer, New York, 2008.

G. Elidan, I. Nachman, and N. Friedman. "Ideal Parent" structure learning for continuous variable Bayesian networks. *Journal of Machine Learning Research*, 8:1799–1833, 2007.

M. A. T. Figuereido. Adaptive sparseness for supervised learning. *IEEE Transactions on Pattern Aanalysis and Machine Intelligence*, 25(9):1150–1159, 2003.

N. Friedman and I. Nachman. Gaussian process networks. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 211–219. 2000.

N. Friedman, I. Nachman, and D. Pe'er. Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm. In K. B. Laskey and H. Prade, editors, *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 206–215, 1999.

N. Friel and A. N. Pettitt. Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Methodology)*, 70(3):589–607, 2008.

S. Gama-Castro, V. Jiménez-Jacinto, M. Peralta-Gil, A. Santos-Zavaleta, M. I. Peñaloza-Spinola, B. Contreras-Moreira, J. Segura-Salazar, L. Muñiz-Rascado, I. Martínez-Flores, H. Salgado, C. Bonavides-Martínez, C. Abreu-Goodger, C. Rodríguez-Penagos, J. Miranda-Ríos, E. Morett, E. Merino, A. M. Huerta, L. Treviño-Quintanilla, and J. Collado-Vides. RegulonDB (version 6.0): gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and textpresso navigation. *Nucleic Acids Research*, 36(Database Issue):120–124, 2008.

A. Gelman, J. B. Carlin, H. S. Stern, and D. Rubin. *Bayesian data analysis*. Chapman Hall/CRC, second edition, 2004.

E. I. George and R. E. McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

E. I. George and R. E. McCulloch. Approaches for Bayesian variable selection. *Statistica Sinica*, 7:339–373, 1997.

J. Geweke. Variable selection and model comparison in regression. In J. Berger, J. Bernardo, A. Dawid, and A. Smith, editors, *Bayesian Statistics 5*, pages 609–620. Oxford University Press, 1996.

Z. Ghahramani, T. L. Griffiths, and P. Sollich. Bayesian nonparametric latent feature models. In J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, editors, *Bayesian Statistics 8*, pages 201–226. Oxford University Press, 2006.

M. Girolami and S. Rogers. Variational Bayesian multinomial probit regression with Gaussian process priors. *Neural Computation*, 18(8):1790–1817, 2006.

J. E. Griffin and P. J. Brown. Alternative prior distributions for variable selection with very many more variables than observations. Technical report, University of Warwick, 2005.

D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1:49–75, 2000.

K. A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the 22nd international conference on Machine learning*, pages 297–304. ACM, 2005.

R. Henao and O. Winther. Bayesian sparse factor models and DAGs inference and comparison. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 736–744. The MIT Press, 2009.

R. Henao and O. Winther. PASS-GP: Predictive active set selection for gaussian processes. In *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*, pages 148–153, 2010.

R. Henao and O. Winther. Sparse linear identifiable multivariate modeling. *Journal of Machine Learning Research*, To appear, 2011.

D. Hernandez-Lobato, J. M. Hernandez-Lobato, and A. Suarez. Expectation propagation for microarray data classificatio. *Pattern Recognition Letters*, 31 (12):1618–1626, 2010.

P. O. Hoyer, S. Shimizu, A. J. Kerminen, and M. Palviainen. Estimation of causal effects using linear non-Gaussian causal models with hidden variables. *Interantional Journal of Approximate Reasoning*, 49(2):362–378, 2008.

P .O. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Schölkopf. Nonlinear causal discovery with additive noise models. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 689–696. 2009.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley-Interscience, 2001.

H. Ishwaran and A. Papana. Orthogonalized smoothing for rescaled spike and slab models. In *Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh*, volume 3 of *IMS Collections*, pages 267–281. IMS, 2008.

H. Ishwaran and J. S. Rao. Spike and slab variable selection: Frequentist and Bayesian strategies. *Annals of Statistics*, 33(2):730–773, 2005.

