University of Groningen

Divergences for prototype-based classification and causal structure discovery

Mwebaze, Ernest

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

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

Publication date:
2014

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):
Mwebaze, E. (2014). Divergences for prototype-based classification and causal structure discovery: Theory and application to natural datasets [S.l.]: [S.n.]

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Download date: 11-02-2018

Divergences for prototype-based classification and causal structure discovery

Theory and application to natural datasets

Ernest Mwebaze

university of
groningen

Divergences for prototype-based classification and causal structure discovery

Theory and application to natural datasets

PhD thesis

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus Prof. E. Sterken
and in accordance with the decision by the College of Deans.

This thesis will be defended in public on

Friday 19 September 2014 at 14.30 hours

by

Ernest Mwebaze

born on 7 August 1979
in Kampala, Uganda.

Supervisor
Prof. dr. M. Biehl

Co-supervisor
Dr. J. A. Quinn

Assessment committee
Prof. dr. U. Seiffert
Prof. dr. E. Merenyi
Prof. dr. J. Nerbonne

Contents

Acknowledgements **ix**

1 Introduction **1**
- 1.1 Prototype-based classification 2
  - 1.1.1 Divergences as dissimilarity metrics 3
- 1.2 Causal modeling 3
  - 1.2.1 Causal modeling with observational data 4
- 1.3 Application of machine learning to natural datasets 5
- 1.4 Outline of thesis 6

2 Competitive Learning **9**
- 2.1 Introduction 9
- 2.2 Learning Vector Quantization and its variants 10
  - 2.2.1 Classical LVQ 11
  - 2.2.2 Generalized LVQ 12
  - 2.2.3 Generalized Matrix LVQ 13
- 2.3 Dissimilarity metrics and measures 14
  - 2.3.1 Metrics and measures 14
  - 2.3.2 Minkowski metrics 15
  - 2.3.3 Divergences as distance measures 15

3 Divergence based Classification in LVQ **19**
- 3.1 Introduction 19
- 3.2 Divergence based Learning Vector Quantization 20
  - 3.2.1 Influence of parameter γ in γ-divergences 22
- 3.3 Data sets and classification problems 23

- 3.3.1 Wisconsin Breast Cancer (WBC) data 23
- 3.3.2 Lung cancer data 23
- 3.3.3 Cassava Mosaic Disease (CMD) data 24
- 3.4 Computer experiments 24
- 3.5 Results 25
  - 3.5.1 Euclidean distance and Cauchy-Schwarz divergence 25
  - 3.5.2 The family of γ-divergences 26
- 3.6 Summary and Conclusion 28

4 LVQ with Combined Distance Measures **31**

- 4.1 Introduction 31
- 4.2 Adaptive distance measures in LVQ 32
- 4.3 Combining distance measures in LVQ 34
- 4.4 Experiments 36
  - 4.4.1 Analysis 37
  - 4.4.2 Matrix initialization 39
- 4.5 Prototype membership 40
- 4.6 Conclusion 41

Appendices

- 4.A Derivatives of prototype updates for *Combined-Distances LVQ* 43
- 4.B Derivatives of the matrix update for *Combined-Distances LVQ* 44

5 Causal Structure Learning **45**

- 5.1 Introduction 45
- 5.2 Bayesian graph representations 46
  - 5.2.1 Causal Bayesian networks 50
- 5.3 Model search 52
  - 5.3.1 Constraint-based search 52
  - 5.3.2 Bayesian search and score 54

6 Causality and LVQ **57**

- 6.1 Introduction 57
- 6.2 Identifying causes in observational data 58
- 6.3 RLVQ 59
- 6.4 Causal-RLVQ 61
- 6.5 Experiments and results 62
- 6.6 Conclusion 65

7 **Committee-Based Structure Learning** **67**
7.1 Introduction 67
7.1.1 Skeleton discovery 68
7.1.2 Causal discovery committee 70
7.1.3 Expected Partial Correlation (EPC) method 71
7.1.4 Other committee members 73
7.2 Evaluation and results 74
7.2.1 Evaluation results using known causal structures 75
7.2.2 Evaluation results for the LOCANET challenge 77
7.3 Conclusion 79

8 **Committee-based Independence Testing** **81**
8.1 Introduction 81
8.2 Hard-decision approach, DD1 83
8.2.1 Prediction of test performance 83
8.2.2 Combining independence tests 85
8.3 Soft-decision based approach, DD2 87
8.3.1 Generation of synthetic data 88
8.3.2 Formulating the DD2 test 88
8.4 Experiments 90
8.4.1 Approach 1, DD1 experimental results 91
8.4.2 Approach 2, DD2 experimental results 95
8.5 Conclusion 97

9 **Causal Structure Learning for Famine Prediction** **99**
9.1 Background to the problem 99
9.2 Causal structure discovery 100
9.3 Famine data 101
9.4 Experiments and results 102
9.4.1 *A Priori* graph 103
9.4.2 Graph learnt with uninformative prior 105
9.4.3 *A Posteriori* graph 105
9.5 Classification of famine risk 107
9.6 Conclusion 107

10 **Implementation of a Crop Disease Surveillance System** **111**
10.1 The problem 111
10.2 Improved survey system design 113
10.3 Mobile data collection and mapping 114
10.4 Image-based diagnosis of crop disease 116

10.4.1 Data collection 117
10.4.2 Feature extraction 117
10.4.3 Classification 120
10.4.4 Empirical results 120
10.4.5 Deployment on mobile devices 121
10.4.6 Automated whitefly count 122
10.5 Practical considerations for system deployment 122
10.6 Potential impact and lessons learnt 124

11 Summary **127**
11.1 Future work 129

Bibliography **131**

Samenvatting **139**

Acknowledgments

The journey that has culminated in this work has been very stimulating and challenging both intellectually and socially. I am grateful to all the people that have made this possible. In the first place, I would like to thank my promoters Prof. Michael Biehl and Dr. John A. Quinn. The Ph.D. would not have been possible without them. I am very grateful to Michael for having accepted to take me on as a student on the NUFFIC NPT program and making my time at the University of Groningen and my stay in the Netherlands very comfortable and beneficial both academically and socially. I thank John for the dedicated and inspiring supervision I got while in Makerere and for all the innovative ideas and opportunities he exposed me to. I will be eternally thankful for the supervision I received from Michael and John.

Special thanks go to the LVQ group at the University of Groningen: Kerstin Bunte, Petra Schneider, Aree Witoelar and Gjalt Bearda with whom I worked closely during my stay in Groningen. Many thanks for all the assistance and discussions and social activities we had. Thank you especially for your willingness to help and offer advice all the time.

To the intelligent systems research group at the University of Groningen, I would like to say a heartfelt thank you for all the support and all the discussions we had during my several trips to Groningen. Particularly I would like to thank Prof. Nicolai Petkov, the head of the group who accepted me to join and do research under this group. I would also like to thank Dr. Michael Wilkinson for so many informative discussions during my stay in Groningen. To the other group members: George Azzopardi, Ioannis Giotis, Ugo Moschini, Laura Fernandez, and all the rest, I would like to say sincere thanks for all the good times we had.

I would like to acknowledge all the members of the Ugandan students group in Groningen without whom life in Groningen would have been cheerless. To my fellow Ugandan Ph.D. students in Groningen: Johnson Mwebaze, Fred Kiwanuka, Deborah Mudali, Peace Tumuheki, Mercy Amiyo, Paul Ssemaluulu and the rest, special thanks for being there in all those trying times. To the other Ugandan and African friends in Groningen, very many thanks for all those wonderful social activities that made me feel so at home even in the trying times of the Ph.D.

To my family: my wife Rita Mwebaze, and my three children Trinity, Troy and Teresa, you are my life. Heartfelt thanks for all the moral assistance and love that you accorded me during my studies. During those long absences from home you kept me focused on the goal - I cannot say thank you enough. To my parents, Mr. and Mrs. Bwiragura, my siblings, and relatives, many thanks as well for all the support and love and encouragement that you offered me.

Finally I would like to thank the NUFFIC NPT program team particularly Prof. John Nerbonne, Erik Haarbrik and Gonny Lakerveld for taking care of my well being whilst in Groningen. To the administrative staff at the Johann Bernoulli Institute in Groningen, very many thanks for all the assistance offered to me. Particular thanks go to Ineke Schelhaas, Esmee Elshof and Desiree Hansen. Thank you for making my visits to the institute and to the Netherlands harmonious.

Ernest Mwebaze
Groningen

Chapter 1

Introduction

Data proliferation is increasingly becoming a challenge in the modern world, not only for organizations and companies but for individuals as well. The challenge faced is that at present there is an abundance of data presented in numerous formats e.g. structured, unstructured, heterogeneous or semantic that is daily produced from an ever increasing number of sensors. Innovative ways of handling such data need to be sought.

One branch of science that deals with making sense of data is machine learning. Machine learning is a branch of Artificial Intelligence that attempts to empower a machine or computer with the ability to perform a certain task accurately or intelligently given new data after having been trained on earlier observations of *similar* data. The machine learning task is to enable the learner (computer, machine) to identify the underlying data-generating mechanism from the training (historical) data and be able to generalize well on new data. Generalization in this case means the ability to make accurate deductions given new unseen data. Machine learning is more broadly broken down into a taxonomy of three main types of learning; supervised, unsupervised and reinforcement learning.

Supervised learning involves training the learner on labelled data so that it can predict the label associated with new data from a similar distribution as the training data. The labels can be viewed as classes in which case the learning can also be termed classification. In unsupervised learning the training data is not labelled. A common application of unsupervised learning is in clustering where the learner attempts to group the data in related clusters. Other examples include density estimation, correlation analysis and dimensionality reduction. In reinforcement learning, the learning is guided by a reward system that reinforces *good* behaviour and penalizes *bad* behaviour of the learner; a good application of such methods is in the field of robotics. Detailed coverage of machine learning can be obtained from the literature, see for example (Bishop 2006, Mitchell 1997, Duda et al. 2000).

Each branch of machine learning is very broad; this thesis will mainly focus on supervised learning, and classification in particular. A good example of the classification task in every day life is in the implementation of an email spam filter where a classification algorithm is built to assign the class NORMAL to ordinary mail and the class SPAM to spam or junk mail. The classification problem in this case involves building an algorithm that will be able to classify new incoming mail data automatically given an ample amount of previous examples of NORMAL and SPAM mail. The classifier is built from a clever analysis of the example mail data, a process called training. Several classification methods or algorithms abound that can be used for this task including Support Vector Machines, Neural Networks or Decision Trees. In this thesis, our scope will be limited to prototype-based classification methods.

1.1 Prototype-based classification

In prototype-based classification schemes, a set of representative points are defined in the space of the data to represent each of the classes or labels. These points are called prototypes and are characterized by being in the space of the data and being similar to the data. Classification is done through a nearest-prototype scheme where a new data example is assigned to the class of the nearest prototype. The assignment is determined by a similarity (or dissimilarity) or closeness measure that defines how close the example is to the different prototypes. The data example is assigned the class of the *closest* prototype.

The greatest advantage of this method is that the prototypes are easily interpretable since they are defined in the space of the data. As such the trained system offers clear insight to the user of such a system about what the learned prototypes represent. It also offers a straight forward way of validating the systems output. Several prototype-based and pseudo-prototype-based classification systems are in existence including Learning Vector Quantization (LVQ), Radial Basis Neural Networks, *k*-NN classifiers or Support Vector Machines. We narrow our scope in this thesis to the study of Learning Vector Quantization. LVQ is generally viewed as the quintessential prototype-based classification system.

Something is to be said about the calculation of *closeness* of data points to prototypes. Because the prototypes are in the space of the data, a distance measure (or dissimilarity) is used to quantify how *close* an example from the data is to the prototype. Typical dissimilarity measures used are those based on Minkowski-like distance metrics e.g. Euclidean distance metrics or Manhattan distance metrics. A better description of these metrics follows in Chapter 2. Under certain conditions, it is possible to replace the Minkowski-like measures with divergences as a distance measure. Divergences can be used to quantify the amount of information (cross entropy) between the data example and the prototype.

1.1.1 Divergences as dissimilarity metrics

Information theoretic measures (divergences) tend to be suited for certain types of data that can be represented by distributions or that are normalized and non-negative. Divergences as such can be used as *distance* measures in this respect. Divergences are generally not considered to be real metrics because they do not adhere to the triangular properties, but a consistent formulation of divergences can be used in LVQ in a manner similar to real distances (e.g. Minkowski distances). We look at reasons for this in Chapter 2.

Divergences have been discussed in the context of various machine learning frameworks in the literature, including prototype-based clustering, classification, or dimension reduction (Jang et al. 2008, Banerjee et al. 2005, Villmann et al. 2008, Torrkola 2003, Bunte et al. 2010). In this thesis we present novel work on the use of divergences as a distance measure in the training of an LVQ prototype-based classifier. We show that Divergence-based LVQ (DLVQ) – the novel algorithm that we implement – is a better suited classification technique of those types of data that can be represented as distributions and empirical results validate this assertion. We illustrate the technique with both artificial datasets and natural datasets from the medical field and from the field of crop disease surveillance. In Chapter 3 and Chapter 10, we also present actual system implementations of this technique as applied to the problem of cassava crop disease diagnosis and surveillance.

1.2 Causal modeling

The essence of machine learning lies in trying to capture the underlying model in the data (the generative model). The generalization of the model is tightly coupled with how well the machine learning algorithm captures the generative model of the data. Causal modeling has a similar aim but goes deeper to try to discover what *causes* entail that generative model (Pearl 2009). For example if the three variables X, Y, Z are associated with a disease outbreak Q, a machine learning predictive model would try to discover a function f that maps the variables to the disease outbreak $f(X, Y, Z) \mapsto Q$. Causal modeling would try to discover the causal relationships between the variables e.g. $X \rightarrow Y \leftarrow Z$ and perhaps $Y \rightarrow Q$. Understanding the causal relation avails one with the tools to understand the effect of manipulations, or to do appropriate feature selection for prediction.

Causality forms part of our every day life, we are always curious to find out what 'caused' the stock market to fail, or what drug will 'cause' the quickest recovery of a patient, or what was the 'cause' of such a disease outbreak. Formally however it is not trivial to discover causal relations because of the ease of mixing up correlation with causation. This is evident in many media and statistical reports. *Classical paradoxes* is the phrase that has been coined by some to illustrate this mix up (Arah 2008).

A simplistic example would be to consider the correlation between time spent in bed and the death rate. Studying these two variables would show a positive correlation between them. It would hence seem like the most obvious conclusion is spending time in bed 'causes' death, and the converse; not sleeping at all ensures you live for ever. However if we apply an intervention to the system, e.g. forcing people to spend more time in bed, there probably would be no increase in the death rate (depending on how much force you use of course). A close analysis would indicate that there is another variable, sickness in particular, that causes both an increase in time spent in bed and the death rate. Causal modeling is thus challenging because the process is mired by confounding and latent variables and as such certain assumptions have to be made to keep the problem manageable.

Previous studies of causality (Shoham 1998, Suppes 1998) have emphasized temporal precedence as an essential component of defining causation; so if the causal relation $X \rightarrow Y$ exists, then X must have occurred before Y. This condition however does not necessarily hold, temporal information alone cannot be used to infer causation amongst a set of variables. Statistical and philosophical literature is rife with explicit warnings that unless one knows in advance all causally relevant variables or unless one can manipulate variables in a controlled experiment, no genuine inference is possible. For example going to bed after dinner every day does not necessarily mean having dinner caused one to go to bed. To know, with a high degree of certainty, that the relationship $X \rightarrow Y$ actually exists one must be able to manipulate X in a certain way and a corresponding effect should be seen at Y.

1.2.1 Causal modeling with observational data

Manipulating variables to determine causation is ordinarily carried out by experimentation where one set of objects is intervened upon in a certain way and another set of objects is intervened upon in a different way, or not at all. Any results spawning from these different sets are then attributed to the interventions. This is the case with Randomized Controlled Trials (RCTs). While this method tends to work for some situations e.g. determining the effects of new drugs by subjecting alternate groups of people to the drugs and placebos, it does not work for situations where the effects are adverse, for example one cannot force people to smoke heavily to determine if smoking causes lung cancer. Besides these methods tend to be very expensive. One controversial approach to this problem has been to do causal inference from purely observational data. The controversy is clear, given a number of variables with no further information about context it is almost impossible to differentiate the effects of causation, correlation and confounding factors in the relationship between any subset of these variables.

Research from the last two decades (Pearl 2000, Pearl 2009, Spirtes et al. 2000) however, indicates that under certain assumptions of the generating process it is possible to infer a causal relationship between variables from non-temporal statistical data (observational data). Causal modeling is pursued generally under two broad categorizations: (1) causal Bayesian models, (2) structural equation models (SEMs). Causal Bayesian models specify a density for a variable as a function of the values of its causes (a child variable is specified by its parents). Structural equation models specify the value of a variable as a function of the values of its causes plus some noise term. Results from these two approaches are generally similar. In this thesis we consider discovery of causal structures from observational data using causal Bayesian models and delve into some of the assumptions that make this feasible. We present some novel methods of doing causal discovery from purely observational data and also present examples from natural datasets of famine where such methods improve the understanding of causes of such phenomena. We also discuss about prediction using causally relevant features.

1.3 Application of machine learning to natural datasets

Some recent publications (Wagstaff 2012, Langley 2011) argue that the most part of machine learning research is wholly divorced from real world problems. Arguments lie around the over reliance on so-called benchmark datasets e.g. UCI datasets (Asuncion et al. 1998), the use of abstract evaluation metrics and the lack of a link between the impact of experimental results and real world problems. The arguments tend to focus on the question "what is the real impact of the results from the machine learning gymnastics ?" and relatedly "what do the results say about the real problem ?", e.g. a 10% improvement in accuracy of a classifier that predicts the outcome of which student will pass a course in school has a markedly different impact from a 10% improvement in accuracy of a cancer-onset classifier. In some other types of problems for example medical diagnosis problems, more has to be said about the specificity and sensitivity as well.

In most cases machine learning problems also tend to be limited in definition. A typical problem describes a set of data whereupon an algorithm is built that can process the data to achieve some task e.g. classification or clustering. Real world problems however tend to span the whole Knowledge Discovery process (KDD) including business understanding, data understanding, data preparation, modeling, evaluation, and deployment. The machine learning part is but a component (albeit one could argue a significant one) in the solution to the problem. In this thesis we discuss machine learning algorithms and attempt to describe natural datasets and real problems where we implement these algorithms and in Chapters 9 and 10 we describe the development and deployment of a machine learning solution to real-world problems and some of the lessons learned from such a process.

1.4 Outline of thesis

This thesis is a two-part thesis. In the first part we discuss extensions of LVQ prototype-based classifiers that use information theoretic measures as distance measures. We also present work on the use of different representations of data, in this case histograms of images, in the same LVQ system. We show the possibility of formulating one combined distance measure for the heterogeneous dataset formed by a combination of their individual representative distance measures.

In the second part we delve into causal structure discovery and its application to real world problems. We also present a first attempt at leveraging some of the techniques of causal learning and applying them to feature relevance learning in LVQ. We also present some deployment examples of some of the techniques. The chapter wise breakdown is as follows.

In Chapter 2, we present some background material on some origins of LVQ – competitive learning; we show how LVQ and its variants derive from this. We also discuss some variants of LVQ whose relevance appears in future chapters. In this chapter, we also present different distance measures used to determine the similarity (dissimilarity) between prototypes and data points and argue a case for the use of divergences in situations where the data can be represented in a non-negative and potentially normalized form.

In Chapter 3, we discuss divergence-based classification in LVQ, a novel extension to the family of LVQ that uses divergences as a distance measure. We present the formalization of the method, discuss the choice of divergences and show some experiments on both artificial and natural datasets of the usage of this method. We also discuss the effect on performance of tuning certain parameters of the divergences.

In Chapter 4, we discuss a variant of DLVQ that combines different representations of the data (histograms) and show how to train the combined distance measure with a matrix that represents the distance-wise correlations of the different data representations. We discuss different ways of combining the sub-distances and show the superiority of a matrix-based combination over a linear combination of the sub-distances. We also present results from the application of this combined distance formulation on artificial and natural datasets.

In Chapter 5, we change our focus to the field of causal structure learning. In this chapter we present a treatise of the field of causal structure learning from purely observational data and try to explain the foundational concepts on which the Chapters 6, 7, 8 and 9 depend. We explain the assumptions that are made for any realistic causal structure discovery to be done and explain why these assumptions are important.

Chapter 6 presents a novel algorithm called Causal-RLVQ that extends Relevance LVQ (RLVQ), a member of the family of LVQ that outputs the relevance of each feature for the classification. This novel method extends RLVQ by redefining the relevance to describe the causal relationship between the features and the variable to be predicted. This is a first attempt at marrying the two disciplines and as such we present results of the method on artificial datasets and discuss possible improvements to the method.

Chapter 7 delves into the use of a committee of weak structure learners to do causal structure discovery from observational data. In here we present a novel causal discovery algorithm Expected Partial Correlation (EPC) that uses partial correlation in the discovery of causal relationships between variables. We also present an ensemble/committee method for causal discovery that uses a committee of differently oriented causal algorithms to vote on the causal relationship between variables. The chapter is concluded with some experiments and results of a competition where this method gave superior performance.

Chapter 8 looks at one of the core aspects of causal discovery: independence testing. This chapter describes some work we did on combining several independence tests in to one global independence test that takes into consideration the meta properties of the data. We illustrate how this method offers superior performance (compared to other independence tests) when applied on several natural datasets.

In Chapter 9, we present an application of causal learning to address the problem of predicting the state of food insecurity of a developing-world household in Uganda. We use data from over 5000 households and apply the committee causal discovery technique from Chapter 7 to develop a causal graph relating the different socio-economic variables to food insecurity. Causally relevant variables from the graph are used in a prediction algorithm as well and we show that they give a comparative performance.

In Chapter 10, we report on the use of machine learning techniques for the development and deployment of a crop disease surveillance system. This chapter fills in most of the practical and social details related to more technical and scientific details about the collection of the crop image data used in the testing of some of the methods in Chapter 3 and 4. It further explains in a holistic manner, how the system was actually implemented at the server-side and at the front-end, what kind of devices were used and why, and some practical field considerations or lessons for the broader machine learning community.

Chapter 11 presents a concise summary of the thesis highlighting the major contributions in the thesis, some additional contextual information on the experiments and a look at what future work could spawn from this thesis.

Chapter 2

Competitive Learning

Abstract

This chapter serves as a general introduction to concepts of competitive learning specifically prototype-based learning. We introduce the LVQ family, the quintessential prototype-based learning algorithms, the classical LVQ, and variants there of. We also introduce the concept of distance measures as used in prototype learning to determine the similarity or dissimilarity between prototypes and data points. We conclude with a discussion on the use of divergences as distance measures. Divergences can be used in situations where the data can be represented in a non-negative and potentially normalized form. We highlight some important divergences and formalize their representation as distances.

2.1 Introduction

Previously in Chapter 1 we discussed a classification mechanism based on prototypes. Prototype-based classification can be viewed as a simplification of competitive learning (Rumelhart and Zipser 1985, Grossberg 1976). In competitive learning the basic precept is that representative data points or prototypes defined in the space of the data compete for the available data points. A winning prototype is defined for each data point as the prototype *closest* to it. The interpretation of *closest* is based on some dissimilarity measure. We discuss several measures in Section 2.3. In competitive learning, the *closest* prototype to a data point $\mathbf{x}$ also called the winning prototype, $\mathbf{w}^L$ is updated or moved towards the data point. This movement or update is specified by Eq. (2.1).

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \Delta\mathbf{w}_t, \tag{2.1}$$

$$\Delta\mathbf{w}_t = \begin{cases} \eta(\mathbf{x} - \mathbf{w}^L) & \text{for } \mathbf{w}^L \text{ with } (\mathbf{w}^L - \mathbf{x})^2 \leq (\mathbf{w}^j - \mathbf{x})^2 \text{ for all } j = 1, \dots, M, \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

where $\mathbf{x} \in I\!R^N$ are the data points and $\mathbf{w} \in I\!R^M$ are the prototypes with $\mathbf{w}^j \in I\!R^N$ $(j = 1, \dots, M)$. The parameter η determines the step size of the update.

Competitive learning can be applied for unsupervised as well as supervised learning. Equations (2.1) and (2.2) form the update equations for an unsupervised learning system that results in clusters of points *close* to each other. As we have already determined to look at classification schemes we will investigate supervised competitive learning schemes where the winning prototype has to be of the same class as the data point. LVQ and its variants are forms of supervised competitive learning.

2.2 Learning Vector Quantization and its variants

Learning Vector Quantization (LVQ) is a supervised classification scheme that was introduced by (Kohonen 1986). LVQ and its variants are a family of classifiers that have found much popularity within the machine learning field. This is mainly because they are relatively easy to implement and understand. Interpreting the results is fairly intuitive as well since the optimized prototypes, the result of the classification training process, are in the space of the data. LVQ based algorithms can also be scaled to handle multi-class problems without distinctively increasing the complexity of the system.

A similar algorithm to the LVQ family of algorithms is the k-Nearest Neighbour algorithm (Cover and Hart 1967) which skips the training phase and directly uses all the available training data in the classification of a new data point. To classify a new data point, k of its closest neighbours are obtained using some measure of closeness for example Euclidean distance, and the average class of these k neighbours is awarded to the new data point. The complexity of the system, however, grows with the number of training data points N since the distance to each point needs to be calculated every time. In contrast, LVQ systems only calculate the distance to the prototypes, a complexity that can be controlled by the number of prototypes one defines for the system.

An LVQ system can generally be defined as follows. For a particular classification task, we assume that a set of labelled example data is available:

$$\{\mathbf{x}^\mu, y^\mu\}_{\mu=1}^{P},$$

where the $\mathbf{x}^\mu \in I\!\!R^N$ are feature vectors and the labels $y^\mu \in \{1, 2, \ldots C\}$ specify their class membership.

In an LVQ system we denote by $W = \left\{(\mathbf{w}^j, c(\mathbf{w}^j)\right\}_{j=1}^{M}$ a set of M prototype vectors $\mathbf{w}^j \in I\!\!R^N$ which carry labels $c(\mathbf{w}^j) \in \{1, 2, \ldots C\}$. One or several prototypes can be assigned to each class. Prototype vectors are identified in feature space and serve as typical representatives of their classes. Together with a given distance measure $d(\mathbf{x}, \mathbf{w})$, they parametrize the classification scheme. Most frequently, a *Winner-Takes-All* scheme is applied; an arbitrary input $\mathbf{x}$ is assigned to the class $c(\mathbf{w}^L)$ of the closest prototype with $d(\mathbf{x}, \mathbf{w}^L) \leq d(\mathbf{x}, \mathbf{w}^j)$ for all j.

The purpose of training is the computation of suitable prototype vectors based on the available example data. The ultimate goal is generalization; the successful application of the classifier to novel, unseen data. LVQ training can follow heuristic ideas as in Kohonen's original LVQ1 (Kohonen 1986) or follow a variety of modifications to LVQ1 that have been suggested in the literature, aiming at better convergence or favourable generalization behaviour. A prominent and appealing example is the cost function based Generalized Learning Vector Quantization (GLVQ) (Sato and Yamada 1996). We discuss this and other variants of LVQ in Sections 2.2.1, 2.2.2 and 2.2.3.

2.2.1 Classical LVQ

The classical LVQ also called LVQ1 is a heuristic algorithm introduced by Kohonen that updates prototypes based on how close they are to a presented data point given the class of the prototype and that of the data point. The training process is represented by Algorithm 1.

Algorithm 1: LVQ1

```
Input : Training data: $\{\mathbf{x}^\mu, y^\mu\}_{\mu=1}^{P}$, and prototypes $W = \left\{(\mathbf{w}^j, c(\mathbf{w}^j))\right\}_{j=1}^{M}$.
foreach training data example : $(\mathbf{x}, y)$ do
    Determine the winning prototype $\mathbf{w}^L$ with $d(\mathbf{w}^L, \mathbf{x}) \leq d(\mathbf{w}^j, \mathbf{x})$
                                  for all $j = 1, \ldots, M$.
    Update $\mathbf{w}^L$ according to:
        $\mathbf{w}^L \leftarrow \mathbf{w}^L + \eta(\mathbf{x} - \mathbf{w}^L)$,      if $c(\mathbf{w}^L) = y$,
        $\mathbf{w}^L \leftarrow \mathbf{w}^L - \eta(\mathbf{x} - \mathbf{w}^L)$,      if $c(\mathbf{w}^L) \neq y$.
end
Return: $\mathbf{w}^L$, the trained prototype.
```

To improve generalization of this heuristic algorithm, further modifications to it were made by Kohonen resulting in the Optimized learning-rate LVQ (OLVQ) and LVQ2.1 (Kohonen 1990, Kohonen et al. 2001). In LVQ2.1, not only one but two prototypes are updated at each training step subject to some window-rule that controls the diverging of the prototypes.

2.2.2 Generalized LVQ

The Generalized LVQ (GLVQ) algorithm is a variant of LVQ introduced by Sato and Yamada (Sato and Yamada 1996) that incorporates an objective (cost) function in the LVQ system. The objective function can be perceived as a function that approximates the classification error; in which case we want to minimize it, or it could be perceived as a function that describes the classification accuracy; in which case we want to maximize it. Either way the advantage of an objective function based LVQ system is that you can use gradient methods (descent or ascent, online or batch) to optimize it. The GLVQ cost function can be stated in the following form

$$E(W) = \sum_{\mu=1}^{P} \Phi\left(\frac{d(\mathbf{x}^\mu, \mathbf{w}^J) - d(\mathbf{x}^\mu, \mathbf{w}^K)}{d(\mathbf{x}^\mu, \mathbf{w}^J) + d(\mathbf{x}^\mu, \mathbf{w}^K)}\right), \tag{2.3}$$

where $\mathbf{w}^J$ denotes the closest correct prototype with $c(\mathbf{w}^J) = y^\mu$ and $\mathbf{w}^K$ is the closest incorrect prototype ($c(\mathbf{w}^K) \neq y^\mu$).

Note that the argument of Φ in Eq. (2.3) is restricted to the interval $[-1, +1]$. Φ is a monotonically increasing function, e.g. a sigmoidal function or a logistic function $\Phi(x) = 1/(1 + exp(-x))$ or the identity $\Phi(x) = x$ in the simplest case. The function Φ generally determines the active region of the algorithm.

In principle, a variety of numerical optimization procedures are available for the minimization of the cost function (2.3). On-line training using stochastic gradient descent is one particularly simple method which has proven useful in many practical applications.

In stochastic gradient descent, a single randomly selected example $\mathbf{x}$ is presented and the corresponding winners $\mathbf{w}^J, \mathbf{w}^K$ are updated incrementally by

$$\Delta\mathbf{w}^J = \frac{-\eta\ d_K(\mathbf{x})}{(d_J(\mathbf{x}) + d_K(\mathbf{x}))^2}\ \frac{\partial}{\partial \mathbf{w}^J}\, d_J(\mathbf{x})\,, \tag{2.4}$$

$$\Delta\mathbf{w}^K = \frac{+\eta\ d_J(\mathbf{x})}{(d_J(\mathbf{x}) + d_K(\mathbf{x}))^2}\ \frac{\partial}{\partial \mathbf{w}^K} d_K(\mathbf{x}), \tag{2.5}$$

where $d_L(\mathbf{x}) = d(\mathbf{x}, \mathbf{w}^L)$ and $\partial/\partial\mathbf{w}^L$ denotes the gradient with respect to $\mathbf{w}^L$. In this case $L \in \{J, K\}$. The learning rate η controls the step size of the algorithm. Training is performed by running the system through several epochs; in each epoch, all examples in the training data are presented in randomized order.

Practical prescriptions are obtained by inserting a specific distance measure $d(\mathbf{x}, \mathbf{w})$ and its gradient. Section 2.3 highlights some of these distance measures. In general, meaningful dissimilarities should satisfy these conditions: $d(\mathbf{x}, \mathbf{w}) \geq 0$ for all $\mathbf{x}, \mathbf{w}$ and $d(\mathbf{x}, \mathbf{w}) = 0$ for $\mathbf{w} = \mathbf{x}$.

Algorithm 2: GMLVQ

```
Input : Training data: $\{\mathbf{x}^\mu, y^\mu\}_{\mu=1}^{P}$, and prototypes $W = \{(\mathbf{w}^j, c(\mathbf{w}^j))\}_{j=1}^{M}$.
foreach training data example : $(\mathbf{x}, y)$ do
    Determine the closest correct prototype, $\mathbf{w}^J$ with $c(\mathbf{w}^J) = y$, and
        $d(\mathbf{w}^J, \mathbf{x}) \leq d(\mathbf{w}^j, \mathbf{x})$, for all $\mathbf{w}^j$, with $c(\mathbf{w}^j) = y$,
    ..and the closest incorrect prototype, $\mathbf{w}^K$ with $c(\mathbf{w}^K) \neq y$, and
        $d(\mathbf{w}^K, \mathbf{x}) \leq d(\mathbf{w}^j, \mathbf{x})$, for all $\mathbf{w}^j$, with $c(\mathbf{w}^j) \neq y$.
    Update $\mathbf{w}$ according to:
        $\mathbf{w}^J \leftarrow \mathbf{w}^J + \Lambda \cdot \Delta\mathbf{w}^J$    $( y = c(\mathbf{w}^J) ).$
        $\mathbf{w}^K \leftarrow \mathbf{w}^K - \Lambda \cdot \Delta\mathbf{w}^K$    $( y \neq c(\mathbf{w}^K) ).$
    Update $\Omega$ according to:
        $\Omega \leftarrow \Omega + \Delta\Omega$    ( if $y = c(\mathbf{w}^J)$ ).
        $\Omega \leftarrow \Omega - \Delta\Omega$    ( if $y \neq c(\mathbf{w}^K)$ ).
    Normalize $\Omega$ such that Tr(...) = 1 for $\Lambda = \Omega^\top\Omega$.
end
Return: $\mathbf{w}^L$, $\mathbf{w}^K$, the trained prototypes.
```

2.2.3 Generalized Matrix LVQ

Generalized Matrix Learning Vector Quantization (GMLVQ) (Schneider et al. 2009a) is an extension of GLVQ where a matrix $\Lambda = \Omega^\top\Omega$ that captures the correlations between different data dimensions in the distance measure is added to the cost-function based GLVQ scheme. With squared Euclidean distance as the distance measure, the distance $d_\Omega(\mathbf{x}, \mathbf{w})$ is formulated as

$$d_\Omega(\mathbf{x}, \mathbf{w}) = (\mathbf{x} - \mathbf{w})\, \Lambda\, (\mathbf{x} - \mathbf{w})^\top. \tag{2.6}$$

An extra partial derivative of the cost function, Eq. (2.3), with respect to the matrix Ω is calculated to further optimize Ω with respect to the classification. Omega is updated incrementally together with the prototypes for every training step as follows

$$\Delta\Omega = -\psi \cdot \left(\frac{2d_K(\mathbf{x})}{(d_J(\mathbf{x}) + d_K(\mathbf{x}))^2} \frac{\partial}{\partial\Omega} d_J(\mathbf{x}) \; - \; \frac{2d_J(\mathbf{x})}{(d_J(\mathbf{x}) + d_K(\mathbf{x}))^2} \frac{\partial}{\partial\Omega} d_K(\mathbf{x}) \right). \tag{2.7}$$

The learning rate ψ controls the step size of the Ω update. The prototype updates in this case follow Algorithm 2. Normalization is done after every learning step to prevent numerical problems.

We highlight the LVQ variant GMLVQ here because a similar approach is used in Chapter 4 to combine different distance measures using a matrix into one global distance measure. A similar formalism to GMLVQ is employed albeit with entirely different meaning.

2.3 Dissimilarity metrics and measures

The gist of competitive learning or more specifically prototype-based classification schemes is essentially that some measure of *closeness* or similarity or dissimilarity has to be determined to quantify how *close* (apart) the prototype is from the data point. The goal being to adapt the prototypes by whatever scheme based on the relative distances of the data points until a point of convergence in the data space is arrived at.

2.3.1 Metrics and measures

Dissimilarity measures or distance measures are principally used to quantify this *closeness* between the prototype and the data point. In the literature several authors have used *measures* and *metrics* interchangeably especially when dealing with what would technically be called *metrics* only. For this work however, we make a clear distinction between dissimilarity/similarity metrics and dissimilarity measures. A metric in general terms adheres to the triangular inequality and is symmetric while a distance measure does not necessarily adhere to these properties. A formal definition of a distance metric is as follows.

1. *Definition. Metric*
A function $d : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$, where $\mathcal{L}$ is an arbitrary set, is called a distance metric or simply a metric (on $\mathcal{L}$) iff it satisfies the following three conditions $\forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{L}$:
(i) $d(\boldsymbol{x}, \boldsymbol{y}) = 0$, *iff $\boldsymbol{x} = \boldsymbol{y}$,*
(ii) $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$ (symmetry),
(iii) $d(\boldsymbol{x}, \boldsymbol{z}) \leq d(\boldsymbol{x}, \boldsymbol{y}) + d(\boldsymbol{y}, \boldsymbol{z})$ (triangle inequality).

For our purposes in this thesis we extend normal distance metrics to the space of divergences and this violates the pristine metric conditions as stated in Definition 1. Particularly we violate conditions (ii) and (iii) and for condition (i) we take the special case where the condition $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ but $\boldsymbol{x} \neq \boldsymbol{y}$; the *pseudometric* case. Condition (i) is also violated when dealing with relevances e.g. in GRLVQ – when

Λ is semi-definite, $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ for $\boldsymbol{x} \neq \boldsymbol{y}$. For these reasons we will use the general term distance measure or measure for the most part to refer to any function that can be used to evaluate the dissimilarity $d(\boldsymbol{x}, \boldsymbol{y})$.

2.3.2 Minkowski metrics

Minkowski metrics form the most common type of metrics used in both classification and clustering. The Minkowski family of measures can be defined as follows.

$$d_k(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=m}^{m} |x_i - y_i|^k \right)^{\frac{1}{k}}, \tag{2.8}$$

where $k \geq 1$ is a parameter of the distance measure. It contains the following special cases:

(i) $k = 1$: Manhattan or city block distance,

(ii) $k = 2$: Euclidean distance,

(iii) $k \to \infty$: Maximum distance.

The Euclidean measure is one of the best-known and most commonly used distance measures. An important advantage of the Euclidean metric is that it is invariant w.r.t. orthogonal linear transformations (translation, rotation, reflection). While all distance measures from the Minkowski family are invariant w.r.t. translation, only the Euclidean distance is invariant w.r.t. rotation and (arbitrary) reflection. However the Euclidean distance, as well as other distance measures from the Minkowski family, are not scale-invariant.

2.3.3 Divergences as distance measures

Divergences define functions that can be used to quantify the *distance* between probability distributions or, more loosely, any positive quantities which could be normalized or not. Typical examples of representative data are spectral data, histograms or any temporal or functional data associated with distributions. More formally a divergence can be defined as follows.

2. *Definition*. *Divergence*
A function $D(.\|.) : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$, *where* $\mathcal{L}$ *represents the space of all probability distributions, is called a divergence on* $\mathcal{L}$ *iff it satisfies the following axioms* $\forall p, q \in \mathcal{L}$*:*

(i) $D(p\,||\,q) \geq 0,$ (non-negativity),
(ii) $D(p\,||\,p) = 0,$ (identity),
(iii) $D(p\,||\,q) = 0,$ (*iff* $p \equiv q$).

Divergences unlike Minkowski type distances do not necessarily satisfy triangular inequalities or symmetry. However, the LVQ framework does not require them to. In both training and testing phases, only distances between data and prototype vectors have to be calculated; the distances between two prototypes or two data vectors are never used. Divergences can thus be used in the LVQ framework in a straight forward way if used in a consistent form (same order of arguments).

Several divergences abound generally in the fields of mathematics and statistics. (Cichocki et al. 2009) give a concise categorization of the different classes of divergences. Owing to some general properties these are split into three groups: Bregman-divergences, γ-divergences and Csiszár's f-divergences.

A brief formalization of some of the divergences relevant to this thesis is given here. Because the data in the experiments we will use will be in the form of discrete vector representations, we present the divergences as discretized versions of their functional form.

Bregman divergences

The *eta* (η) and the *beta* (β) divergences are popular examples of the Bregman divergences.

(i) *Eta*-divergence

The η-divergence (Nielsen and Nock 2009) also known as norm-like divergence can be expressed as follows

$$D_\eta(\mathbf{x}\,||\,\mathbf{w}) = \sum_j \left(x_j^\eta + (\eta - 1) \cdot w_j^\eta - \eta \cdot x_j \cdot w_j^{\eta-1} \right). \tag{2.9}$$

For η = 2, the form of Eq. (2.9) reduces to that of Euclidean distance.

(ii) *Beta*-divergence

For two normalized vectors $\mathbf{x}$ and $\mathbf{w}$ representing a data point and a prototype, the β divergence is expressed as

$$D_\beta(\mathbf{x}\,||\,\mathbf{w}) = \sum_j \left(x_j \cdot \frac{x_j^{\beta-1} - w_j^{\beta-1}}{\beta - 1} \right) - \sum_j \frac{x_j^\beta - w_j^\beta}{\beta}. \tag{2.10}$$

The β-divergence also has a unique characteristic; as $\beta \to 1$, it approximates the well known Kullback-Leibler divergence (Kullback and Leibler 1951).

Csiszár's f-divergences

The Generalized Rényi-divergence (Amari and Nagaoka 2000) is the most popular of the Csiszár's f-divergences. It is expressed as follows

$$D_\alpha(\mathbf{x}\,||\,\mathbf{w}) \;=\; \frac{1}{\alpha-1}\;\log\left[1+\sum_j (x_j^\alpha \cdot w_j^{1-\alpha} - \alpha \cdot x_j + (\alpha-1)w_j\right]. \tag{2.11}$$

The Generalized Rényi-divergence too like the β-divergence approximates the Kullback-Leibler divergence for the special case when $\alpha \to 1$. The Generalized Rényi-divergence reduces to the Rényi divergence (Rényi 1970) when the quantities being compared are probability densities.

Gamma (γ) divergences

In this thesis we delve a lot deeper into the γ-divergence (Fujisawa and Eguchi 2008). The γ-divergence tends to be a more robust divergence with respect to outliers. The γ-divergences can generally be expressed as follows

$$D_\gamma(\mathbf{x}\,||\,\mathbf{w}) \;=\; \frac{1}{\gamma+1}\;\log\left[\left(\sum_j x_j^{\gamma+1}\right)^{1/\gamma}\cdot\left(\sum_j w_j^{\gamma+1}\right)\right] - \log\left[\left(\sum_j x_j\,w_j^\gamma\right)^{1/\gamma}\right]. \tag{2.12}$$

As $\gamma \to 0$ the γ divergence approximates the Kullback-Leibler divergence.

(i) Cauchy-Schwarz Divergence

$$D_{cs}(\mathbf{x}\,||\,\mathbf{w}) \;=\; \frac{1}{2}\;\log\left[\frac{\left(\sum_j x_j^2\right)^{1/2}\cdot\left(\sum_j w_j^2\right)^{1/2}}{\sum_j (x_j \cdot w_j)}\right]. \tag{2.13}$$

The Cauchy-Schwarz divergence (Jenssen et al. 2006) is a class of the γ divergences with $\gamma = 1$. The Cauchy-Schwarz divergence is a symmetric measure like Euclidean distance and is invariant under scalar multiplication with positive constants; $D_{cs}(\mathbf{x}\,||\,\mathbf{w}) = D_{cs}(c_1 \cdot \mathbf{x}\,||\,c_2 \cdot \mathbf{w})\,\forall\, c_1, c_2 > 0$. The Cauchy-Schwarz divergence has frequently been applied to Parzen window estimation and spectral clustering.

(ii) Generalized Kullback-Leibler divergence

The well known and well used Generalized Kullback-Leibler divergence can be expressed as follows

$$D_{kl}(\mathbf{x}\,||\,\mathbf{w}) = \sum_j \left(x_j \cdot \log \frac{x_j}{w_j} - (x_j - w_j) \right). \tag{2.14}$$

When the quantities are probability distributions, it reduces to the ordinary Kullback-Leibler divergence.

A detailed discussion of these and more examples of divergences from each class can be found here (Cichocki et al. 2009, Villmann et al. 2010, Villmann and Haase 2009). Also, more recent work that deals with the application of divergences in prototype based vector quantization can be found in (Haase 2014). In Chapter 3 we take a deeper look at the γ-divergence; it exhibits unique properties that make it a suitable candidate for an LVQ system because its parameter γ can be tuned across the whole spectrum of key divergence measures. We apply divergence-based LVQ schemes to spectral data and histogram data and compare performance as parameters of the divergences are tuned. Because these divergences are used in a cost-function based LVQ scheme, derivatives of these formalisations must exist. These derivatives are generally called Fréchet derivatives and the respective formalisations can be found in (Cichocki et al. 2009, Villmann et al. 2010, Villmann and Haase 2009).