T. Joachims and C-N. J. Yu. Sparse kernel SVMs via cutting-plane training. *Machine Learning*, 76:179–193, 2009.

A. M. Kagan, YU. V Linnik, and C. Radhakrishna Rao. *Characterization Problems in Mathematical Statistics*. Probability and Mathematical Statistics. Wiley, New York, 1973.

K. C. Kao, Y-L. Yang, R. Boscolo, C. Sabatti, V. Roychowdhury, and J. C. Liao. Transcriptome-based determination of multiple transcription regulator activities in Escherichia Coli by using network component analysis. *PNAS*, 101(2):641–646, 2004.

Y. V. Karpievitch, J. Stanley, T. Taverner, J. Huang, J. N. Adkins, C. Ansong, F. Heffron, T. O. Metz, W.-J. Qian, H. Yoon, R. D. Smith, and A. R. Dabney. A statistical framework for protein quantitation in bottom-up MS-based proteomics. *Bioinformatics*, 25:2028–2034, 2009.

S. S. Keerthi, O. Chappelle, and D. DeCoste. Building support vector machines with reduced classifier complexity. *Journal of Machine Learning Research*, 7: 1493–1515, 2006.

A. Keller, A. I. Nesvizhskii, E. Kolker, and R. Aebersold. Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Analytica Chemistry*, 74(20):5384–5392, 2002.

J. F. C. Kingman. The coalescent. *Stochastic processes and their applications*, 13(3):235–248, 1982a.

J. F. C. Kingman. On the genealogy of large populations. *Journal of Applied Probability*, 19:27–43, 1982b.

D. Knowles and Z. Ghahramani. Infinite sparse factor analysis and infinite independent components analysis. In M. E. Davies, C. C. James, S. A. Abdallah, and M. D. Plumbley, editors, *7th International Conference on Independent Component Analysis and Signal Separation*, volume 4666 of *Lecture Notes in Computer Science*, pages 381–388. Springer-Verlag, Berlin, 2007.

Balaji Krishnapuram, Lawrence Carin, Mario A.T Figuereido, and Alexander J. Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Transactions on Pattern Aanalysis and Machine Intelligence*, 27(6):957–968, 2005.

M. Kuss and C. E. Rasmussen. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research*, 6: 1679–1704, 2005.

M. Kuss and C. E. Rasmussen. Assessing approximations for gaussian process classification. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 699–706. The MIT Press, Cambridge, MA, 2006.

N. D. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 600–616. The MIT Press, Cambridge, MA, 2003.

N. D. Lawrence, J. C. Platt, and M. I. Jordan. Extensions of the informative vector machine. In J. Winkler, N. D. Lawrence, and M. Niranjan, editors, *Proceedings of the Sheffield Machine Learning Workshop*. Springer-Verlag, Berlin, 2005.

K. E. Lee, N. Sha, E. R. Dougherty, M. Vanucci, and B. K. Mallick. Gene selection: a Bayesian variable selection approach. *Bioinformatics*, 19(1):90–97, 2003.

F. B. Lempers. *Posterior Probabilities of Alternative Linear Models*. Rotterdam University Press, 1971.

F. Liang, K. Mao, M. Liao, S. Mukherjee, and M. West. Nonparametric bayesian kernel models. Discussion paper, Duke University ISDS, 2005-09.

J. Lucas, C. Carvalho, Q. Wang, A. Bild, J. R. Nevins, and M. West. *Bayesian Inference for Gene Expression and Proteomics*, chapter Sparse Statistical Modeling in Gene Expression Genomics, pages 155–176. Cambridge University Press, 2006.

D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

J. Mooij and D. Janzing. Distinguishing between cause and effect. In *JMLR Workshop and Conference Proceedings*, volume 6, pages 147–156, 2010.

I. Murray. *Advances in Markov Chain Monte Carlo Methods*. PhD thesis, Gatsby computational neuroscience unit, University College London, 2007.

A. Naish-Guzman and S. Holden. The generalized FITC approximation. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1057–1064. MIT Press, Cambridge, MA, 2008.

R. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

R. Neal. Density modeling and clustering using Dirichlet diffusion trees. In J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, editors, *Bayesian Statistics 7*, pages 619–629. Oxford University Press, 2003.

A. Nesvizhskii. A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics. *Journal of Proteomics*, 73(11):2092–2113, 2010.

T. Park and G. Casella. The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6: 1939–1959, 2005.

P. Rai and H. Daume III. The infinite hierarchical factor regression model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1321–1328. The MIT Press, 2009.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA, 2006.

K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

M. W. Schmidt, A. Niculescu-Mizil, and K. P. Murphy. Learning graphical model structure using L1-regularization paths. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, pages 1278–1283, 2007.

M. Seeger. *Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations*. PhD thesis, University of Edinburg, 2003.

N. Sha, M. Vanucci, M. G. Tadesse, P. Brown, I. Dragoni, N. Davies, T. C. Roberts, A. Contestabile, M. Salmon, C. Buckley, and F. Falciani. Bayesian variable selection in multinomial probit models to identify molecular signatures of disease stage. *Biometrics*, 60:812–819, 2004.

B. Shahbaba and R. Neal. Nonlinear models using Dirichlet process mixtures. *Journal of Machine Learning Research*, 10:1829–1850, 2009.

S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7: 2003–2030, 2006.

R. Silva. *Causality in the Sciences*, chapter Measuring Latent Causal Structure. Oxford University Press, 2010.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. The MIT Press, second edition, 2001.

A. Statnikov, C F. Alferis, I. Tsamardinos, D. Harris, and S. Levy. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21(5):631–643, 2005.

Y. W. Teh, H. Daume III, and D. Roy. Bayesian agglomerative clustering with coalescents. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1473–1480. MIT Press, 2008.

M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pages 548–549, 2005.

R. Thibaux and M. I. Jordan. Hierarchical beta processes and the indian buffet process. In M. Meila and X. Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, pages 564–571, 2007.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodology)*, 58(1):267–288, 1996.

R. Tillman, A. Gretton, and P. Spirtes. Nonlinear directed acyclic structure learning with weakly additive noise models. In *Advances in Neural Information Processing Systems 22*, pages 1847–1855. Y. Bengio and D. Schuurmans and J. Lafferty and C. K. I. Williams and A. Culotta, 2009.

M. Tipping. Sparse Bayesian learning and the relevance vector machin. *Journal of Machine Learning Research*, 1:211–244, 2001.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

M. West. On scale mixtures of normal distributions. *Biometrika*, 74(3):646–648, 1987.

M. West. Bayesian factor regression models in the "large p, small n" paradigm. In J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, editors, *Bayesian Statistics 7*, pages 723–732. Oxford University Press, 2003.

G. Whaba. A comparison of GCV and GML for choosing the smoothing parameter in the generalized spline smoothing problem. *Numerical Mathematics*, 24: 383–393, 1975.

C. K. I. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In T. K. Leen, T. D. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.

A. K. Zaas, M. Chen, J. Varkey, T. Veldman, A. O. Hero, J. Lucas, Y. Huang, R. Turner, A. Gilbert, R. Lambkin-Williams, N. C. Øien, B. Nicholson, S. Kingsmore, L. Carin, C. W. Woods, and G. S. Ginsburg. Gene expression signatures diagnose influenza and other symptomatic respiratory viral infections in humans. *Cell*, 6(3):207–217, 2009.

K. Zhang and A. Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 647–655. AUAI Press, 2009.

K. Zhang and A. Hyvärinen. Distinguishing causes from effect using nonlinear acyclic causal models. In *JMLR Workshop and Conference Proceedings*, volume 6, pages 157–164, 2010.

X. Zhou, K.-Y. Liu, and S. T. C. Wong. Cancer classification and prediction using logistic regression with Bayesian gene selection. *Journal of Biomedical Informatics*, 37:249–259, 2004a.

Xiaobo Zhou, Xiaodong Wang, and Edward R. Dougherty. Gene prediction using multinomial probit regression with Bayesian gene selection. *EURASIP Journal on Advances in Signal Processing*, 2004(1):115–124, 2004b.