Work based on:

E. Mwebaze, P. Schneider, F.-M. Schleif, J.R. Aduwo, J.A. Quinn, S. Haase, T. Villmann and M. Biehl – *"Divergence based classification in Learning Vector Quantization"* Neurocomputing, pp. 1429–1435, 2011.

E. Mwebaze, P. Schneider, F.M. Schleif, S. Haase, T. Villmann, M. Biehl –*"Divergence based Learning Vector Quantization"*, in European Symposium on Artificial Neural Networks (ESANN), M. Verleysen(ed.), d-side publishing, 247-252, 2010.

Chapter 3

Divergence based Classification in LVQ

Abstract

In this Chapter, we discuss the use of divergences in dissimilarity based classification. Divergences can be employed whenever vectorial data consists of non-negative, potentially normalized features. This is, for instance, the case in spectral data or histograms. In particular, we introduce and study Divergence Based Learning Vector Quantization (DLVQ). We derive cost function based DLVQ schemes for the family of γ*-divergences which includes the well-known Kullback-Leibler divergence and the Cauchy-Schwarz divergence as special cases. The corresponding training schemes are applied to three different real world data sets. The first one, a benchmark data set (Wisconsin Breast Cancer) is available in the public domain. The second one is mass spectra lung cancer diagnosis data while in the third, color histograms of leaf images are used to detect the presence of Cassava Mosaic Disease (CMD) in cassava plants. We compare the use of standard Euclidean distances with DLVQ for different parameter settings. We show that DLVQ can yield superior classification accuracies and Receiver Operating Characteristics.*

3.1 Introduction

Distance based classification schemes can be implemented efficiently in the framework of the popular Learning Vector Quantization (LVQ). LVQ systems are flexible, easy to implement, and can be applied in multi-class problems in a straightforward fashion. These and several other advantages have been expounded on in Chapter 2. It suffices to mention however, that LVQ classifiers are widely used in a variety of areas including image processing tasks, medical applications, control of technical processes, or bioinformatics. An extensive bibliography including applications can be found in (NNRC 2002).

A key step in the design of any LVQ system is the choice of an appropriate distance measure. Most frequently, practical prescriptions make use of Euclidean metrics or more general Minkowski measures, without further justification. In Section 2.2 we have highlighted some of these distance measures and motivated the use of divergences as distance measures.

Here, we take a deeper look at the use of divergences as distance measures. Divergences, it turns out, can be employed as distances in supervised or unsupervised vector quantization, provided the feature vectors and prototypes consist of non-negative, potentially normalized components.

Information theoretic distance measures have been discussed in the context of various machine learning frameworks, previously. This includes prototype based clustering, classification, or dimension reduction, see (Jang et al. 2008, Banerjee et al. 2005, Villmann et al. 2008, Torrkola 2003, Bunte et al. 2010) for just a few recent examples. Frequently, divergences are employed to quantify the similarity of the prototype density with the observed distribution of data. Note that, here, we use divergences to quantify directly the distance between individual data points and prototype vectors. Moreover, we derive gradient based update schemes which exploit the differentiability of the divergences.

After setting up the general framework in Section 3.2, we present the family of so-called gamma (γ)-divergences as a specific example. The family of γ-divergences is specified by choice of a parameter γ and includes the well-known Kullback-Leibler and the so-called Cauchy-Schwarz divergence as special cases.

We develop the corresponding divergence based LVQ (DLVQ) schemes and apply them to three different classification problems. First, the Wisconsin Breast Cancer data set from the UCI data repository (Asuncion et al. 1998). Then mass spectra lung cancer data from the medical domain and thirdly, a data set that relates to the identification of the Casssava Mosaic Disease based on color histograms representing leaf images. Performance is evaluated in terms of Receiver Operator Characteristics and compared with the standard LVQ scheme using Euclidean distance. We also look at the influence of the parameter γ on the classification performance.

3.2 Divergence based Learning Vector Quantization

We have already presented the cost-function based LVQ, Generalized LVQ (GLVQ), an improvement to the classic LVQ, which while still being heuristic, achieves better convergence or favorable generalization behavior. For completeness we present the GLVQ framework introduced in Chapter 2, again here.

Consider a set $\{\mathbf{x}^\mu, y^\mu\}_{\mu=1}^P$ of example data. Here $\mathbf{x}^\mu \in I\!R^N$ and the labels $y^\mu \in$

$\{1, 2, \ldots C\}$ correspond to one of the classes. $W = \{(\mathbf{w}^j, c(\mathbf{w}^j)\}_{j=1}^{M}$ comprises a number M of N-dim. prototype vectors $\mathbf{w}^j$ which carry labels $c(\mathbf{w}^j) \in \{1, 2, \ldots C\}$. Given a distance measure $d(\mathbf{x}, \mathbf{w})$, the LVQ classifier assigns an arbitrary input $\mathbf{x}$ to the class $c(\mathbf{w}^L)$ of the closest prototype with $d(\mathbf{x}, \mathbf{w}^L) \leq d(\mathbf{x}, \mathbf{w}^j)\ \forall j$.

GLVQ training is guided by the optimization of a cost function of the form

$$E(W) = \sum_{\mu=1}^{P} \Phi \left(\frac{d(\mathbf{x}^\mu, \mathbf{w}^J) - d(\mathbf{x}^\mu, \mathbf{w}^K)}{d(\mathbf{x}^\mu, \mathbf{w}^J) + d(\mathbf{x}^\mu, \mathbf{w}^K)} \right), \tag{3.1}$$

where $\mathbf{w}^J$ denotes the closest correct prototype with $c(\mathbf{w}^J) = y^\mu$ and $\mathbf{w}^K$ is the closest incorrect prototype ($c(\mathbf{w}^K) \neq y^\mu$). We consider here the simple case where $\Phi(x) = x$.

In stochastic gradient descent, the randomly selected example $\mathbf{x}$ is presented and the corresponding winners $\mathbf{w}^J, \mathbf{w}^K$ are updated incrementally by

$$\Delta \mathbf{w}^J = \frac{-\eta\ d_K(\mathbf{x})}{(d_J(\mathbf{x}) + d_K(\mathbf{x}))^2}\ \nabla_J d_J(\mathbf{x})\,,\ \Delta \mathbf{w}^K = \frac{+\eta\ d_J(\mathbf{x})}{(d_J(\mathbf{x}) + d_K(\mathbf{x}))^2}\ \nabla_K d_K(\mathbf{x}), \tag{3.2}$$

where $d_L(\mathbf{x}) = d(\mathbf{x}, \mathbf{w}^L)$ and ∇_L denotes the gradient with respect to $\mathbf{w}^L$ and $L \in \{J, K\}$. The so-called learning rate η controls the step size of the algorithm.

In the following we assume that the data consists of vectors of non-negative components $x_j \geq 0$ which are normalized to $\sum_{j=1}^{N} x_j = 1$. Potential extensions to non-normalized positive data are discussed towards the end of this chapter.

Normalized non-negative components, x_j, can be interpreted as probabilities. This interpretation may be just formal, as for instance in the case of the first example data set (WBC) where we normalize the data before applying these techniques to it. In other cases, the probabilistic interpretation appears naturally, for instance, whenever the vectors $\mathbf{x}$ represent histograms or spectra. An important example for the former is the characterization of images by normalized gray value or color histograms, which we also consider in our third example data set. Frequently, spectral data is conveniently normalized to constant total intensity and is employed for classification in a large variety of fields including remote sensing or bioinformatics (NNRC 2002). Assuming normalized non-negative data suggests, of course, the consideration of prototype vectors which satisfy the same constraints. Hence, we enforce $w_j \geq 0$ and $\sum_{j=1}^{N} w_j = 1$ explicitly after each training step.

Under the above assumptions, information theory provides a multitude of potentially useful dissimilarity measures. Different classes of divergences and their mathematical properties have been detailed in Chapter 2, a more detailed discussion can be found in (Villmann and Haase 2009, Villmann et al. 2010).

Because GLVQ employs gradients in the optimization, the choice of a distance measure has to be differentiable. The distance measures that we looked at, and their derivatives are presented here.

a) Squared Euclidean distance

$$d_{eu}(\mathbf{x},\mathbf{w}) = \frac{1}{2}(\mathbf{x}-\mathbf{w})^2, \qquad \frac{\partial d_{eu}(\mathbf{x},\mathbf{w})}{\partial w_k} = -(x_k - w_k). \tag{3.3}$$

b) Cauchy-Schwarz divergence (as introduced in (Principe et al. 2000)):

$$d_{cs}(\mathbf{x},\mathbf{w}) = \frac{1}{2}\log\left[\mathbf{x}^2\mathbf{w}^2\right] - \log \mathbf{x}^T\mathbf{w}, \quad \frac{\partial d_{cs}(\mathbf{x},\mathbf{w})}{\partial w_k} = \frac{w_k}{\mathbf{w}^2} - \frac{x_k}{\mathbf{x}^T\mathbf{w}}. \tag{3.4}$$

It might be interesting to note that, apart from the logarithm, the CS-measure is formally identical to the Pearson correlation (Strickert et al. 2006) in the case of zero mean data. Pearson correlation has also been used in the context of LVQ, see (Strickert et al. 2006). However, it is important to note that the CS-divergence is applied only to non-negative and, consequently, non-zero mean data.

3.2.1 Influence of parameter γ in γ-divergences

The CS-divergence is part of the general family of gamma(γ)-divergences (Chapter 2). Again for completeness, the γ-divergence distance measure can be expressed as

$$D_\gamma(\mathbf{y},\mathbf{z}) = \frac{1}{\gamma+1}\log\left[\left(\sum_j y_j^{\gamma+1}\right)^{\frac{1}{\gamma}} \cdot \left(\sum_j z_j^{\gamma+1}\right)\right] - \log\left[\left(\sum_j y_j\, z_j^{\gamma}\right)^{\frac{1}{\gamma}}\right], \tag{3.5}$$

for $\mathbf{y},\mathbf{z} \in I\!R^N$.

The precise form of this dissimilarity measure is controlled by the parameter $\gamma > 0$. Note that for $\gamma = 1$, we obtain the symmetric Cauchy-Schwarz divergence as a special case, while the limit $\gamma \to 0$ yields the popular Kullback-Leibler divergence.

In general, for $\gamma \neq 1$, $d_\gamma(\mathbf{x},\mathbf{y}) \neq d_\gamma(\mathbf{y},\mathbf{x})$. The asymmetry is also reflected in the derivatives with respect to the first or second argument, respectively. Consequently, the use of $d(\mathbf{x},\mathbf{w}) = D_\gamma(\mathbf{x},\mathbf{w})$ yields a DLVQ scheme different from the one derived for $d(\mathbf{x},\mathbf{w}) = D_\gamma(\mathbf{w},\mathbf{x})$.

In the following we consider only training processes which make consistent use of either $D_\gamma(\mathbf{w},\mathbf{x})$ or $D_\gamma(\mathbf{x},\mathbf{w})$, respectively. One and the same measure is employed throughout the entire training process and when evaluating the classification performance. Accordingly, one of the following derivatives has to be inserted into Eq. (3.2) to yield the actual DLVQ prescription:

$$\frac{\partial\, D_\gamma(\mathbf{w},\mathbf{x})}{\partial\, w_j} = \frac{1}{\gamma}\,\frac{w_j^\gamma}{\sum_k\, w_k^{\gamma+1}} - \frac{1}{\gamma}\,\frac{x_j^\gamma}{\sum_k\, w_k\, x_k^\gamma}\,, \tag{3.6}$$

$$\frac{\partial\, D_\gamma(\mathbf{x},\mathbf{w})}{\partial\, w_j} = \frac{w_j^\gamma}{\sum_k\, w_k^{\gamma+1}} - \frac{x_j\, w_j^{\gamma-1}}{\sum_k\, x_k\, w_k^\gamma}\,. \tag{3.7}$$

Note that γ-divergences with $\gamma > 0$ are invariant under rescaling of the arguments; $D_\gamma(\lambda\,\mathbf{y},\mu\,\mathbf{z}) = D_\gamma(\mathbf{y},\mathbf{z})$ for $\lambda,\mu > 0$ (Villmann and Haase 2009, Villmann et al. 2010). Hence, the normalization of the feature vectors, $\sum_j x_j = 1$, is not required in the formalism and has no effect on the results presented here. This invariance does not hold for more general dissimilarities, as discussed in (Villmann and Haase 2009).

3.3 Data sets and classification problems

We used three different real world data sets to evaluate the DLVQ algorithm.

3.3.1 Wisconsin Breast Cancer (WBC) data

We first apply DLVQ to a popular benchmark problem: the Wisconsin Breast Cancer (original) data set (WBC) from the UCI data repository (Asuncion et al. 1998). Disregarding 16 vectors which contain missing values, the WBC set provides 683 examples in 9 dimensions. The data contains labels corresponding to *malignant* (239 examples) and *benign* samples (444 examples). Single features correspond to different score values between 1 and 10, see (Asuncion et al. 1998) for their definition. This does not imply a natural interpretation of feature vectors as histograms or probabilities. However, the application of the divergence based formalism is possible and it is justified to the same degree as the more popular choice of Euclidean distances. For a more detailed description of this data set and further references we refer the reader to (Asuncion et al. 1998) and (Bennett and Mangasarian 1992). We apply a normalization such that $\sum_j x_j = 1$ for the following analysis.

3.3.2 Lung cancer data

The Lung Cancer (LC) data set contains 100 mass spectra with 22304 features each. It has been down-sampled to 3696 features with no significant loss of information.

The data are provided with label information for two classes with 50 examples each, labeled as *cancer* and *control*, respectively. Details about generation and preprocessing of the data can be found in (Boelke et al. 2005) as well as in (Schleif et al. 2008).

3.3.3 Cassava Mosaic Disease (CMD) data

The third data set corresponds to features extracted from leaf images of cassava plants as provided by the National Crops Resources Research Institute in Uganda. Sample images represent 92 healthy plants and 101 plants infected with the cassava mosaic disease. For example images and further details of the image acquisition see Chapter 10.

Standard image processing techniques were employed to remove background and clutter and in order to obtain a set of characteristic features from the leaf images. When aiming at optimal classification performance, various sets of features may be taken into account. Here we limit the analysis to the aspect of discolorization caused by the disease. For the application of DLVQ we consider normalized histograms with 50 bins representing the distribution of hue values in the corresponding image. The application of DLVQ appears natural in this problem, as the information is represented by normalized histograms reflecting the statistical properties of the data.

3.4 Computer experiments

For the following evaluation and comparison of algorithms, we split the available data randomly into training (90% of the data) and test set (10%). If not stated otherwise, all results reported in the following were obtained as averages over 25 randomized splits. In both cases, we consider the simplest possible LVQ system with one prototype per class, only. Their initial prototype positions are obtained as the mean of 50% randomly selected examples from each class in the respective training set.

The choice of the learning rate is dependent on properties of the data set and on the distance measure in use. In order to facilitate a fair comparison, we determined a close to optimal learning rate from preliminary runs with respect to the achieved accuracies after a fixed number of training epochs. For each algorithm we selected the best learning rate from the set $\left\{10^{-3}, 10^{-4}, \ldots, 10^{-12}\right\}$.

Results presented in the following are obtained after 200 training epochs for the WBC data and after 1500 epochs with CMD data. The learning rates employed for the WBC data set were $\eta = 10^{-4}$ when using Euclidean distances and $\eta = 10^{-6}$ for γ-divergences. In the CMD data set, learning rates of $\eta = 10^{-5}$ (Euclidean measure) and $\eta = 10^{-6}$ (γ-divergences) have been used. For the LC data set we employ $\eta = 2.5 \times 10^{-3}$ for both the Euclidean and Cauchy-Schwarz measures.

After training we determine training and test set accuracies of the classifiers and we report the average values obtained over the validation runs. When comparing classifiers, it is important to take into account that greater overall test accuracies do not necessarily indicate *better* performance. The Winner-Takes-All LVQ classifier represents just one working point, i.e. one combination of class 1 error and class 2 error. In particular, for unbalanced data sets a more detailed evaluation of the classification performance is instrumental.

In order to obtain further insight, we introduce a bias θ to the classification rule after training; an input vector $\mathbf{x}$ is assigned to class 1 if

$$d(\mathbf{x}, \mathbf{w}_1) < d(\mathbf{x}, \mathbf{w}_2) + \theta, \tag{3.8}$$

where $\mathbf{w}_i$ is the closest prototype representing class i. The bias of the resulting classification towards one of the classes depends on the sign and magnitude of the threshold. By varying θ, the full Receiver Operating Characteristics (ROC) of the classifier can be obtained. An ROC curve displays the true positive rate (sensitivity) as a function of the false positive rate (1 - specificity) (Duda et al. 2000, Fawcett 2006).

3.5 Results

Results presented in Table 3.1 display a threshold-average over the validation runs (Fawcett 2006). In the ROC, *false positive rates* correspond to the fraction of truly benign cases (WBC data) or healthy plants (CMD data), or non cancer patients (LC data) that are misclassified. Correspondingly, the *true positive rate* gives the fraction of truly malignant (WBC data) or diseased plants (CMD data), or patients with cancer (LC data) that are correctly classified. As an important and frequently employed measure of performance we also determine the corresponding area under curve (AUC) with respect to training set and test set performance.

3.5.1 Euclidean distance and Cauchy-Schwarz divergence

We first compare the two symmetric distance measures discussed here: standard Euclidean metrics and the Cauchy-Schwarz divergence. Figure 3.1 displays the ROC with respect to test set performances in the WBC benchmark problem (left panel), the CMD data set (middle panel) and LC data set (right panel).

Table 3.1 summarizes numerical findings in terms of the observed training and test accuracies for the unbiased LVQ classifier with $\theta = 0$ and the AUC of the Receiver Operator Characteristics.

WBC	training acc.	test acc.	AUC (training)	AUC (test)
$D_{eu}(\mathbf{x}, \mathbf{w})$	0.850 (0.040)	0.845 (0.041)	***0.924 (0.004)***	***0.918 (0.004)***
$D_{cs}(\mathbf{x}, \mathbf{w})$	0.864 (0.003)	0.853 (0.007)	0.923 (0.005)	0.916 (0.005)

CMD	training acc.	test acc.	AUC (training)	AUC (test)
$D_{eu}(\mathbf{x}, \mathbf{w})$	0.790 (0.005)	0.782 (0.007)	0.856 (0.006)	0.848 (0.007)
$D_{cs}(\mathbf{x}, \mathbf{w})$	0.807 (0.002)	0.805 (0.004)	***0.872 (0.003)***	***0.867 (0.003)***

LC	training acc.	test acc.	AUC (training)	AUC (test)
$D_{eu}(\mathbf{x}, \mathbf{w})$	0.780 (0.006)	0.757 (0.004)	0.809 (0.003)	0.787 (0.003)
$D_{cs}(\mathbf{x}, \mathbf{w})$	0.741 (0.005)	0.697 (0.009)	***0.825 (0.002)***	***0.796 (0.003)***

Table 3.1: Numerical results for WBC, CMD and LC data sets; mean accuracies in the unbiased LVQ classifier and AUC with respect to training and test sets, respectively. Numbers in parentheses give the standard deviation over the validation runs. The best performance in each case is indicated in bold type.

Note that for the WBC (original) data set, higher accuracies have been reported in the literature, see (Asuncion et al. 1998) for references. Here, we consider only the simplest LVQ setting in order to compare the use of Euclidean and divergence based distance measures. The optimization of the DLVQ performance is partially addressed in the next chapter.

For the WBC data set, we do not observe drastic performance differences between the considered distance measures. Similarly, in the high-dimensional LC data set, the use of the Cauchy-Schwarz divergence appears to yield a better AUC than when the Euclidean measure is used. The Cauchy-Schwarz based DLVQ scheme, however, does outperform standard Euclidean LVQ in the CMD data. We conjecture that the superior performance accorded to the CMD data set in this case is due to the better representation of the data i.e. as histograms that are a more natural fit for divergence-based LVQ.

3.5.2 The family of γ-divergences

The precise form of the γ-divergence is specified by the parameter γ ($\gamma > 0$) in Eq. (3.5). In our experiments, we compared both measures $D_\gamma(\mathbf{x}, \mathbf{w})$ and $D_\gamma(\mathbf{w}, \mathbf{x})$. We obtain the mean test set accuracies and the AUC of the ROC as functions of γ for both measures of the γ-divergence; Fig. 3.2 (WBC data) and Fig. 3.3 (CMD data).

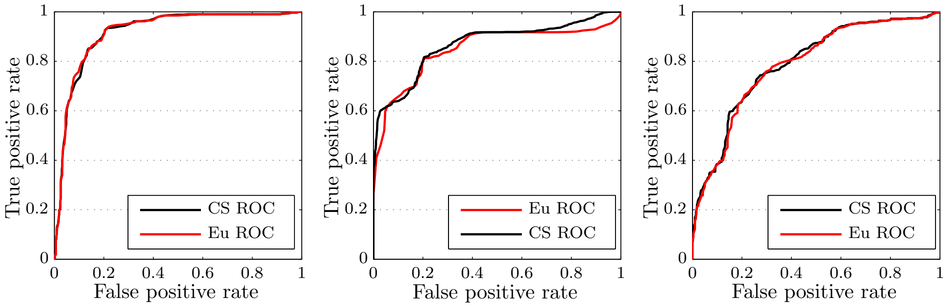


Figure 3.1: ROC curves for the WBC data set (left panel), CMD data set (middle panel) and LC data set (right panel). For the WBC and LC data, results are shown as an average over 100 randomized training set selections, whereas for the CMD data we performed 200 randomized runs. In both cases, the ROC curves were threshold-averaged (Fawcett 2006). Results are displayed for the GLVQ variants based on Euclidean distances (light lines) and Cauchy-Schwarz divergence (dark lines).

In both data sets we do observe a dependence of the AUC performance on the value of γ with a pronounced optimum in a particular choice of the parameter. This is not necessarily paralleled by a maximum of the corresponding test set accuracy as the latter represents only one particular working point of the ROC.

Note that in the range of values of γ displayed in Fig. 3.3 (right panel), corresponding to the use of $D_\gamma(\mathbf{w},\mathbf{x})$, the AUC appears to saturate for large values of the parameter. Additional experiments, however, show that performance decreases weakly when γ is increased further.

For both data sets, the influence of γ appears to be strong and the best achievable AUC is slightly larger in the DLVQ variant using $D_\gamma(\mathbf{x},\mathbf{w})$. Table 3.2 summarizes numerical results in terms of the best observed test set AUC and the corresponding

WBC	γ	AUC (test)
$D_\gamma(\mathbf{x},\mathbf{w})$	***0.6***	***0.922 (0.004)***
$D_\gamma(\mathbf{w},\mathbf{x})$	0.5	0.919 (0.005)

CMD	γ	AUC (test)
$D_\gamma(\mathbf{x},\mathbf{w})$	***0.2***	***0.888 (0.003)***
$D_\gamma(\mathbf{w},\mathbf{x})$	1.2	0.882 (0.004)

Table 3.2: Best performance in terms of the mean test set AUC and corresponding value of γ for the WBC and CMD data sets. Values in parenthesis correspond to the observed standard deviations. The best performance in each case is indicated in bold type.

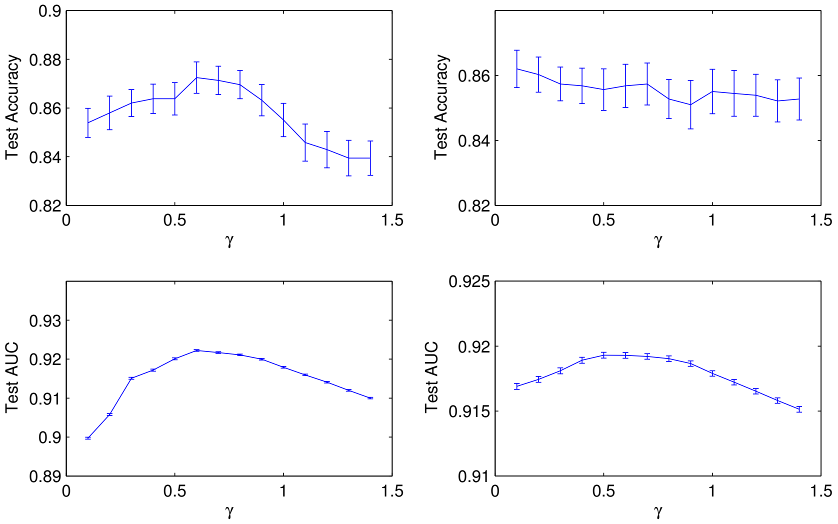


Figure 3.2: Overall test set accuracies for the unbiased LVQ system with $\theta = 0$ (upper panels) and AUC of the ROC (lower panels) as a function of γ for the WBC data set. The left panel displays results for the distance $D_\gamma(\mathbf{x}, \mathbf{w})$, while the right panel corresponds to the use of $D_\gamma(\mathbf{w}, \mathbf{x})$. Error bars mark the observed standard errors of the mean of the scores.

values of γ as found for WBC and CMD data in both variants of the γ-divergence.

3.6 Summary and Conclusion

We have presented DLVQ as a novel framework for distance based classification. The use of divergences as distance measures is, in principle, possible for all data sets that contain non-negative feature values. It appears particularly suitable for the classification of histograms, spectra, or similar data structures for which divergences are the natural representation.

As a specific example of this versatile framework we have considered the family of γ-divergences which contains the so-called Cauchy-Schwarz divergence as a special case and approaches the well-known Kullback-Leibler divergence in the limit $\gamma \to 0$. We would like to point out that a large variety of differentiable measures could be employed analogously; an overview of suitable divergences is given in (Villmann and Haase 2009, Villmann et al. 2010).

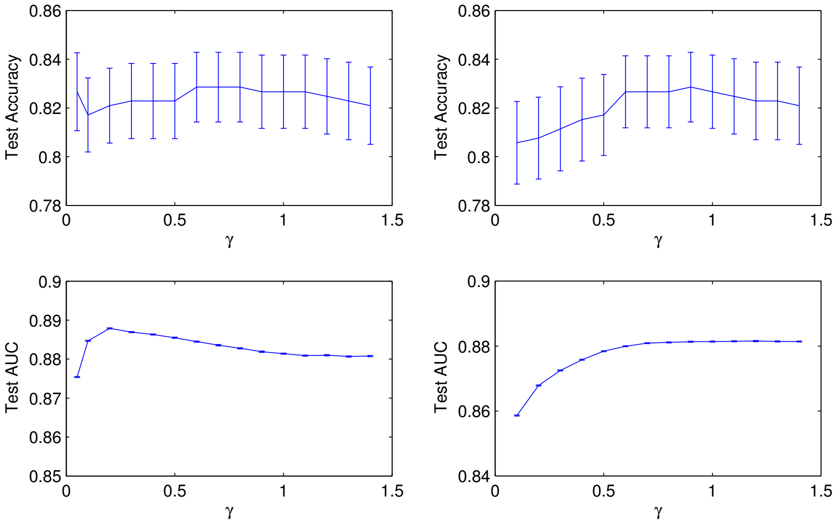


Figure 3.3: Same as Figure 3.2, but here for the CMD data set. The left panel corresponds to the use of $D_\gamma(\mathbf{x}, \mathbf{w})$, while results displayed in the right panel were obtained for $D_\gamma(\mathbf{w}, \mathbf{x})$.

The aim of this work was to demonstrate the potential usefulness of the approach. To this end, we considered three example data sets: WBC data, LC data and CMD data. The Wisconsin Breast Cancer data is available from the UCI Machine Learning Repository (Asuncion et al. 1998) and serves as a popular benchmark problem for two-class classification. The second data set comprises mass spectra data for the diagnosis of Lung Cancer and the third data set comprises histograms which represent leaf images used for the detection of the Cassava Mosaic Disease (this data set is further discussed in more detail in Chapter 10 as well).

In the case of the WBC and LC data, we observed little differences in performance quality when standard Euclidean metrics based LVQ was compared with DLVQ employing the Cauchy-Schwarz divergence. When using the more general γ-divergences, a weak dependency on γ is found which seems to allow for improving the performance slightly by choosing the parameter appropriately.

In contrast to the WBC data set, the CMD data set consists of genuine histogram data and the use of divergences appears more natural. In fact, we find improvement over the standard Euclidean measure already for the symmetric Cauchy-Schwarz divergence. Further improvement can be achieved by choosing an appropriate value of γ in both variants of the non-symmetric distance measure. The dependence on γ and its optimal choice is found to be data set specific.

The application of DLVQ appears most promising for problems that involve data with a natural interpretation as probabilities or positive measures. Potential applications include image classification based on histograms, supervised learning tasks in the medical field, and the analysis of spectral data as in bioinformatics or remote sensing.

Besides the more extensive study of practical applications, future research will also address several theoretical and conceptual issues. The use of divergences is not restricted to the GLVQ formulation we have discussed here, it is possible to introduce DLVQ in a much broader context of heuristic or cost function based LVQ algorithms. Within several families of divergences it appears feasible to employ hyperparameter learning in order to determine, for instance, the optimal γ directly in the training process, see (Schneider et al. 2010) for a similar problem in the context of Robust Soft LVQ (Seo and Obermayer 2003). The use of asymmetric distance measures also raises interesting questions concerning the interpretability of the LVQ prototypes, this is one prospect for future work as well.

Finally, the incorporation of relevance learning (Bojer et al. 2001, Hammer and Villmann 2002, Schneider et al. 2009a) into the DLVQ framework is possible for measures that are invariant under rescaling of the data, such as the γ-divergences investigated here. Relevance learning in DLVQ bears the promise to yield very powerful LVQ training schemes. Again we propose this as a possible topic for future work.

Chapter 4

LVQ with Combined Distance Measures

Abstract

We present an extension to the family of LVQ that provides a more robust classification by combining different distance measures for different representations of data into one LVQ learning scheme. This is applicable for problems where the raw data can be represented in multiple ways for example several different features can be extracted from image data including color information and features about shape and interest points. A similar example is in medical diagnosis problems, where different pieces of data including haematology data, scan images and medical history, also exist for the problem of determining a correct diagnosis. We develop the formalism for the combined distance measures based LVQ and test the algorithm on two datasets of leaf images used to detect the presence of Cassava Mosaic Disease in cassava plants. We show that performance of the classifier improves significantly when multiple facets of data and multiple distance measures are employed.

4.1 Introduction

Previous chapters have presented LVQ and its variants as a family of prototype-based supervised learning algorithms that do classification by computing the similarity (dissimilarity) between new data and learned prototypes from a training set. They essentially require the definition of a distance measure to quantify the dissimilarity between data examples and prototypes defined in the data. *De-facto* distance measures have been based on Minkowski distance metrics for example the Euclidean distance and the Manhattan distance. Some of our earlier work has shown that selecting distance measures that are mapped to the nature of the data provide superior performance. This is the case in Divergence based LVQ (DLVQ) (Chapter 3) where divergences are used as a distance measure for data that is normalized and non-negative (or that can be represented as distributions).

For some classification problems it is possible that the data generating process can output several variants of the represented physical quantity as is the case with generating data from images by feature extraction. The same images can be represented by different feature sets for example HSV, RGB histograms, SIFT features and SURF features. In medical diagnoses, as well, multiple pieces of data are normally required to make the diagnosis including images, lab results or data from other physiological tests. In this chapter, we investigate how to combine different facets of data into one classification scheme by defining a global distance measure. It draws from initial work on the application of LVQ to heterogeneous structured data (Zühlke et al. 2010). The global measure is a combination of the different heterogeneous distance measures corresponding to the varied data facets or representations. We present results from using different combinations of the distance measures: linearly and using a matrix that takes into account correlations between the distance measures.

LVQ with combined distance measures can be viewed as a generalization of these various adaptive measures. We formalize the combination of the different distance measures using two approaches: linearly and using a matrix. We further present the usage of the LVQ with combined distance measures method as an advanced technique of tackling the problem of diagnosing viral disease in crops (looked at in Chapters 3 and 10) by using images of leaves with varied backgrounds (taken *in-situ*).

4.2 Adaptive distance measures in LVQ

In Chapters 2 and 3, we have discussed LVQ and showed the importance of selecting an appropriate distance measure. The most commonly used distance measure is the well known Euclidean distance. Some factors of the Euclidean distance however make it a sub-optimal measure especially for noisy data. One such factor is that Euclidean distance weights all dimensions of data equally. The implications of this are that each dimension will have equal importance in the classification task. For noisy data, this is problematic because the classifier will tend to model the noise in the data resulting in poor generalization. Another factor is that if the features are not scaled uniformly or are correlated then the Euclidean metric again has poor discriminative power.

A solution to this problem is to have adaptive distance measures that are also optimized during the training process. For each unique application, a unique distance measure is learned and used in the classification task. Initial work in this regard was done by (Bojer et al. 2001), from which the so called Relevance Learning Vector Quantization (RLVQ) scheme was spawned.

In RLVQ, the distance measure $d(\mathbf{x}, \mathbf{w})$ is specified as a squared Euclidean distance with an additional parameter λ that represents the relevance profile of the different feature dimensions. The modified distance measure is as follows

$$d^{\lambda}(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{N} \lambda^i (x^i - w^i)^2, \tag{4.1}$$

$$\textit{with} \quad \lambda \in I\!R^N, \quad \lambda^i > 0 \;\; \textit{and} \;\; \sum_i \lambda^i = 1. \tag{4.2}$$

During the training phase, the parameters λ^i are updated to reflect the relevance of different features for the classification. λ is thus called a relevance vector. One of the key advantages of LVQ is the ease with which one can interpret the learned prototypes at the end of the training. This advantage is further boosted when the algorithm outputs a relevance vector that additionally informs the user of which dimensions are critical for the classification task at hand. Several examples in the literature have shown that this advantage cannot be overstated (Mendenhall and Merényi 2008, Biehl et al. 2007, Kietzmann et al. 2008). Further research that presents a theoretical background for the concepts of adaptive distance measures and metric learning can be found in (Hammer et al. 2005a) and extensions of these concepts to other distance-based classifiers e.g. k-NN, can be found in (Shalev-Shwartz et al. 2004) and (Weinberger et al. 2006).

An extension of RLVQ that takes advantage of the differentiable GLVQ cost function (presented in Section 2.2.2) is the Generalized RLVQ (GRLVQ) by (Hammer and Villmann 2002). In GRLVQ a better formulation of the parameter λ that is updated globally during the training process is obtained.

The relevance vectors λ, however, need not be global. In an extension of GRLVQ called Localized GRLVQ (LGRLVQ), different relevance factors can be defined for the different prototypes to take care of the local dimensionality weights in the data space of each prototype.

The most advanced metric for quantifying relevances is the metric based on a matrix that captures the feature-wise correlations within the data. This extension is called Generalized Matrix LVQ (GMLVQ) (Schneider et al. 2009a, Schneider et al. 2009b). In GMLVQ the distance measure takes the following form

$$d^{\lambda}(\mathbf{x}, \mathbf{w}) = (\mathbf{x} - \mathbf{w})^{\top} \Lambda (\mathbf{x} - \mathbf{w}), \tag{4.3}$$

where $\Lambda \in I\!R^{N \times N}$ is the matrix that represents the feature-wise correlations in the data. By enforcing that the matrix Λ is positive semi-definite and symmetric, the distance measure can be viewed as a generalized squared Euclidean distance in an appropriately transformed space. It follows that a simplification of the matrix can be obtained as

$$\Lambda = \Omega^{\top} \Omega \qquad \text{with} \qquad \Omega \in I\!R^{N \times N}. \tag{4.4}$$

Rewriting Eq. (4.3) with the substitution of Λ yields

$$d^{\lambda}(\mathbf{x},\mathbf{w}) \;=\; [(\mathbf{x}-\mathbf{w})^{\top}\Omega^{\top}]\,[\Omega(\mathbf{x}-\mathbf{w})] = [\Omega(\mathbf{x}-\mathbf{w})]^{2}\,. \tag{4.5}$$

The relevance matrices Ω are learned during the training by a gradient descent over the GLVQ cost function with the distances substituted for the GMLVQ distances. Just like in RLVQ, at every training step, the matrix Λ is normalized to prevent the algorithm from degeneration. The normalization is enforced by ensuring

$$\sum_{i}\Lambda_{ii} = \sum_{ij}\Omega_{ij}^{2} = 1. \tag{4.6}$$

The matrix Ω at each step of the training process is updated as follows

$$\begin{aligned}
\Delta\mathbf{\Omega} = &\quad -\psi.\left(\tfrac{2d_K(\mathbf{x})}{(d_J(\mathbf{x})+d_K(\mathbf{x}))^2}\;\tfrac{\partial}{\partial\mathbf{\Omega}}\,d_J(\mathbf{x}) \;-\; \tfrac{2d_J(\mathbf{x})}{(d_J(\mathbf{x})+d_K(\mathbf{x}))^2}\;\tfrac{\partial}{\partial\mathbf{\Omega}}d_K(\mathbf{x})\right),\\
= &\quad -\psi.\left(\mu_J(\mathbf{x})\;\tfrac{\partial}{\partial\mathbf{\Omega}}\,d_J(\mathbf{x}) \;-\; \mu_K(\mathbf{x})\;\tfrac{\partial}{\partial\mathbf{\Omega}}d_K(\mathbf{x})\right), \qquad (4.7)\\
\text{with } \mu_J(\mathbf{x}) = &\quad \tfrac{2d_K(\mathbf{x})}{(d_J(\mathbf{x})+d_K(\mathbf{x}))^2} \quad \text{and} \quad \mu_K(\mathbf{x}) = \tfrac{2d_J(\mathbf{x})}{(d_J(\mathbf{x})+d_K(\mathbf{x}))^2},
\end{aligned}$$

where J and K denote the indices of the closest correct and closest incorrect prototype as in Algorithm 2. In the next section, we extend some of these ideas to datasets where different distance measures can be employed.

4.3 Combining distance measures in LVQ

Consider a problem where the data consists of p varied components, one can define unique distance measures for the different components. For example with image data, one component could represent features extracted as color histograms, in which case the definition of a distance measure could be based on divergences, for another component where shape features have been extracted, one may define Euclidean like metrics. Another possible scenario could be the application of different distance measures to the same data and combining these distance measures into one global distance. Formulating the combination of these different distance measures to describe one global distance for the classification problem is the subject of this section.

The formulation of *combined-distances LVQ* (CDLVQ) is similar to the GMLVQ formulation with the dimensions representing different components of the same (heterogeneous) data. The matrix Λ can then be re-defined as a matrix that defines the correlations of the component-wise distance measures. We formalize the global distance as

$$\mathbf{D}^{\mathbf{\Lambda}} = \begin{bmatrix} d_1(\mathbf{x}^{(\mathbf{1})}, \mathbf{w}^{(\mathbf{1})}) \\ d_2(\mathbf{x}^{(\mathbf{2})}, \mathbf{w}^{(\mathbf{2})}) \\ \vdots \\ d_p(\mathbf{x}^{(\mathbf{p})}, \mathbf{w}^{(\mathbf{p})}) \end{bmatrix}^T \Lambda \begin{bmatrix} d_1(\mathbf{x}^{(\mathbf{1})}, \mathbf{w}^{(\mathbf{1})}) \\ d_2(\mathbf{x}^{(\mathbf{2})}, \mathbf{w}^{(\mathbf{2})}) \\ \vdots \\ d_p(\mathbf{x}^{(\mathbf{p})}, \mathbf{w}^{(\mathbf{p})}) \end{bmatrix}.$$

and more succinctly as

$$\mathbf{D}^{\mathbf{\Lambda}} = \boldsymbol{d}(\mathbf{x}, \mathbf{w})^T \, \Lambda \, \boldsymbol{d}(\mathbf{x}, \mathbf{w}).$$

where Λ is the matrix that defines the correlations between the different distance measures, and where $d(\ldots)$ is the vector of all distances $d_L(\mathbf{x}^{(\mathbf{L})}, \mathbf{w}^{(\mathbf{L})})$, $L = 1, \ldots, P$ (P is the number of components in the data or distance measures) and $\mathbf{x}$ represents a concatenation of all data components and $\mathbf{w}$ a concatenation of all prototype components. A linear combination of the distance measures is a special case of the matrix combination with the off-diagonal elements of the matrix Λ set to *zero*. This is similar to the formulation of GRLVQ with the relevance vector λ being represented by the diagonal of the matrix Λ. As was the case with GMLVQ, Λ can conveniently be re-written as $\Lambda = \Omega^\top \Omega$ which also ensures the matrix remains positive semi-definite.

$$\begin{aligned} \mathbf{D}^{\mathbf{\Omega}} &= \boldsymbol{d}(\mathbf{x}, \mathbf{w}) \, \Omega^\top \Omega \, \boldsymbol{d}(\mathbf{x}, \mathbf{w}), && (4.8) \\ &= \sum_{q,r,s} d_q(\boldsymbol{x_i}, \boldsymbol{w}_j) \Omega^\top_{qs} \Omega_{sr} d_r(\boldsymbol{x_i}, \boldsymbol{w}_j), && (4.9) \end{aligned}$$

where p, q, r, s are considered indices over the different distance measures and elements of $\mathbf{D}^{\mathbf{\Lambda}}$. Training follows a batch gradient descent over a GLVQ cost function and the prototypes $\mathbf{w}$ and matrix $\mathbf{\Omega}$ are updated after every run through the dataset. Each sub-prototype $\Delta \mathbf{w}^{(p)}$ is updated based on the global distance $\mathbf{D}^{\mathbf{\Omega}}$. From the general equation of the GLVQ update given in Eq. (3.2) we obtain

$$\begin{aligned} \Delta \mathbf{w}^{J(P)} &= \frac{-\eta \, D^\Omega_K(\mathbf{x})}{\left(D^\Omega_J(\mathbf{x}) + D^\Omega_K(\mathbf{x})\right)^2} \frac{\partial}{\partial \mathbf{w}^J}^{(P)} D^\Omega_J(\mathbf{x}), && (4.10) \\ \Delta \mathbf{w}^{K(P)} &= \frac{+\eta \, D^\Omega_J(\mathbf{x})}{\left(D^\Omega_J(\mathbf{x}) + D^\Omega_K(\mathbf{x})\right)^2} \frac{\partial}{\partial \mathbf{w}^K}^{(P)} D^\Omega_K(\mathbf{x}). && (4.11) \end{aligned}$$

The derivative for the update of the prototype component (sub-prototype) $\Delta \mathbf{w}^{(p)}$ is thus given as

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}}^{(P)} D^\Omega &= \sum_{q,r,s} \left(\left[\nabla_{w_p} d_q\right] \Omega_{qs} \Omega_{sr} d_r \right) + \sum_{q,s,r} \left(d_q \Omega_{qs} \Omega_{sr} \left[\nabla_{w_p} d_r\right] \right), \\ &= 2 \left[\boldsymbol{d} \Omega^T \Omega\right]_p \nabla_{w_p} d_p. && (4.12) \end{aligned}$$

The full derivation of Eq. (4.12) is given in Appendix 4.A. From Eq. (4.7), the corresponding update for a single matrix element Ω_{rs} is

$$\begin{aligned} \Delta\Omega_{rs} &= \psi\,(\mu_J(\mathbf{x})\frac{\partial D_J^{\Omega}}{\partial\Omega_{rs}} - \mu_K(\mathbf{x})\frac{\partial D_K^{\Omega}}{\partial\Omega_{rs}}), \qquad (4.13) \\ \text{with} \quad \frac{\partial D_i^{\Omega}}{\partial\Omega_{rs}} &= [\Omega\boldsymbol{d}(\boldsymbol{x},\boldsymbol{w}_i)]_s d_r(\boldsymbol{x},\boldsymbol{w}_i) + [\Omega\boldsymbol{d}(\boldsymbol{x},\boldsymbol{w}_i)]_r d_s(\boldsymbol{x},\boldsymbol{w}_i). \qquad (4.14) \end{aligned}$$

The corresponding derivation of the matrix update Eq. (4.13) is given in Appendix 4.B.

At each step in the training the collection of winning sub-prototypes is calculated and updated with respect to the global distance $\mathbf{D}^{\mathbf{\Omega}}$. As such each sub-prototype representing a unique representation of the same object (e.g. an image) is updated based on its relevance to the overall classification as well as its correlation with other representations of the object.

4.4 Experiments

In our experiments we consider the image dataset for diagnosing Cassava Mosaic Disease (CMD) introduced in Chapter 3. We consider two variants of this dataset: (i) a lab-cassava dataset where images of individual leaves are taken in the lab under controlled lighting and uniform background (also expounded on in Chapter 3), and (ii) a field-cassava dataset where images are taken of leaves in the field with typical field background noise including other leaves, ground, trees or shadows. CMD manifests as de-colourization of the plant leaves, as well as deformation of the leaves in mature stages of the disease. For our experiments, we extracted five representative features for each image: three HSV (Hue, Saturation, Value/Intensity) colour histograms to characterize the de-colourization, one set of SIFT features and another set of SURF features to characterize the deformation.

These five different representations of the images form heterogeneous components of the image data for which we can define different distance measures. The different distance measures are combined into one global distance measure. Table 4.1 shows the distance measures we defined for the different components of the image data.

In this section, we present results of applying our combined distance measures based LVQ algorithm on this data. We present results for different combinations of the data components as well as results for different initializations of the distance correlation matrix Ω. We run these experiments for 4 cross-validated folds of the data, in each fold training the system through 2 runs with different initialisations

Data Component	*Distance Measure*	*Parameter Setting*
HSV histograms	γ divergence	γ = 1.5
SIFT data (normalized)	CS divergence	
SURF data	Euclidean	

Table 4.1: Different distance measures for the different image data components.

and 100 training epochs. An adaptive step size scheme is used during the batch gradient descent to control the step size. The step size is adapted based on previous performance over a set number of training epochs. Results are obtained using 1 prototype per class.

Table 4.2 shows the results for Receiver Operating Characteristics AUC scores for different combinations for the two data sets.

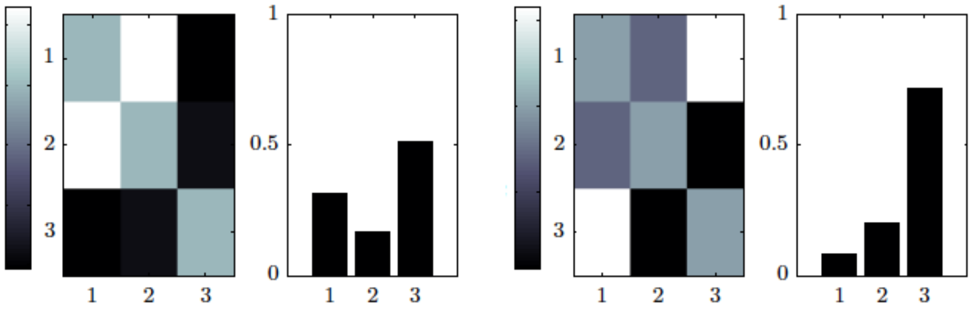
Data Combinations	*Lab-Cassava ds*	*Field-Cassava ds*
H,S,V	0.924	0.976
H,S,V,Sift	0.963	0.977
H,S,V,Surf	0.909	(data unavailable)

Table 4.2: AUC results for different combinations of the 5 data components; Hue histogram (H), Saturation histogram (S), Intensity histogram (V), Sift features and Surf features.

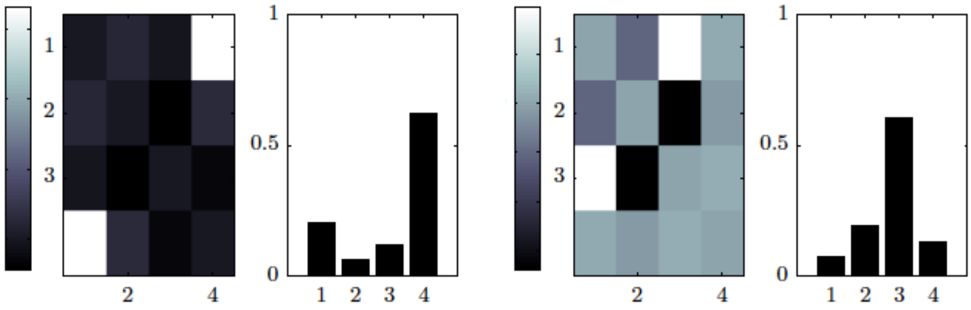
Figure 4.1 shows the corresponding plots of the matrix $\Lambda = \Omega^\top\Omega$ for the different experiments corresponding to Table 4.2 (the first two rows). We do not consider SURF features because the performance degrades with addition of the features as evidenced in Table 4.2. These plots give an idea of how relevant each data component is for the classification.

4.4.1 Analysis

Results in Table 4.2 show an improvement in performance from 92 % to 96 % for the lab-cassava dataset when SIFT features are added to the LVQ system. A slightly smaller improvement is noticed in the field-cassava dataset. Addition of SURF features however seems to degrade performance. As expected adding different representations of the same data in a learning scheme affects the overall performance. In our case we observe both variations of the performance with different combinations of the data components.



(a) H,S,V lab-cassava (b) H,S,V field-cassava



(c) H,S,V,SIFT lab-cassava (d) H,S,V,SIFT field-cassava

Figure 4.1: Image plots for the trained matrix Λ for different combinations of the H,S,V and SIFT data representations of the cassava data. Each combination is represented by a plot of the off-diagonal elements with the diagonal set to 0, and a separate plot of the diagonal vector. Top row shows matrices for the H,S,V combination only while the bottom row shows matrices for the H,S,V and SIFT features combination for the two data set types.

The relevance for classification of the different data components in the learning system is depicted in Figure 4.1. From the plot of the diagonal of the matrix Λ, we notice that for the lab-cassava dataset, the hue (H) and intensity (V) histograms play a pivotal role in the classification. From previous experiments explained in Chapter 3, we noticed that if the experiment is left to run for a longer time, the hue histogram significantly becomes the principle component affecting the classification performance. When SIFT features are added to the HSV histograms for the lab-cassava dataset, we notice a shift in reliance of the classifier from the hue histogram to the SIFT features. This probably explains why there is an improvement in the AUC performance from 92 % to 96 % as observed in Table 4.2.

For the field-cassava dataset we observe a greater reliance of the classification on the intensity (V) histogram. Since these images were taken in the field it makes sense that with a noisy background the hue histogram becomes less reliant as a discriminant factor in the classification. Adding SIFT features has little effect in this case for probably the same reason; with a noisy background edge and interest point features tend to be arbitrary. We hence notice that the intensity histogram still is the major factor determining the classification. Again this explains why there is only a small increase in performance of the classification with addition of the SIFT features.

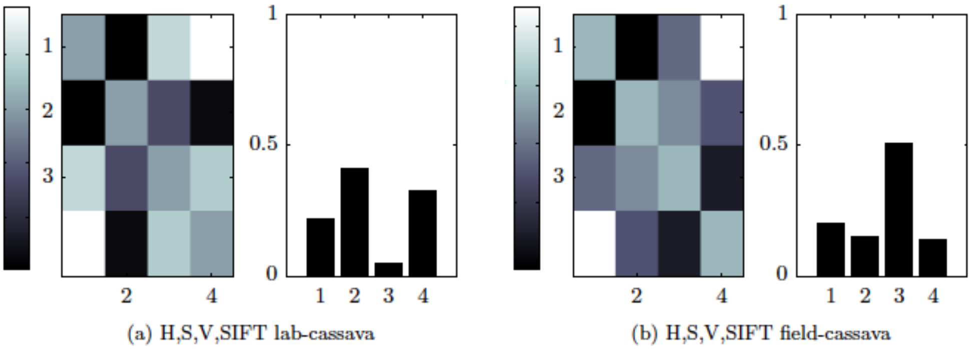
The off diagonal elements in all the plots give a similar intuition to the analysis of the diagonal elements of the matrix. This is to be expected since the matrix was initialized as a matrix with ones on the diagonal and zeros on the off diagonal.

4.4.2 Matrix initialization

The matrix Ω in a sense reflects the relative importance of different data components for the classification. Previous experiments were performed with a unit diagonal matrix at initialization. When Ω is initialized with an arbitrary random matrix, the training and test curves show a turbulent learning process. For the same training parameters as in Section 4.4, slightly poorer performances are obtained for the lab-cassava dataset **(AUC 0.908)** and for the field-cassava dataset **(AUC 0.969)**. Taking the error bars on both results into account, it is safe to say the results are as close to those previously obtained as is statistically valid. It would hence appear that for some random initialization of Ω, the system gravitates towards some optima, and in this case the optima seem to be the same. The final trained matrices from random initialization (Figure 4.2) also show some resemblance to the previously obtained matrices with unit-diagonal initialization.

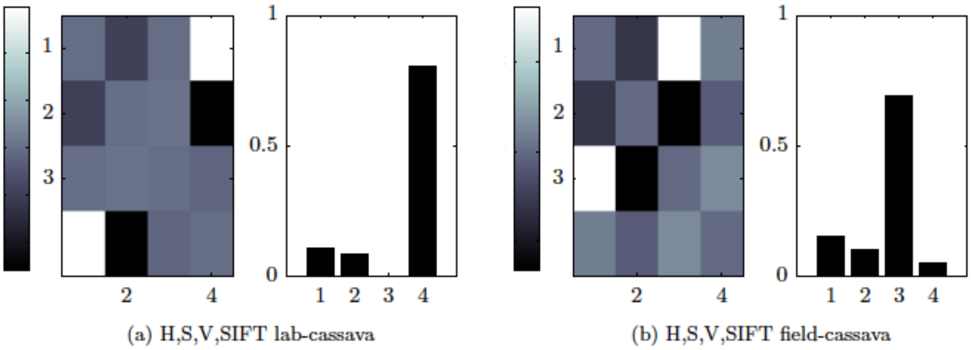
Biasing the initialization

We observed a strong reliance of the classification on the hue histogram and the SIFT data component for the lab-cassava dataset in the first set of experiments. By weighting the initial matrix Ω to favour these component features (e.g. initializing the matrix with a diagonal `[2 1 1 2]` and zeros on the off diagonal) improves the performance on the lab-cassava data **(AUC 0.954)**. A similar improvement however is not observed for the field-cassava data **(AUC 0.951)**. Resultant matrices are shown in Figure 4.3. The off-diagonal matrix for the field-cassava dataset in Figure 4.3 indicates some reliance of the classification on the component-wise correlations between the different component distance measures. This could explain the non-linear relationship between the performance and the biasing of some components in the initialisation matrix.



(a) H,S,V,SIFT lab-cassava (b) H,S,V,SIFT field-cassava

Figure 4.2: Depiction of the correlation matrix Λ and its diagonal for random initialization of the matrix Ω.



(a) H,S,V,SIFT lab-cassava (b) H,S,V,SIFT field-cassava

Figure 4.3: Depiction of the correlation matrix Λ and its diagonal for biased initialization of the matrix Ω

4.5 Prototype membership

Ordinarily the experts who do crop surveillance particularly for the case of CMD categorise the data collected into 5 severity levels. Severity 1 denotes healthy plants, while severity 5 denotes severely diseased plants. For our experiments because of the way the data was collected (Chapters 3,10) we only considered two classes, healthy and diseased. Prototypes defined in the space of the data represent *typical* examples of the different classes of the data. By defining more prototypes per class, we are restricting the prototypes to even smaller representative domains.

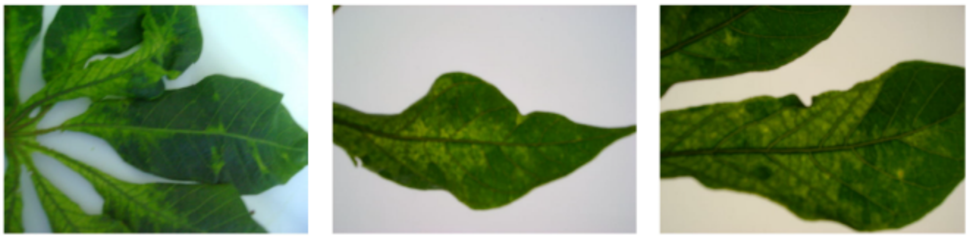
For the lab-cassava data we defined 4 prototypes per class during the training and mapped the trained prototypes from the system to the diseased leaf images by calculating the global distance between the examples and the trained prototypes. This gives an idea of what kind of images the prototypes are representing as well as providing a näive categorisation of the diseased data into possible severity categories. Figure 4.4 depicts some example images in the 4 prototype sub-classes of the diseased class.

Its plausible from Figure 4.4 that we can assign some kind of severity categorisation to the prototypes, but this would need a review from the experts.

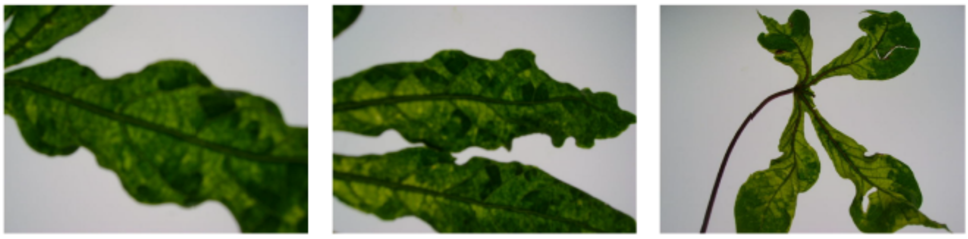
4.6 Conclusion

Experiments were done for a single prototype per class system. Experiments with multiple prototypes per class show a marked improvement in performance of the system. With single prototypes we are already obtaining good performance so no experiments with 2 prototypes were carried out except the 4 prototype experiment for the re-categorization of the diseased data. Training the system for a longer time (larger values of runs and epochs) also results in improved performance albeit not significantly. From previous experiments in Chapter 3 with the histogram data using the γ-divergence, we also noticed optimal performance for $\gamma = 1.5$, thus we did not investigate any other choices of γ here.

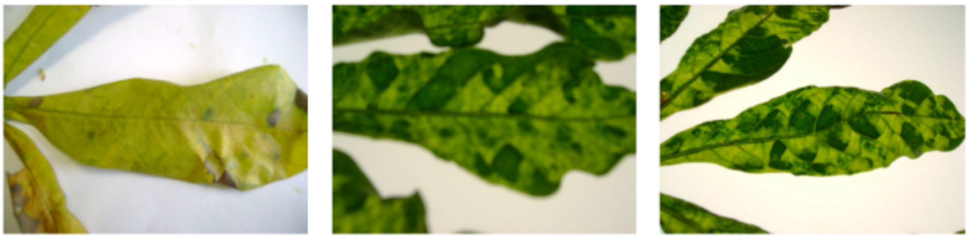
Suffice to say, the major contribution of this chapter is to present an improved way of doing prototype based classification using combined distance measures that appropriately represent problems where data can be represented in multiple forms. While performance over other implementations using a single uniform distance measure for example, was not our concern here, from our experiments, we still observe superior performance over previous implementations with the same dataset (Chapter 3).



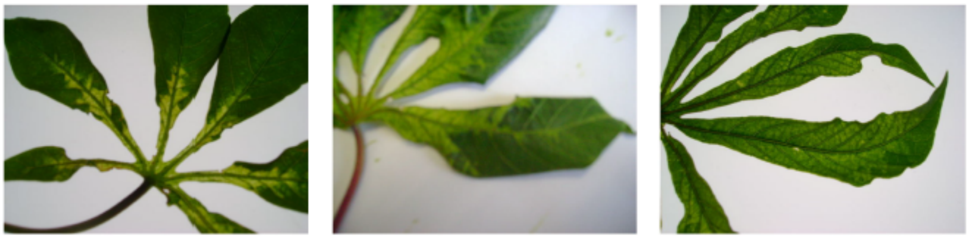
(b) Example prototype 1 images



(e) Example prototype 2 images



(h) Example prototype 3 images



(k) Example prototype 4 images

Figure 4.4: Images represented by prototypes defined in the diseased class for the lab-cassava data

4.A Derivatives of prototype updates for *Combined-Distances LVQ*

The combined prototype $\mathbf{w}$ consists of P non-related sub-prototypes with each sub-prototype being updated separately. The derivative of D^{Ω} with each $\mathbf{w}_p$ (Eq. (4.12)) is given by;

$$\begin{aligned}
\boldsymbol{\nabla}_{w_p} D^{\Omega} &= \sum_{q,r,s} \left(\left[\nabla_{w_p} d_q \right] \Omega_{qs} \Omega_{sr} d_r \right) + \sum_{q,s,r} \left(d_q \Omega_{qs} \Omega_{sr} \left[\nabla_{w_p} d_r \right] \right) \\
&= \sum_{q,r,s} \left(\left[\sum_t \frac{\partial d_{q,t}}{\partial w_{p,t}} \right] \Omega_{qs} \Omega_{sr} d_r \right) + \sum_{q,s,r} \left(d_q \Omega_{qs} \Omega_{sr} \left[\sum_t \frac{\partial d_{r,t}}{\partial w_{p,t}} \right] \right) \\
&= \sum_{r,s} \left(\left[\sum_t \frac{\partial d_{p,t}}{\partial w_{p,t}} \right] \Omega_{ps} \Omega_{sr} d_r \right) + \sum_{q,s} \left(d_q \Omega_{qs} \Omega_{sp} \left[\sum_t \frac{\partial d_{p,t}}{\partial w_{p,t}} \right] \right) \\
&= \sum_{r,s} \left(\left[\nabla_{w_p} d_p \right] \Omega_{ps} \Omega_{sr} d_r \right) + \sum_{q,s} \left(d_q \Omega_{qs} \Omega_{sp} \left[\nabla_{w_p} d_p \right] \right) \\
&= \sum_{r,s} \left(d_r \Omega_{ps} \Omega_{sr} \left[\nabla_{w_p} d_p \right] \right) + \sum_{q,s} \left(d_q \Omega_{qs} \Omega_{sp} \left[\nabla_{w_p} d_p \right] \right) \\
&= \sum_{q,r} \left(d_q \Omega_{pr} \Omega_{rq} \left[\nabla_{w_p} d_p \right] \right) + \sum_{q,r} \left(d_q \Omega_{qr} \Omega_{rp} \left[\nabla_{w_p} d_p \right] \right) \\
&= \sum_{q,r} \left(d_q \Omega^T_{rp} \Omega^T_{qr} \left[\nabla_{w_p} d_p \right] \right) + \sum_{q,r} \left(d_q \Omega_{qr} \Omega_{rp} \left[\nabla_{w_p} d_p \right] \right) \\
&= \sum_{q,r} \left(d_q (\Omega_{qr} \Omega_{rp})^T \left[\nabla_{w_p} d_p \right] \right) + \sum_{q,r} \left(d_q \Omega_{qr} \Omega_{rp} \left[\nabla_{w_p} d_p \right] \right) \\
&= \sum_{q,r} \left(d_q \Omega_{qr} \Omega_{rp} \left[\nabla_{w_p} d_p \right] \right) + \sum_{q,r} \left(d_q \Omega_{qr} \Omega_{rp} \left[\nabla_{w_p} d_p \right] \right) \\
&= 2 \sum_{q,r} \left(d_q \Omega_{qr} \Omega_{rp} \left[\nabla_{w_p} d_p \right] \right) \\
&= 2 \left[\boldsymbol{d} \Omega^T \Omega \right]_p \nabla_{w_p} d_p, \qquad (4.15)
\end{aligned}$$

where $d_p = d_p(\boldsymbol{x}, \boldsymbol{w}_i)$.

4.B Derivatives of the matrix update for *Combined-Distances LVQ*

For the matrix update, the derivative of D^{ω} with respect to Ω for a single element Ω_{rs} (Eq. (4.13)) is given by;

$$
\begin{aligned}
\frac{\partial D^{\Omega}}{\partial \Omega_{rs}} &= \frac{\partial(\sum_{p,q,t} d_p \Omega_{pt} \Omega_{tq} d_q)}{\partial \Omega_{rs}} \\
&= \sum_{p,q,t} d_p \frac{\partial \Omega_{pt}}{\partial \Omega_{rs}} \Omega_{tq} d_q + \sum_{p,q,t} d_p \Omega_{pt} \frac{\partial \Omega_{tq}}{\partial \Omega_{rs}} d_q \\
&= \sum_{q} d_r \frac{\partial \Omega_{rs}}{\partial \Omega_{rs}} \Omega_{sq} d_q + \sum_{p} d_p \Omega_{pr} \frac{\partial \Omega_{rs}}{\partial \Omega_{rs}} d_s \\
&= \sum_{q} d_r \Omega_{sq} d_q + \sum_{p} d_p \Omega_{pr} d_s \\
&= d_r \left[\Omega \boldsymbol{d}\right]_s + \left[\boldsymbol{d}^T \Omega^T\right]_r d_s \\
&= d_r \left[\Omega \boldsymbol{d}\right]_s + d_s \left[\Omega \boldsymbol{d}\right]_r \qquad (4.16)
\end{aligned}
$$

Therefore; (4.17)

$$\frac{\partial D_i^{\Omega}}{\partial \Omega_{rs}} = [\Omega \boldsymbol{d}(\boldsymbol{x}, \boldsymbol{w}_i)]_s d_r(\boldsymbol{x}, \boldsymbol{w}_i) + [\Omega \boldsymbol{d}(\boldsymbol{x}, \boldsymbol{w}_i)]_r d_s(\boldsymbol{x}, \boldsymbol{w}_i) \qquad (4.18)$$

Chapter 5

Causal Structure Learning

Abstract

Causal discovery is the science of finding causes and effects of phenomena. In causal structure learning, we attempt to discover a causal structure; a structure of nodes and directed edges between the nodes, that entails the data being observed. Previously causal discovery was done by use of experiments e.g. Randomized Control Trials (RCTs) where effects from the experiment are said to have been caused by a deliberate action by the experimenters. Recent research in causal machine learning, however, attempts to recover the underlying causal structure from strictly observational datasets. This is a non-trivial task. In this chapter we review causal structure discovery highlighting the assumptions that need to be made and why they need to be made to obtain reasonable confidence in learned structures from purely observational data. This chapter also provides foundational material on which chapters 6, 7 and 9 are built.

5.1 Introduction

Presently the data proliferation in various domains of science necessitates better techniques that can handle the different varieties and sizes of data. A significant section of machine learning attempts to give a better understanding of this data by modeling the data generating process. The field of causal discovery can be viewed as an extension of the machine learning task where we are interested in finding out the causal relationships inherent in the data generating process. Not only do we accrue the advantages that come from solving some basic machine learning problems e.g. better prediction and better generalization, with causal discovery the effects of manipulations (interventions) done on the data can be known.

The study of causality gives us the tools to answer questions about the data generating mechanism for example how certain variables take on the values that they have and how these values would change if other variables were manipulated. Researchers might for example gather data about the social habits and life expectancy (or education and career success, or dietary habits and life expectancy) of a populace with two aims in mind: (1) to find out what social habits affect life expectancy and (2) to predict what the effects of advising people to change certain social habits would be.

Previous study of causality was limited to problems with only a handful of variables and a careful integration of background knowledge and statistical testing was used to infer causality. Experiments had to be done to ascertain causal relationships. In the present situation, the proliferation of data means that these methods stand limited. Problems now span the space of several hundreds of variables, for example in the modeling of gene expressions. Experimentation also tends to be problematic because of the sheer volume of experiments that need to be carried to unearth any meaningful causal relationships. Besides there are ethical and financial considerations attached to experimentation.

The last few decades of research however provides some new ways of doing causal discovery from a careful analysis of observational data (Pearl 2009). This has been aided in no small part by the advances in the computational abilities of current computing equipment. Advances in graphical modeling have also greatly paved the way to novel ways of doing causal discovery. In this chapter we motivate some of the causal discovery ideas from the basics of graphical modeling and discuss the assumptions that make causal discovery possible using these methods and the sorts of interpretations that can be made.

5.2 Bayesian graph representations

Causal modeling generally entails two things: a causal graph that represents the causal relations between the different variables (as we see in the next section) and a statistical model that expresses what the effects of manipulations on the variables will be. Just discovering the causal structure or graph provides already very useful information in understanding the data generating process and can be useful in prediction even when the distribution of the data changes, since causes remain predictive even in this case. In this chapter we focus on discovering this graph, and in chapter 9, we present the application of some of these methods to a specific real-world problem of understanding the different factors affecting food insecurity and how we can predict its onset given some observational data. Deeper insight into the concepts presented in this section and more are addressed in (Pearl 2009).

Causal graphs can be thought of as generalizations of Bayesian networks. Bayesian networks are essentially graphical structures that describe probabilistic relationships between (random) variables. The graphs represent the joint probability distribution of the random variables. Bayesian networks can also be looked at as encapsulating the conditional independence relationships between the variables in a graph. The graph theory concept of a DAG (Directed Acyclic Graph) is particularly useful in formalizing causal relations.

A graph is defined to consist of a set V of vertices (nodes) and a set E of edges that connect pairs of vertices. The vertices in the graphs correspond to variables and the edges denote the relationship between the pair of variables. A directed graph is a pair (V, E) where V is a finite, nonempty set whose elements are called nodes and E is a set of ordered pairs of distinct elements of V. Elements of E are called directed edges, and if $(X, Y) \subseteq E$, then there is an edge from X to Y as shown in Figure 5.1.

A directed graph G is called a Directed Acyclic Graph (DAG) if it contains no cycles. We define a cycle in a directed graph as a path from a node to itself. If the direction of the edge $U \rightarrow Z$ in Figure 5.1 were reversed for example the graph would no longer be acyclic because a cycle would exist: $W \rightarrow Y \rightarrow Z \rightarrow U \rightarrow W$.

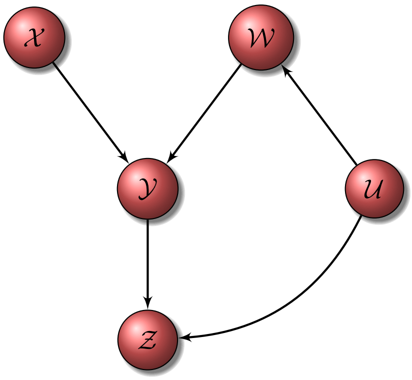


Figure 5.1: Directed Acyclic graph

For a DAG $G = (V, E)$ with nodes $X, Y, Z, U, W \in V$ (Figure 5.1), X is called a parent of Y because of the edge from X to Y, Z is called a descendant of X and X is called an ancestor of Z because of the path from X to Z. U is called a nondescendant of X if U is not a descendant of X and U is not a parent of X as is the case.

It then follows that a *causal DAG* is a DAG in which the edges represent causal relationships between the variables, so that $X \rightarrow Y$ represents the causal relationship X being a direct cause of Y. To move from DAGs to Causal DAGs several assumptions have to be made which we look at in the next section.

Bayesian DAGs however without any assumptions related to causality represent probability distributions. The connection between causation and probabilities can be expressed by what is known as the *Markov condition*.

3. *Definition. The Markov condition*
If we have a joint probability distribution $P(X_i, \dots, X_n)$ of the random variables in some set V and a DAG $G = (V, E)$, we say that (G, P) satisfies the Markov condition if for each variable $X \subseteq V$, X is conditionally independent of the set of all its nondescendents given the set of all its parents. Denoting sets of parents and nondescendents of X as $pa\,(X)$ and $nd\,(X)$, respectively, we have:

$$X \perp\!\!\!\perp nd(X) \mid pa(X).$$

The implication of Definition 3 is that the probabilities entailed by the DAG can be represented by the chain rule

$$P(X_i, \dots, X_n) = \prod_i P(X_i | pa(X_i)). \tag{5.1}$$

P and G are said to be compatible when Eq. (5.1) holds. Ascertaining compatibility between DAGs and probabilities is important in statistical modeling primarily because compatibility is a necessary and sufficient condition for a DAG G to explain a body of empirical data represented by P (Pearl 2009). From the DAG, the set of distributions represented can be read off by expressing the conditional independencies that the graph embodies. This is done using a criterion known as d-*Separation*.

For a disjoint set of variables X, Y and Z represented by a DAG G, to ascertain whether X is independent of Y given Z in any distribution compatible with G, we need to test whether the nodes corresponding to variables Z block all paths from nodes X to nodes in Y.

4. *Definition. d-Separation criterion*
A path p is said to be d-Separated (blocked) by a set of nodes Z iff the following conditions are true:

i. p contains a chain $X_i \rightarrow Z_i \rightarrow Y_i$ or a fork $X_i \leftarrow Z_i \rightarrow Y_i$ such that the middle node Z_i is in Z, or

ii. p contains a collider $X_i \rightarrow Z_i \leftarrow Y_i$ such that the middle node Z_i is not in Z and such that no descendant of Z_i is in Z.

A set Z is said to d-Separate X from Y iff Z blocks every path from a node in X to a node in Y.
Figure 5.2 gives the intuition behind this criterion. Node Z blocks the only direct path connecting X and Y so X and Y are d-Separated by Z in this DAG. From the d-Separation criterion we can thus assume that in all distributions this DAG can represent X is independent of Y conditional on Z.

Of course there are many DAGs that represent exactly the same set of independence relations (also called d-Separation equivalent or Markov equivalent). These also represent the same set of distributions as well. The reverse is however also true: a given set of independence relations can also be represented in more than one DAG. Many algorithms that do causal inference, for example the PC algorithm (Spirtes et al. 1993), generate many Markov equivalent DAGs from a set of conditional independencies derived from data. Figure 5.3 illustrates this; from a single conditional independence assertion, three DAGs can be inferred. These DAGs are called Markov equivalent or d-Separated equivalent graphs.

The tri-variate DAGs named chain, fork and collider are worth mentioning. These 3 variable DAG configurations have the following conditional independences.

Chain : $X \perp\!\!\!\perp Y \mid Z$ $\{X \rightarrow Z \rightarrow Y \text{ or } X \leftarrow Z \leftarrow Y\}.$
Fork : $X \perp\!\!\!\perp Y \mid Z$ $\{X \leftarrow Z \rightarrow Y\}.$
Collider : $X \perp\!\!\!\perp Y$ but $X \not\perp\!\!\!\perp Y \mid Z$ $\{X \rightarrow Z \leftarrow Y\}.$

For causal discovery, the collider configuration is the most important of these because it has a unique set of conditional independences. The collider is a well known pattern in the study of causality also called *explaining away* or *selection bias*; observations of a common consequence of two independent causes almost always implies the causes are dependent, because information about one of the causes tends to make the other more or less likely, given the observed consequence (Pearl 2009). For example if the wetness of the grass can be attributed to two phenomena, the rain and the sprinklers, if the grass was found wet and it was known it had not rained before, then we would infer that the sprinkler had been used, and vice versa. Causal structure learning algorithms rely on finding the collider configurations within graphs and orienting edges from there. The PC Algorithm 3 illustrates this.

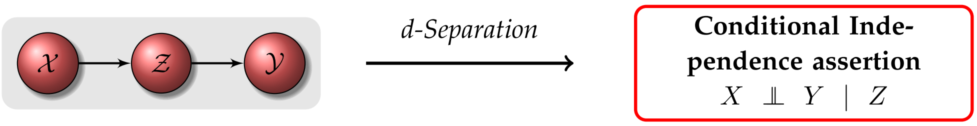

𝒳
𝒵
𝒴
d-Separation
Conditional Independence assertion
$X \perp\!\!\!\perp Y \mid Z$

Figure 5.2: d-Separation Criterion. Node Z effectively d-Separates (blocks) X from Y. The implications on the resulting conditional independence mapping is that $X \perp\!\!\!\perp Y \mid Z$

.

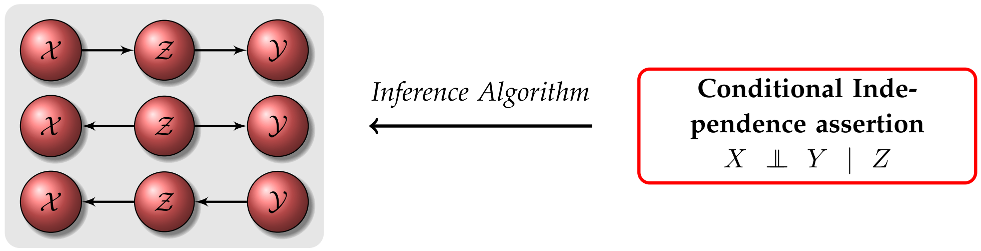

$\mathcal{X}$ → $\mathcal{Z}$ → $\mathcal{Y}$
$\mathcal{X}$ ← $\mathcal{Z}$ → $\mathcal{Y}$
$\mathcal{X}$ ← $\mathcal{Z}$ ← $\mathcal{Y}$
Inference Algorithm
Conditional Independence assertion
$X \perp\!\!\!\perp Y \mid Z$

Figure 5.3: Inference with a causal algorithm. From a set of conditional independences several d-Separated equivalent graphs can be obtained.

5.2.1 Causal Bayesian networks

The Markov condition and d-Separation in general do not necessarily imply a causal relationship. However to interpret any DAG in a causal way the two conditions have to hold as true. The DAGs in this case that are interpreted in a causal manner are called Causal Bayesian networks or causal graphs. An arrow from $X \rightarrow Y$ in this case necessarily means X causes Y. More formally a Causal Bayesian network is defined as follows.

5. *Definition. Causal Bayesian network*
Let V be a set of discrete random variables and P be a joint probability distribution over all possible realizations of V. Let G be a Directed Acyclic Graph (DAG) and let all nodes of G correspond one-to-one to members of V. For every node $X \in V$, X is probabilistically independent of all non-descendants of X, given the parents of X (Markov condition). The triplet $\{V, G, P\}$ is called a (discrete) Bayesian network or a probabilistic network.
A causal Bayesian network is hence a Bayesian network $\{V, G, P\}$ with additional semantics that if $\forall X \in V$ and $\forall Y \in V$, then if there exists an edge from X to Y in G then X directly causes Y.

The Markov condition applying to a causal graph is also renamed the causal Markov condition to reflect the strictness of necessitating the Markov condition and the d-Separation for causal interpretation of a DAG. In order to use samples from probability densities to make causal inferences, the causal Markov condition or assumption of necessity has to be met.

In order to make the leap of inferring a causal structure from non-temporal observational statistical data we have to make some additional assumptions.

Causal faithfulness assumption
The faithfulness assumption holds that if the causal Markov condition is assumed

then there cannot be any other independence relations that are not a consequence of this condition. More simply, a set of variables is deemed faithful if all the independencies that are entailed by the set are not coincidental.

6. *Definition*. *Faithfulness*
Consider a joint probability distribution P of the random variables in some set V and a DAG $G = (V, E)$. We say that (G, P) satisfies the faithfulness condition if all and only the conditional independencies in P are entailed by G. P and G are further said to be faithful to each other.

The faithfulness assumption is generally assumed (perhaps implicitly) in most models. It essentially specifies that the structure of the model entails all the observations as opposed to observations as a result of luck. For causal inference this assumption is however critical because it emphasizes that there is an exact mapping between the conditional independencies observed in a population or set of variables and the structure that generated the data.

Causal sufficiency assumption
The sufficiency assumption is an assumption about the measurement of the variables present in the population as opposed to those not measured.

7. *Definition*. *Sufficiency*
The sufficiency assumption holds that for a causal DAG G, the set of all measured vertices (nodes) V in G should include all the common causes of all pairs of the nodes in V.

The implications of this assumption are that any inferred structure or graph derived from measured conditional independencies in the population can not be affected by latent variables (or unmeasured variables). This weakens whatever deductions we make about the discovered causal structure from the data. So a single conditional independence relation in a collider configuration $X \rightarrow Z \leftarrow Y$ would imply at least an additional nine graphs that are still faithful to this conditional independence if this assumption is not made. For example one could consider a common cause L between X and Z resulting in the graphs $\{L \rightarrow X, L \rightarrow Z, Y \rightarrow Z\}$, $\{L \rightarrow X, L \rightarrow Z, Y \rightarrow Z\}$, $\{L \rightarrow X, L \rightarrow Z, X \rightarrow Z, Y \rightarrow Z\}$, $\{L \rightarrow Y, L \rightarrow Z, X \rightarrow Z\}$, etc. Some algorithms have been developed that relax this assumption and try to detect the presence of possible latent variables in their discovery of a structure, for example IvLiNGAM (Hoyer et al. 2006).

In general, however, by making these assumptions the aim is to improve the fidelity of the causal inferences we deduct; outside these assumptions any deductions made will have a weak causal interpretation.

5.3 Model search

Historically the gold standard for causal discovery and inference has been through the use of randomized controlled experiments (RCTs). Typically for some variable, a subset of instances are manipulated in some way while another set of instances is held constant. Any observations thereafter are attributed to a possible causal relationship. For several investigations however, this tends to be impractical because of the various reasons we have aforementioned including ethics, finance and practicability. For example one cannot force people to smoke to try to determine if smoking actually causes lung cancer.

The role of causal discovery is to discover causal relationships in observational data. The search is generally a two-part search. The search for the causal graph is one part, and the estimation of the parameters of the graph from the sample data and the causal graph is the second part. Typically standard statistical methods are used for the estimation of these parameters, for example maximum likelihood estimation in causal structural equation models (SEMs). The more challenging part we deal with here is the search for the causal graph amongst all the possibilities given the variables of interest.

Typically Occam's Razor is applied in the model search for causal graphs; simpler models are preferred to complex models. The ability to make any reliable inference generally decreases with complexity of the model. The causal faithfulness assumption in part serves as a simplifying assumption for the model selection. Generally two types of approaches are taken: constraint-based search and a Bayesian approach of search and score.

5.3.1 Constraint-based search

Constraint-based methods constrain the search given the causal assumptions to find the Markov equivalence class. The idea is to estimate all the conditional independences from the population and from these infer the causative relationships. The conditional independence tests necessarily employ hypothesis testing where a decision of independence or dependence is made for a pair of variables in the population based on some p-value. By altering the type of independence test, data of different types can be accommodated. For example the Chi-Square test is commonly used for discrete data and Fishers Z-test is used when multivariate Gaussian distributions are suspected. For linear Gaussian data, partial correlation can also be used, as we show in detail in Section 7.1.3 as well.

The PC algorithm (Spirtes et al. 1993) is the quintessential constraint-based causal discovery algorithm. Essentially conditional independence tests for every pair of variables are carried out conditioning on all other sets of variables. The idea is that if two variables are found to be independent with every possible conditioning set, then we know there shouldn't be an edge between them. The skeleton derived from the conditional independence tests is oriented by a search for colliders in the skeleton. The algorithm for the PC algorithm is given in Algorithm 3.

Algorithm 3: PC Algorithm

```
Input  : Dataset $\mathcal{D}$ from observed variables $X_1, \ldots, X_n$
Output: Partially oriented graph (PDAG) $G$

/*Initialization */ ;
$G \leftarrow$ graph with $n$ nodes and no edges ;

/*edge addition*/ ;
for every pair of variables $X_i, X_j$ where $i, j \in \{1, \ldots, n\}$ do
    for every conditioning set of variables
    $S \in \{X_1, \ldots, X_n\} \setminus \{X_i, X_j\}$ do
        Test for independence $X_i \perp\!\!\!\perp X_j | S$ in the data $\mathcal{D}$ ;
    end
    if no independence was found then
        Add undirected edge $(i, j)$ to $G$ ;
    end
end

/*Edge Orientation*/ ;
for every $i, j, k$ for which there are edges $(i, k)$ and $(j, k)$ in $G$
but not in $(i, j)$ do
    Orient the edges $i \to k$ and $j \to k$ in $G$ ;
end

/*Constraint Propagation*/ ;
for every undirected edge $(i, j)$ in $G$ do
    Orient edge $i \to j$ or $j \to i$ if both ;
    (i) no new colliders are created ;
    (ii) no directed cycles are introduced in $G$ ;
end
```

If we assume reliable testing of conditional independencies and that the causal Markov and causal faithfulness assumptions hold, then the PC algorithm is known to output a Markov equivalence class that contains the true causal model. Several other algorithms based on PC have since been proposed that improve its performance including IC* (Pearl 2000), HITON_PC (Aliferis et al. 2003), BN-PC-B (Cheng et al. 2002) and TC (Pellet and Elisseeff 2007).

The advantages of constraint-based methods are that (i) they can easily be extended under relaxed sufficiency assumptions for example IvLiNGAM (Hoyer et al. 2006) is a constraint-based method that can handle DAGs with latent variables, (ii) they can be used with all manner of conditional independence tests that represent the data. Recently non parametric tests have been shown to work with the PC algorithm for example (Sun et al. 2007).

The disadvantages center round the conditional independence testing. Generally, (i) for small sample sizes the conditional independence tests are not reliable especially when conditioning on many variables, and this translates in low fidelity graphs, (ii) the search outputs a Markov equivalence class or more commonly a partial-DAG with no indication of comparative advantage over other classes of models, (iii) mistakes made early in the search are magnified as the search progresses.

However, there are several improved constraint-based algorithms that have tried to solve these problems, for example LiNGAM (Shimizu et al. 2006) finds a fully oriented graph by making assumptions about the non-Gaussianity of parent node distributions; some other algorithms use information-theoretic independence measures (Mooij et al. 2010) while others incorporate some prior information for example using Maximal Ancestral Graphs (MAGs) (Borboudakis et al. 2011).

5.3.2 Bayesian search and score

Bayesian methods generally are based on continually updating a belief in light of new evidence or further information. The Bayesian approach to causal structural discovery follows the same; from *a priori* knowledge about the relationships between variables, a prior probability is placed over each causal DAG G and a posterior probability is calculated for each of the DAGs.

Algorithms which implement this process are typically called *search and score* methods because the space of all posterior DAGs, obtained from the prior and the observed data, is searched and each DAG is scored using some scoring criterion and the DAG with the highest score is selected. This strategy combines three components. First a score to be used is selected. Because the search space of graphs is generally large, the score also serves to penalize complex models over simpler models. Examples of scores include the uniform Bayesian Dirichlet prior (BDeu) (Heckerman 1998), the Bayesian Information Criterion (BIC), Akaike's Information Criterion (AIC) and Minimum Description Length (MDL). The second component is some iterative way of switching between different models e.g. reversing or removing edges in the graph, and the third component is a search strategy for traversing the search space.

Some disadvantages of these methods is that they tend to be slow, are usually dependent on the prior which may be difficult to specify in the absence of background knowledge, and the search space exponentially increases with the number of nodes. A naïve scoring of all possible configurations of a network with just 10 variables yields 10^{18} configurations and quickly becomes intractable with bigger networks. To remedy some of these disadvantages, some algorithms combine constraint-based methods and search and score, for example the Maximum Minimum Hill Climbing (MMHC) algorithm (Tsamardinos et al. 2006) that finds a skeleton initially by using constraint-based methods then orients the edges using a search methodology. We discuss examples of these differently motivated algorithms in Chapter 7.

Work based on:

E. Mwebaze, M. Biehl, J.A. Quinn – *"Causal relevance learning for robust classification under interventions"*, In Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), i6doc.com, 2011.

Chapter 6

Causality and LVQ

Abstract

In some classification problems the distribution of the test data is different from that of the training data because of external manipulations to the variables we observe. We propose a classification scheme which is robust to outside interventions by identifying causes in the training data, given that causes of a target variable remain predictive even when the data is manipulated. We do this by extending Relevance Learning Vector Quantization (RLVQ), a classification scheme that learns a relevance profile for the classification task presented. Our proposed algorithm, Causal-RLVQ, learns a relevance profile that weights causally relevant features more strongly. The algorithm can determine a trade-off between robustness to intervention and accuracy on non-manipulated data, yielding RLVQ as a special case.

6.1 Introduction

The task of performing classification on data for which some of the variables have been externally manipulated is a specific case of the dataset shift problem (Candela et al. 2009). Dataset shift occurs when there is a shift in the joint distribution of the data between training and test stages. A classifier that has been trained on a training set with a particular distribution may not generalize well if future data is of a different distribution. Dataset shift is present in a lot of practical applications, for reasons ranging from the bias introduced by experimental design to the irreproducibility of the testing conditions at training time. For example, we may be trying to classify whether a region is at risk of a disease outbreak based on some environmental and demographic factors, when some of those factors have been directly influenced by other parties in ways not seen in the training data.

In Chapter 5, we looked at causal structure learning of a local neighbourhood given a target variable. The local neighbourhood or the Markov blanket (from the definition of the Markov condition) contains variables that are predictive of the target variable. Inferring the causal local neighbourhood of a variable is akin to feature selection because the variables in the local neighbourhood tend to be the most predictive of the target. In the Chapter 9 we use the variables in the discovered structure for prediction and we recognise some improvement in performance. Other examples of coupling feature selection and causality can be found in the literature, see for example (Guyon et al. 2007). In a situation where dataset shift has occurred however, variable predictive power and causality generally get coupled. For instance, both smoking and coughing may be predictive of lung cancer (the target) in the absence of external intervention; however, prohibiting smoking (a possible cause) may prevent lung cancer, but administering a cough medicine to stop coughing (a possible consequence) would not (Guyon et al. 2008).

Finding the causes of a target variable is thus primarily useful in order to predict the effects of interventions on that variable, and in the last decade a number of methods have been developed to do this on purely observational data (Chapter 5). It follows therefore that there exists a strong link between prediction and causal discovery, and the algorithm we describe in this chapter carries out both tasks simultaneously in a prototype-based learning scheme.

Our method extends the LVQ derivative, relevance learning vector quantization (RLVQ) (Bojer et al. 2001). RLVQ generalises the distance measure of input data such that the features relevant to the target variable are weighted more strongly. Our extension, Causal-RLVQ, introduces a new parameter α which determines how far to bias the relevance weights towards causative features. When $\alpha = 0$, the method is equivalent to standard RLVQ. When $\alpha > 0$, causative features are favoured, giving us robust classification at test time under such cases where the new data is suspected to have been intervened upon. We do this by trying to identify so called V-structures or colliders (section 6.2) in the data.

6.2 Identifying causes in observational data

Techniques for recovering causal structure from observational data have been reviewed in Chapter 5. A fundamental concept in these techniques is conditional independence. When we have sets of three or more variables, conditional independence properties can help to rule in or out particular causal configurations of the variables. X is conditionally independent of Y given Z, written $X \perp\!\!\!\perp Y \mid Z$, if $P(X|Y,Z) = P(X|Z)$.

Figure 6.1 shows the three common configurations of three variables X, Y and Z named consecutively from left to right: *collider* ($X \perp\!\!\!\perp Y$ but $X \not\perp\!\!\!\perp Y \mid Z$), *chain*

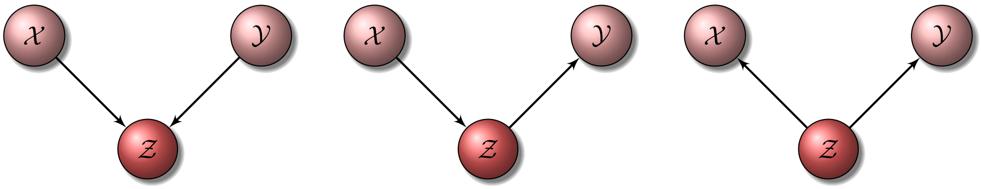


Figure 6.1: Conditional independence configurations of 3 variables: Collider, Chain and Fork

$(X \perp\!\!\!\perp Y \mid Z)$, and *fork* $(X \perp\!\!\!\perp Y \mid Z)$. The chain and fork are in the same conditional independence class, but the collider is uniquely specified by its conditional independence properties. The collider, it turns out, is a very important structure in causal discovery. In hypothesis tests, such as the prototypical Inductive Causation algorithm (Pearl 2000, §2.5) and PC algorithm, colliders are identified by looking for any variables X and Y which are unconnected (dependent upon every conditioning set) with each other, but both of which are connected with a third variable Z. When such a configuration is found, the edges are orientated as in Figure 6.1 (left). This is the approach that we adopt here, using the RLVQ framework itself rather than hypothesis tests.

6.3 RLVQ

The basic LVQ scheme is defined by a set of prototypes defined in the space of the data, and training occurs by comparing the prototypes with the data examples using some dissimilarity measure. Chapter 2 presents a complete introduction to LVQ. We describe a portion of it here to facilitate our discussion of Causal-LVQ.

Formally, for a dataset $D = \{\mathbf{x}^\mu, y^\mu\}_{\mu=1}^{P}$ with $\mathbf{x}^\mu \in \mathbb{R}^N$ and labels $y^\mu \in \{1, 2, \ldots, C\}$, the prototypes are defined by $W = \left\{\mathbf{w}^j, c(\mathbf{w}^j)\right\}_{j=1}^{M}$ with $c(\mathbf{w}^j) \in \{1, 2, \ldots, C\}$. For a particular dissimilarity/distance measure $d(\mathbf{x}, \mathbf{w})$, the LVQ classifier employs a Winner-Takes-All scheme where an arbitrary input is assigned to the class $c(\mathbf{w}^L)$ of the closest prototype with $d(\mathbf{x}, \mathbf{w}^L) \leq d(\mathbf{x}, \mathbf{w}^j)$ for all j.

From Chapter 2, the update for the prototypes $\mathbf{w}$ for a single example $(\mathbf{x}, y)$ can be given as:

$$\mathbf{w}^L = \mathbf{w}^L + \eta d(\mathbf{x}, \mathbf{w}^L), \qquad \text{if } c(\mathbf{w}^L) = y, \tag{6.1}$$

$$\mathbf{w}^L = \mathbf{w}^L - \eta d(\mathbf{x}, \mathbf{w}^L), \qquad \text{if } c(\mathbf{w}^L) \neq y. \tag{6.2}$$

Many modifications of Kohonen's original formulation (Kohonen et al. 2001) have been suggested with the aim of achieving better convergence and generalization behaviour. A specific set of modifications have been towards accounting for heterogeneous datasets where features can have different meanings and magnitudes. These are the class of relevance learning schemes which employ adaptive scaling factors for each dimension in the feature space. For our purposes it will suffice to follow the standard formulation of RLVQ proposed by (Bojer et al. 2001).

RLVQ modifies the distances $d(\mathbf{x}, \mathbf{w})$ by attaching a scaling or relevance factor to each dimension in feature space. The scaling factors are called global relevances if the same factor is applied to all prototypes in the LVQ system. When this is done, the resulting classification boundaries are piecewise linear. Extensions of the standard RLVQ to per-prototype scaling factors have been carried out in the literature (Hammer et al. 2005b, Schneider et al. 2009b, Schneider et al. 2009a). The *local* relevances in this case result in piecewise quadratic boundaries.

The resultant trained RLVQ system outputs relevances that represent the relevance of the different features for the classification problem at hand. If for instance, the factor attached to dimension j in feature space becomes zero, the corresponding feature might as well be omitted from the data set. Thus, relevance learning can serve as a tool for the detection of noisy features which are of little use or can even deteriorate the classification performance if included. Several practical examples in the literature have used this technique successfully (Mendenhall and Merényi 2008, Biehl et al. 2007). RLVQ is formalized as follows.

Consider a generalized Euclidean distance of the form

$$d(\mathbf{x}, \mathbf{w}^J) = \sum_{j=1}^{N} \lambda_j (x_j - w_j^J)^2, \tag{6.3}$$

as the dissimilarity measure where λ_j are the adaptive relevance factors. The special case $\lambda_j = 1/N$ for all $j = 1, \ldots N$ is analogous to the original LVQ1 formulation. Each update of the winning prototype $\mathbf{w}^J$ is accompanied by a corresponding update in the relevance factor $\lambda_j(t)$ as follows

$$\lambda_j(t) = \lambda_j(t-1) - \eta_\lambda \phi \cdot (x_j - w_j^J)^2, \tag{6.4}$$

where $\lambda_j(t)$ is restricted to non-negative values and obeys the normalization $\sum_{j=1}^{N} \lambda_j = 1$, and $\phi \in [1, -1]$. Parameter ϕ is 1 for the nearest correct prototype and -1 for the nearest incorrect prototype.

The λ update hence decreases the relevance factor λ_j if the winning prototype $\mathbf{w}^J$ does represent the correct class but the contribution $(\mathbf{x}_j - \mathbf{w}_j^J)^2$ to $d(\mathbf{x}, \mathbf{w}^J)$ is relatively large. Conversely the weight of a feature with relatively small $(\mathbf{x}_j - \mathbf{w}_j^J)^2$ is increased in such a case. The learning rates η_w and η_λ control the magnitude of the prototype and relevance factor updates at each step.

6.4 Causal-RLVQ

CRLVQ is our extension of the RLVQ scheme where updates favor features that are causally related to the target feature. Our assessment of causal relevance is based on identifying V-structures with respect to the target. RLVQ gives a profile that represents how strongly each single dimension of the data is predictive of the target. In CRLVQ instead of looking at a single dimension, we look for evidence that there are two dimensions x_i and x_k that are predictive of the target. To ascertain that x_i and x_k are in a V-structure with the target we also check that they are independent of each other.

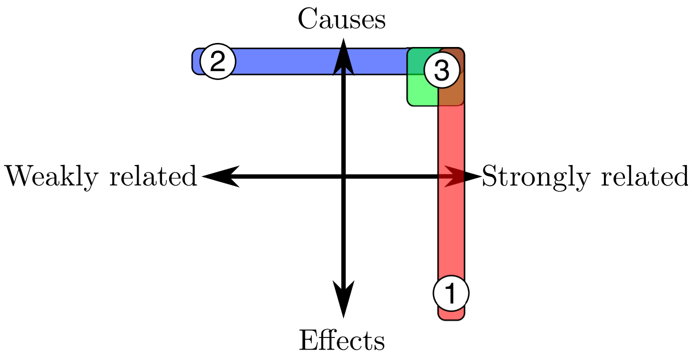


Figure 6.2: RLVQ and CRLVQ formulations. An illustration of placement of variables across the predictive-causal space.

Figure 6.2 illustrates this salient distinction between RLVQ and CRLVQ. We conceive of features $(x_1, \ldots, x_P)$ as being characterisable along two dimensions: their predictiveness of the target variable, and their causal effect on the target variable. Standard relevance learning aims to give higher weight to a set of variables (1) which are highly predictive; causal structure learning aims to find causes (2) or effects of a target; our work here identifies variables which are predictive causes (3). Figure 6.3 shows how differently RLVQ and CRLVQ consider individual features. RLVQ considers each feature separately; CRLVQ takes features pairwise in order to identify V-structures with the target.

CRLVQ extends the RLVQ update by adding two extra terms to Eq. (6.4). The three criteria in total are in a sense a distance-based formulation of the V-structure condition. For every example presented to the CRLVQ classification scheme, each component of λ in CRLVQ is updated for every dimension x_j as follows:

$$\lambda_j(t) = \lambda_j(t-1) - \eta_\lambda \phi \cdot (x_j - w_j^J)^2 - \alpha \eta_\lambda \cdot \left(\min_{k \neq j} \left(\phi \cdot (x_k - w_k^J)^2 - (x_j - x_k)^2 \right) \right) \quad (6.5)$$

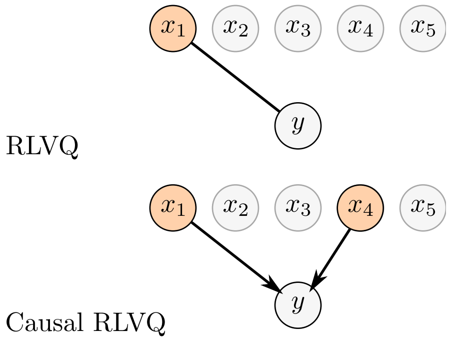


Figure 6.3: The difference between RLVQ and CRLVQ in the way they treat individual features. RLVQ considers singular features at a time while CRLVQ considers pairs of features in order to work out existent V-structures or colliders.

The parameter α is a parameter that weights the two new criteria. Standard RLVQ is hence a special case of CRLVQ when $\alpha = 0$. We evaluate the independence of the different data dimensions by looking at their absolute difference. In Z-score transformed data, this will be a small quantity for pairs of dimensions that are positively correlated. The update hence rewards any feature x_j if it has a strong correlation (small difference) with the target/label vector as represented by a correct prototype $(x_j - w_j^J)^2$, and also if there is strong evidence of another feature with a strong correlation to the target as well $(x_k - w_k^J)^2$, and if feature x_j and x_k are weakly correlated (large difference apart).

Note however that there might be other relationships between the dimensions that make them dependent on each other, but that this update would not capture – for example negative correlation.

6.5 Experiments and results

Testing our method involved two phases: (i) testing that the modified RLVQ, CRLVQ not only identifies relevant features given a dataset, but identifies causally relevant features, and (ii) verifying that the CRLVQ algorithm will offer better or comparable performance when the distribution of the test set is different from that of the training set. In order to achieve these two goals we simulated two datasets whose structure is known and for the second phase, whose variables we can manipulate.

The first dataset was formulated as a 5-feature linear Gaussian network with 2 causes, 2 effects and 1 irrelevant feature, shown in Figure 6.4. The CRLVQ algorithm was run on this dataset several times with varying values of α. Figure 6.5 shows the

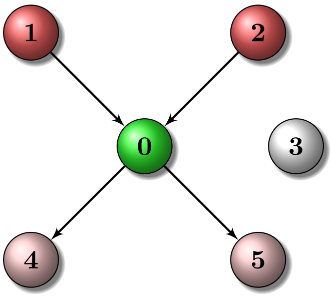


Figure 6.4: Simulated network with 2 causes, 2 effects and 1 irrelevant feature.

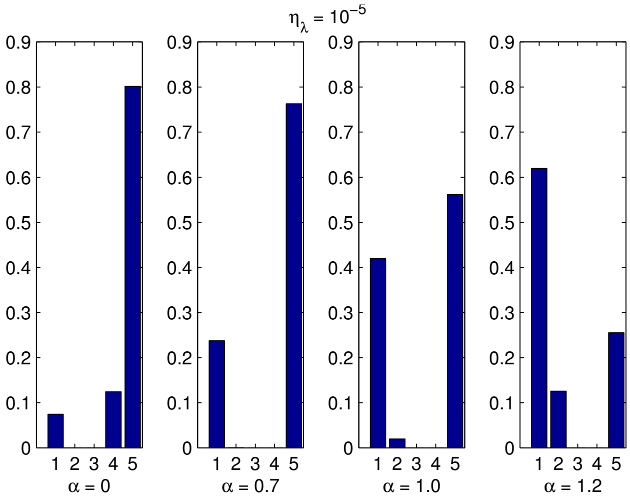


Figure 6.5: CRLVQ relevance weights for simulated network data. Relevance profile under RLVQ and under CRLVQ for different α parameter settings.

results. With $\alpha = 0$, the system represents the standard RLVQ and as expected it selects out relevant features. From the profile of $\alpha = 0$, it appears the algorithm favors the effects ***4*** and ***5*** over the causes. Suspiciously cause **2** appears not to be relevant. It is however plausible that for the classification, the effects are considered most relevant and cause **2** is not all that relevant.

As we increase α the algorithm starts to weigh the causes more importantly. Even cause **2** that was deemed irrelevant by the standard RLVQ obtains a significant weight. For large α ($\alpha = 1.2$) the previously most relevant and highest weighted feature effect ***5*** is reduced in relevance but is still not totally irrelevant. The balancing effect of the RLVQ part of the algorithm is manifested here.

For the second dataset, we used data simulated from a Bayes net and used for several causal competitions as a trial set for tuning causal algorithms. It is commonly called the Lucas dataset (Guyon 2009, Workbench 2010) and tries to predict lung cancer based on different features. This same dataset is used in Chapter 7 as well. The advantage with this dataset is that it represents an intuitive problem of identifying what features are related to lung cancer. Three versions of the dataset were used:

Lucas0 - the natural network (unmanipulated) drawn from a Bayes network represented by the graph in Figure 6.6. The training set for the CRLVQ algorithm was also derived from this dataset.

Lucas1 - this dataset represents the original graph with a number of features (variables) manipulated. A graph of this dataset is shown in Figure 6.6. The manipulated features are highlighted with a light border and are shaded (except the target variable ***0***). The effect of manipulating a feature is to cut off (block) all its parents and ancestors. For example the edge ***1*** → ***6*** represents *Smoking* → *Yellow Fingers*. By intervening on feature ***6*** (*Yellow Fingers*), by painting the fingers yellow for example, then it is no longer possible to say with certainty that *Smoking* causes the fingers to get yellow. From the graph, variables ***6, 1, 11*** and ***8*** have been manipulated making them independent from their parents.

Lucas2 - Lucas2 is a graph of Lucas1 with more variables manipulated. It represents the manipulation of all variables except the target variable. As can be seen, only causes of the target variable are left with any effect on the target variable.

To ascertain the prediction performance for these three datasets, different test sets were drawn from the three datasets. The training set used was from the Lucas0 (unmanipulated) dataset. Table 6.1 shows the test error scores for the three datasets Lucas0, Lucas1 and Lucas2 for CRLVQ with varying α parameters. The first row with $\alpha = 0$ represents RLVQ. Test results are obtained after 50 epochs through the data with $\eta_\lambda = 10^{-5}$.

For Lucas0, the unmanipulated dataset we notice a better test error for RLVQ ($\alpha = 0$) than for any other value of α (CRLVQ). This is plausible because for good classification performance *effects* (of the target variable and possibly other features) are just as relevant as causes. For the manipulated datasets, we notice that test error increases for RLVQ as expected because there are fewer relevant features. We however notice an increase in performance as we tune up α because then only possible causes are identified which remain relevant even under manipulations. Figure 6.7 illustrates the relevance profile for the different values of α. In practice one

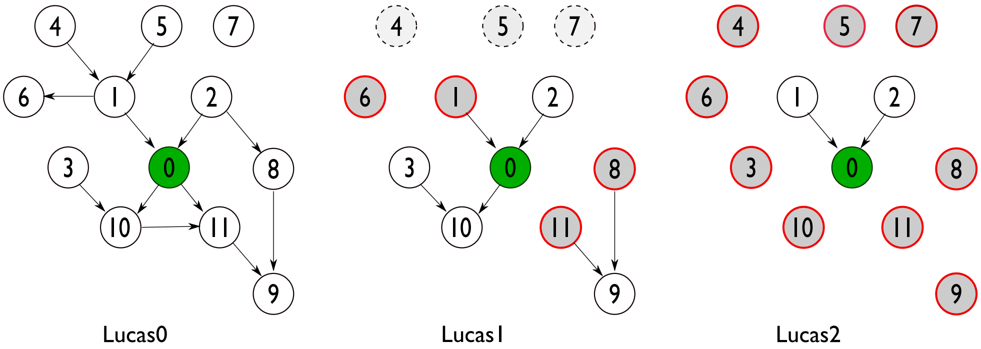


Figure 6.6: Lucas graphs showing the network structures for the unmanipulated set, Lucas0 and the manipulated sets, Lucas1 and Lucas2. The different features in the graphs are labelled as follows: 0 - Lung cancer, 1 - Smoking, 2 - Genetics, 3 - Allergy, 4 - Anxiety, 5 - Peer Pressure, 6 - Yellow Fingers, 7 - Born an Even Day, 8 - Attention Disorder, 9 - Car Accident, 10 - Coughing, 11 - Fatigue.

	Dataset		
α	Lucas0	Lucas1	Lucas2
0	0.1970	0.2370	0.2620
1.0	0.2040	0.2040	0.2030
2.0	0.2040	0.2040	0.2030

Table 6.1: Test error results for RLVQ ($\alpha = 0$) and CRLVQ ($\alpha > 0$) for the different datasets Lucas0, Lucas1 and Lucas2.

would determine the appropriate value of α by doing cross-validation and looking at where the generalisation error deteriorates.

6.6 Conclusion

This chapter serves to combine techniques in both the fields of LVQ and causal discovery into an algorithm that can be parametrized to take full advantage of the strengths of both fields to improve classification. The parameter α allows us to adjust the bias towards causative features, where $\alpha = 0$ yields standard RLVQ. The goal is to assure robust classification in the scenario that some or all variables in the test data could have been subject to external manipulation.

We have validated our algorithm using simulated data because we can control the data generating process. While our assessment of independence amongst features in the data seems to work for these datasets, we realise that this would have to

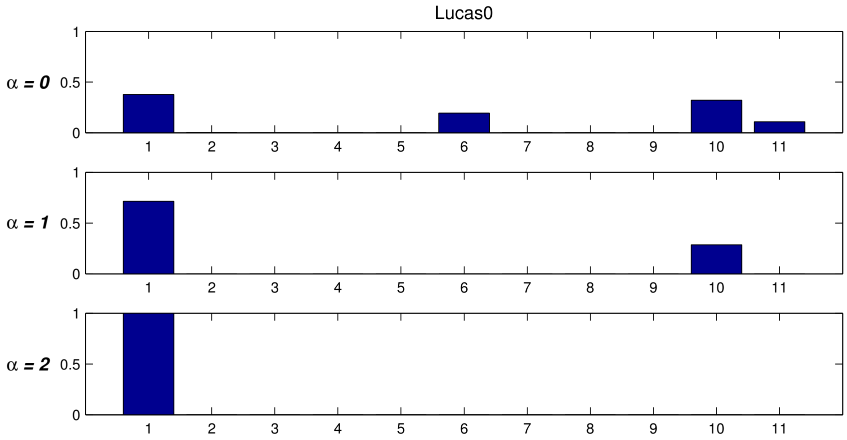


Figure 6.7: RLVQ as α changes.

be tested on other datasets and refined to obtain more certainty about the method. This will constitute future work. Results however are positive and it can be hoped that future attempts with improved versions of RLVQ and better formulation of the assessment of independence will hold more success. An interesting direction to address this could be to extend our current work to matrix relevance LVQ where a full matrix is adapted during the training to reflect the relevance of correlated features of the data (Schneider et al. 2009b).

Work based on:

E. Mwebaze, J.A. Quinn – *"Fast committee-based structure learning"* In Journal of Machine Learning Research W&CP., volume 6, pages 203 - 214, 2009.

Chapter 7

Committee-Based Structure Learning

Abstract

Current methods for causal structure learning tend to be computationally intensive or intractable for large datasets. Some recent approaches have speeded up the process by first making hard decisions about the set of parents and children for each variable, in order to break large-scale problems into sets of tractable local neighbourhoods. We use this principle in order to apply a structure learning committee for orientating edges between variables. We find that a combination of weak structure learners can be effective in recovering causal dependencies. Though such a formulation would be intractable for large problems at the global level, we show that it can run quickly when processing local neighbourhoods in turn. Experimental results show that this localized, committee-based approach has advantages over standard causal discovery algorithms both in terms of speed and accuracy.

7.1 Introduction

Current methods for causal structure learning tend to be computationally intensive or intractable for large datasets. Most approaches towards causal structure learning can be categorized into two classes: constraint-based approaches that use independence tests and score-based techniques that search for Bayesian networks (Chapter 5). The former are slow because independence has to be tested between variables under many different conditioning sets. The latter are slow because of the possible number of Bayesian networks; a naïve scoring with just 10 variables would have to consider around 10^{18} configurations (Robinson 1977).

Some recent approaches have speeded up the process by finding the network skeleton first and then doing local neighborhood learning to orient the skeleton edges, such as MMHC (Tsamardinos et al. 2006). Building the skeleton of a network is an easier task than orientating the edges as we only look at associations between variables and not the causal relationships between them.

We propose a method for fast structure learning based on finding the set of parents and children for each variable (skeleton), and then applying a committee of structure learners to make a joint decision about edge orientation. Some of the structure learning methods we use would be intractable when applied globally to a dataset with many variables, but can run rapidly at neighbourhood level. When the structure learners are based on different principles (e.g. a mixture of constraint-based and score-based) it is significant when they agree with each other, and in particular we find that this strategy gives good worst-case accuracy. This chapter can be summarised as follows:

i. We generalise previous work on restricting the search space to speed up structure learning;

ii. We present a novel local structure learning algorithm, EPC, specifically intended for analysing a target variable and its immediate neighbourhood;

iii. We show how different structure learners can be combined in a committee to give results with better consistency.

7.1.1 Skeleton discovery

The overall aim of skeleton discovery is to consider each variable in a dataset and find the set of directly neighbouring variables. To find the neighbourhood of one variable, we begin by considering all variables as potential neighbours and then filtering down this set in two phases. We first employ Relevance Learning Vector Quantization (RLVQ), a fast prototype based classification method, to do an initial feature selection for each variable. The variables found to have low relevance during this stage are removed from the estimated set of neighbours. We then apply the HITON algorithm on the resulting variables to narrow down this set.

LVQ and RLVQ are prototype-based classification methods applied in supervised learning and have been reviewed in earlier chapters. They employ a distance measure (typically Manhattan distance or quadratic Euclidean distance) that quantifies the similarity of a given feature vector with a prototype (representative) of any particular class.

Because the features have varied meanings and magnitudes in the data, quantifying their similarity by a uniform distance measure tends to be problematic. These differences are accounted by relevance learning schemes like RLVQ that employ adaptive scaling factors that scale the features based on their relevance for classification.

The RLVQ adapts the prototypes and the relevance factors for each training run through the data until the error rate is at a minimum. Further details on LVQ and

RLVQ can be obtained from Chapter 6.

LVQ-based methods have been used is several applications because they are intuitive, easy to implement and their complexity can be controlled by the user. For our purposes however we draw from the fact that they are very fast and have been shown to give high accuracy in identifying relevant features for classification in the case of RLVQ (Biehl et al. 2007).

Given i.i.d samples from a set of variables, we employ a fast two-phase skeleton discovery scheme to obtain the local neighborhoods of the different variables or features. Phase 1 uses Relevance Learning Vector Quantization (RLVQ) (Chapter 6), a variant of the LVQ family that not only does classification given labeled data, but also outputs a relevance profile of the variables relative to the classification. Because the features are statistically dependent on each other to different degrees, quantifying their similarity by a uniform distance measure (as seen in Chapter 2) tends to be problematic. These differences are accounted for by relevance learning schemes like RLVQ that employ adaptive scaling factors that scale the features based on their relevance for classification.

LVQ-based methods have been used is several applications because they are intuitive, easy to implement and their complexity can be controlled by the user. For our purposes however we draw from the fact that they are very fast and have been shown to give high accuracy in identifying relevant features for classification in the case of RLVQ (Biehl et al. 2007).

HITON is a standard algorithm for feature selection that, assuming the joint data distribution is faithful to a Bayesian network, carries out statistical tests on the data to determine the Markov boundary and the Markov blanket of a target variable. Advantages have been shown to accrue to HITON in contrast to other features selection algorithms: 1) it reduces the number of variables in the prediction models roughly by three orders of magnitude relative to the original variable set while improving or maintaining accuracy, and 2) it outperforms the baseline algorithms by selecting smaller variable sets than the baselines (Aliferis et al. 2003). Because HITON takes several hours to run for datasets with hundreds or thousands of variables, the RLVQ preprocessing step is useful to speed the process of obtaining Markov boundaries for each variable.

RLVQ is applicable to categorical variables. For the artificial datasets we used in the evaluation in Section 7.2.1 all features are represented as categorical variables. For the evaluation of real data from the LOCANET challenge in Section 7.2.2 all these datasets focus on predicting a target or getting the structure round a target variable. The target variable for these datasets is categorical.

To summarise, for each variable in the local neighbourhood a set of features relevant for its classification are obtained using RLVQ. For each of these sets of relevant features, the HITON algorithm is used to further narrow down the set of parents and children of the variable under consideration. Given this skeleton of undirected edges between variables, a committee of structure learning methods is then used to vote on the causes (parents) and effects (children), as described in the next section.

7.1.2 Causal discovery committee

Once a skeleton of the network is found we apply a structure learning committee for orientating edges between variables. We find that a combination of weak structure learners can be effective in recovering causal dependencies. Though such a formulation would be intractable for large problems at the global level, we show that it can run quickly when processing local neighbourhoods in turn.

The structure learning committee method takes the neighbourhood of each variable and applies different algorithms to determine whether each neighbour of that variable is a cause or an effect. If the majority of the algorithms determine that a given neighbour is a cause, then we classify it as a cause. Effects are classified in the same way. We do not apply any conflict resolution at the moment; our method might return bi-directional causes. Algorithm 4 shows the committee voting method.

Algorithm 4: Local Neighbourhoods Causal Discovery committee framework

```
Input : $\mathbf{c}, \mathbf{b_1}, \ldots, \mathbf{b_N}$, data vectors for target variable $C$ and set of features
        (parents/children) $B_1, \ldots, B_N$

foreach local neighbourhood i : $(B_{i1}, \ldots, B_{iN}, C_i)$ do
    foreach algorithm $Algo_j$ do        // PC,EPC,GES,MWST,LiNGAM,K2
        $C_i causes, C_i effects \leftarrow Algo_j\,(B_{i1}, \ldots, B_{iN}, C_i)$ ;
    end
    $\mathbf{C_i} \leftarrow$ majorityVote( $C_i causes$)        // vector of causes for
    target $C_i$ ;
    $\mathbf{E_i} \leftarrow$ majorityVote( $C_i effects$) ;
end
Return: {C,E}, Local structure of variables c
```

One of the methods in the committee is a novel causal discovery algorithm that uses partial correlation. In the next section, we describe the method and then highlight the rest of the committee members in the sections that follow.

7.1.3 Expected Partial Correlation (EPC) method

EPC is a local neighborhood structure discovery algorithm. Given the set of parents and children of a target variable, it returns a probability of each neighbourhood variable being either a cause or an effect. It is based on partial correlation as a measure of conditional independence, which is true in certain cases such as networks with jointly Gaussian distributions (Baba et al. 2004). We denote the Pearson correlation coefficient between A and B as ρ_{AB}, and the partial correlation between A and B conditioned on C as $\rho_{AB \cdot C}$.

The algorithm works by considering different three-variable subsets of the target C and its neighbourhood. There are three possibilities, excluding cycles: the collider or V-structure $(A \to C \leftarrow B)$; the chain $(A \to C \to B, A \leftarrow C \leftarrow B)$; and the fork $(A \leftarrow C \to B)$. The chain and the fork have the same conditional independency $A \perp\!\!\!\perp B \mid C$, while the V-structure has the unique property $A \perp\!\!\!\perp B$ but $A \not\perp\!\!\!\perp B \mid C$ (Chapter 5).

We can see that the V-structure is the only case where conditioning on the variable C increases the scale of the correlation between A and B, from the distribution of $|\rho_{AB \cdot C}| - |\rho_{AB}|$ in Figure 7.2.

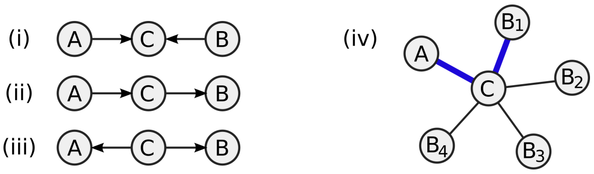


Figure 7.1: Possible 3-variable structures: (i) collider, (ii) chain, (iii) fork. Panel (iv) shows an example local neighbourhood for a variable C. The EPC algorithm orientates the edge AC by looking at the supporting evidence from each of the B_i's.

Given a particular sample size and type of distribution, we can work out what distribution of empirical correlation and partial correlation we expect from each different class. We show histograms of correlation and partial correlation in simulated networks in Figure 7.2. 10,000 binary models in each class (collider, chain, fork) were randomly created, with conditional probability tables sampled from the uniform distribution. The histogram in Figure 7.2 (right) gives us a probability distribution on the likelihood $P\left(\delta_{ABC} | class(A,B,C)\right)$, where $\delta_{ABC} = |\rho_{AB \cdot C}| - |\rho_{AB}|$ and $class(A,B,C)$ can be *collider* or *chain/fork*. By specifying priors on $P(class(A,B,C))$ we can then calculate the probability that A is a cause of C, using the assumption that in the *collider* class A is always a cause of C, whereas in the *chain/fork* class, there are 3 possible orientations, in only one of which A is a cause of C. While trying to

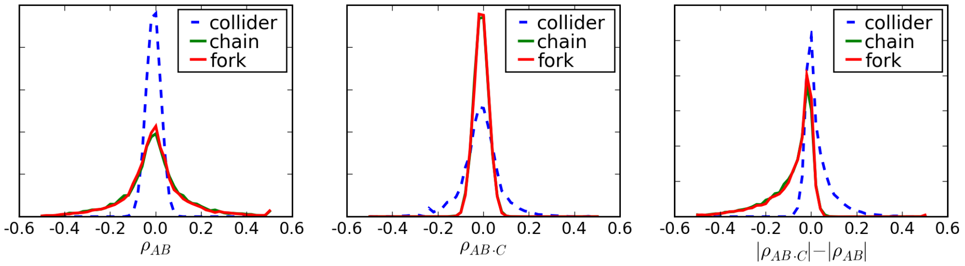


Figure 7.2: Histograms of correlation and partial correlation from 10,000 simulated 3-variable binary networks of each class, with 1000 samples drawn from each. Chains and forks have indistinguishable correlation distributions.

calculate whether A is a cause or an effect, we incorporate evidence from each of the B_i's in the neighbourhood and obtain $P\left(Cause(A,C)|\delta_{AB_1C}, \delta_{AB_2C}, \ldots, \delta_{AB_{N-1}C}\right)$ for a neighbourhood of size N, as illustrated in Figure 7.1 (right).

Algorithm 5: EPC Algorithm to distinguish between local causes and effects

```
Input : $\mathbf{c}, \mathbf{b_1}, \ldots, \mathbf{b_N}$, data vectors for target variable $C$ and set of
        parents/children $B_1, \ldots, B_N$.
        $P(Cause(B_i, C))$ for all $i$, priors for each $B_i$ being a cause of $C$

foreach variable $B_i$ do
    foreach variable $B_{j \neq i}$ ($B_i$ not a neighbour of $B_j$) do
        $\delta_{ij} \leftarrow |\rho_{B_iB_j\cdot C}| - |\rho_{B_iB_j}|$ ;
        Compute likelihoods $L(\delta_{ij}|class(B_i, B_j, C))$ ;
        where $class(B_i, B_j, C) \in \{$"collider', "chain/fork"$\}$ ;
    end
    $causeodds(i) \leftarrow$
    $P(Cause(B_i, C)) \prod_{i \neq j} (L(\delta_{ij}|\text{collider}) + L(\delta_{ij}|\text{chain/fork}))$ ;
    $effectodds(i) \leftarrow (1 - P(Cause(B_i, C))) \prod_{i \neq j} 2L(\delta_{ij}|\text{chain/fork})$ ;
    $P(Cause(B_i, C)|\mathbf{c}, \mathbf{b_1}, \ldots, \mathbf{b_N}) \leftarrow \frac{causeodds(i)}{causeodds(i)+effectodds(i)}$
end

Return: $P(Cause(B_i, C)|\mathbf{c}, \mathbf{b_1}, \ldots, \mathbf{b_N})$ for each $i$, posterior probabilities
        that each $B_i$ is a cause of $C$
```

The algorithm is limited to certain distributions, such as binary or Gaussian networks, where partial correlation is a reasonable measure of conditional independence. The method would fail in non-linear relationships between variables such as an XOR function. We also do not have an analytical form for the likelihood function; we currently have to estimate the distribution through simulations.

However the advantages of the algorithm are as follows. First, it is cheap to run: $O(N^2)$ in the neighbourhood size and $O(M)$ in the sample size. Second, it provides probabilities rather than categorical outputs – most methods based on CI constraints simply accept or reject a causal hypothesis. Third, we have the ability to incorporate prior beliefs about the orientations of edges. Fourth, it is useful as a committee member, as it gives high confidence when there is a V-structure and low confidence otherwise.

7.1.4 Other committee members

Standard algorithms were used in conjunction with EPC to form the structure learning committee. The strength of the committee method is derived from applying each of these methods to the same skeleton obtained from Section 7.1.1 for each dataset. These methods were selected to include methods based on different principles. These methods were as follows:

PC / IC

PC algorithm (Algorithm 3) is the reference causal discovery algorithm first introduced in 1993 by (Spirtes et al. 1993). It is a more detailed version of a very similar algorithm Inductive Causation (IC) introduced in 1991 (Pearl 2000). The PC algorithm is a constraint-based discovery algorithm that builds the structure based on statistical tests that evaluate the conditional dependence relationships between the different variables. Chapter 5 gives a description of this algorithm. A confidence level of 0.05 was used with this method.

K2

The K2 algorithm (Cooper and Herskovits 1992) is a probabilistic algorithm that maximizes structure probability given the data. It defines the Bayesian measure (BIC/BDeu) which is a quality measure of the network given the data. We use it in the committee to vote on whether a feature is an effect of the target variable only and not a cause because it is easier to specify a node order for the former. For our experiments we used the Bayesian Score (BIC) as our scoring function.

Maximum Weight Spanning Tree (MWST)

The MWST algorithm was introduced by Chow and Liu (Chow and Liu 1968) and is based on the maximum weight spanning tree. It essentially associates a weight with each edge obtained according to some similarity criterion (mutual information between variables or BDeu score) and then builds the maximum spanning tree of the obtained graph. For our experiments we used the mutual information between variables as a measure of (conditional) dependence.

Greedy Equivalent Search (GES)

The greedy search (GS) algorithm is an implementation of a standard optimization heuristic. Greedy Equivalent Search is an extension of the GS algorithm that optimizes searching the DAG space by searching in the Markov equivalent space. This method initially starts with an empty graph, adds arcs until the score cannot be improved then tries to suppress some irrelevant arcs (Munteanu and Bendou 2002). For our experiments we used the Bayesian Information Criterion (BIC) as our scoring function with an instantiation cache of 300.

LiNGAM

LiNGAM (Shimizu et al. 2006) is a more specific technique that attempts to discover the causal structure in linear non-Gaussian acyclic models. We include it in the committee because it provides a relatively different technique form the rest of the committee members and hence can account for certain distributions on which the other members may produce poor results. For our experiments default settings were used, as provided in the authors implementation.

7.2 Evaluation and results

Evaluation of our committee method was done in two ways: (i) Evaluation was done by comparing the discovered structure from the committee of algorithms with the output from the individual algorithms. This was done on six common data sets used in the field of causal discovery for testing structure learning algorithms. These datasets have known structures since they are generated from Bayesian networks with known distributions. (ii) Evaluation in the second part was done within the framework of the LOCANET causality challenge organized for WCCI 2008 (Guyon et al. 2008).

The causal structures found by our methods are evaluated using the same edit distance score based evaluation method used to evaluate the causality challenge entries. In this method, a confusion matrix C_{ij} that specifies the number of relatives confused for another type of relative is computed. It evaluates the 14 types of relatives in a depth-3 network. A cost matrix A_{ij} is also computed to account for the edit distance between the relatives. The edit distance specifies the number of substitutions, insertions, or deletions to go from one string to another. A score for a particular structure is then computed as $S = \sum_{ij} A_{ij}C_{ij}$.

7.2.1 Evaluation results using known causal structures

Datasets with known causal structure that have been cited in a lot of research work related to causality were used for the evaluation. These included the following:

LUCAS : LUCAS (LUng CAncer Simple set) (Guyon 2009) is a toy dataset generated artificially from causal Bayesian networks with binary variables. It represents a network where the target variable is Lung Cancer and the other variables are either not related to the target variable or related to it in some form (children, parents, grand parents, etc).

LUCAP : LUCAP (LUng CAncer set with Probes) (Guyon 2009) is the LUCAS dataset with probes added. Probes are artificial variables added to the dataset. Because probes are artificial variables they can be manipulated. The idea is to provide a dataset that can be used to evaluate causal algorithms in a situation where some of the data is manipulated.

ALARM : The ALARM (A Logical Alarm Reduction Mechanism) (Beinlich et al. 1989) dataset is a Bayesian network designed to provide an alarm message system for patient monitoring. It contains 37 discrete and binary variables.

ASIA : The ASIA dataset (Lauritzen and Spiegelhalter 1998) is a binary dataset that has been used in several causal studies. It represents an artificially created network about a person visiting Asia and the likelihood of contracting a prevalent disease in Asia. The dataset represents possible causes of the disease and possible effects of the disease in one causal network.

INSURANCE : The INSURANCE (Binder et al. 1997) dataset represents a network for evaluating car insurance risks. It contains 27 discrete and binary variables.

HAILFINDER : The HAILFINDER dataset (Abramson et al. 1996) is a Bayesian network designed to forecast severe summer hail in northeastern Colorado. It combines meteorological data with expert judgment, based on both experience and physical understanding, to forecast severe weather. It contains 56 discrete and binary features.

Method	Lucas (2000)	Lucap (2000)	Alarm (5000)	Asia (2000)	Insurance (2000)	Hailfinder (20000)
PC	1.91	2.14	2.43	2.08	2.81	1.79
EPC	**0.91**	**1.81**	**0.57**	2.94	2.2	2.2
GES	1.86	2.14	1.5	2.96	3.38	2.58
MWST	2.86	2.46	2.21	1.7	2.7	1.68
K2	2.18	1.95	2.1	1.78	**2.15**	1.79
LiNGAM	1.73	3.08	1.93	**1.38**	2.81	**1.43**
Comm (M)	1.65	1.9	1.07	2.86	2.46	2.39
Comm (U)	3.64	3.38	4	1.51	4	1.50
PC‡	2.91	3.38	2.72	3.29	2.81	2.73

Table 7.1: Evaluation of edit distances for various algorithms with known networks (sample size in brackets). Comm (M) denotes the committee decision with *Majority Voting* while Comm (U) denotes *Unanimous Voting* criteria. PC‡ represents results obtained on running the standard PC on the whole dataset. The best performing algorithm in each case is indicated in bold type.

The results for running the committee algorithms on these datasets are summarised in Table 7.1. The table shows the performance of the different standard methods and the EPC algorithm when applied in a local neighbourhood setting. Each of the algorithms in Table 7.1 are applied on the same skeleton (for each dataset) obtained using feature selection/reduction techniques discussed in Section 7.1.1. We then show the performance of all methods when combined in committee, using a majority voting scheme and unanimous voting scheme. The majority voting scheme is where an edge is oriented based on the majority vote of all committee members. The unanimous voting scheme is where all committee members need to vote the same for an edge to be oriented. As a benchmark we then show performance of the PC algorithm when applied globally to the whole datasets (with no localization).

For the benchmark datasets we find that the quality of the committee decisions is close to that of the best committee member in each case. For applications where high specificity is required, the unanimous voting strategy can be used which has low recall but high precision (few spurious causes are found).

Table 7.3 shows confusion matrices for the committee output of three of the datasets: Lucap, Alarm and Hailfinder. The confusion matrices give an idea of the recall and precision rates of the method. An ideal confusion matrix would be a diagonal matrix indicating the true positives and true negatives. The figures that are not on the diagonal represent the numbers of false positives (spurious causes, bot-

Dataset-Method	Precision	Recall	Fmeasure
Lucap-EPC	0.43	0.54	0.47
Lucap-Committee	0.30	0.33	0.32
Alarm-EPC	0.12	0.45	0.19
Alarm-Committee	0.12	1.00	0.22
Hailfinder-LiNGAM	0.07	0.07	0.07
Hailfinder-Committee	0.13	0.26	0.18

Table 7.2: Evaluation of Precision, Recall and Fmeasure scores for three of the datasets contrasting different individual algorithms with the committee

tom left) and the numbers of false negatives (spurious independencies, top right).

Execution time of the committee algorithm versus the PC algorithm applied globally to the whole dataset was compared for the HAILFINDER dataset. Using the standard PC algorithm structure discovery took 4452.4 seconds, while for the committee this time was 190 seconds for obtaining the skeleton and 13.3 seconds for obtaining the local graph, a total of 232.3 seconds, significantly less than the full global search.

Table 7.2 shows other performance metrics used for all the datasets. We calculate precision (ratio of true causes found to total causes found), recall (proportion of true causes found to actual number of causes), and Fmeasure = $\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$.

7.2.2 Evaluation results for the LOCANET challenge

The LOCANET (LOcal CAusal NETwork) challenge consisted of a number of tasks related to finding the local causal structure around a given target variable. This challenge was a follow-on challenge organised for the World Congress on Computational Intelligence (WCCI2008). The original challenge was to evaluate causal modeling techniques, with a focus on prediction when the data has been manipulated by an external agent (Guyon et al. 2008). LOCANET sought to discover the local neighbourhood of the target to a depth 3. The features in the local neighbourhood are generally more predictive of the target than any other features.

Of the four datasets in the competition, we tested the committee structure discovery method on three of these datasets (Guyon et al. 2009).

CINA : CINA (Census Is Not Adult) is a dataset derived from the census data from the UCI machine-learning repository Adult database (Asuncion et al. 1998). The data consists of census records for a number of individuals. The causal discovery task is to uncover the socio-economic factors affecting high income (the target value indicates whether the income exceeds 50K). The 14 original attributes (features) including age, work-class, education, education, marital status, occupation, native country were coded to eliminate categorical variables. Distractor features (artificially generated variables, which are not causes of the target) were added as well.

REGED : REGED (REsimulated Gene Expression Dataset) is a dataset used to find genes, which could be responsible of lung cancer. The data are re-simulated i.e. generated by a model derived from real human lung-cancer micro-array gene expression data. From the causal discovery point of view, it is important to separate genes whose activity cause lung cancer from those whose activity is a consequence of the disease.

SIDO : SIDO (SImple Drug Operation mechanisms) contains descriptors of molecules, which have been tested against the AIDS HIV virus. The target values indicate the molecular activity (+1 active, -1 inactive). The causal discovery task is to uncover causes of molecular activity among the molecule descriptors. This would help chemists in the design of new compounds, retaining activity, but having perhaps other desirable properties (less toxic, easier to administer). The molecular descriptors were generated programmatically from the three dimensional description of the molecule, with several programs used by pharmaceutical companies for QSAR studies (Quantitative Structure-Activity Relationship).

Table 7.4 shows the edit distance based scores for our challenge entry (the committee method) versus other entries. The scores for the other entries represent the maximum and minimum scores obtained for each dataset in the challenge. The lower the score the more faithful the discovered graph is to the original graph. Based on this score matrix, the committee method offers comparable performance as is seen in the table.

The majority vote of the committee is among the best of the results. The unanimous vote gives poor results in most cases, but good results in others. As would be expected, the recall is low using the unanimous vote (we only decide on a cause if all committee members agree), but precision is high (we find very few spurious causes). Overall, for applications where speed of execution is a primary feature, the committee method is the most appropriate.

	→	x
→	66	123
x	89	20458

(a) lucap2-EPC

	→	x
→	78	236
x	178	20244

(b) lucap2-Committee

	→	x
→	10	22
x	74	1263

(c) alarm-EPC

	→	x
→	18	18
x	131	1202

(d) alarm-Committee

	→	x
→	2	27
x	28	3079

(e) hail-LiNGAM

	→	x
→	7	27
x	45	3057

(f) hail-Committee

Table 7.3: Confusion matrices showing the algorithms with the winning edit distance and the corresponding majority committee decisions in terms of true causes found (top left), spurious causes (bottom left), true independencies (bottom right) and spurious independencies (top right) for Lucap (a-b), Alarm (c-d), and Hailfinder (e-f).

Dataset	Committee Score	Others (range)
CINA	2.32	1.70 - 3.31
REGED	0.22/0.439	0.27 - 0.50
SIDO	3.46	3.31- 3.48

Table 7.4: LOCANET challenge results showing the edit distance for the committee causal discovery method compared to submissions from other competitors. The submission from other competitors indicates the minimum to the maximum edit distance from all submissions. An edit distance of 0 indicates a perfect match of the discovered graph to the true graph.

7.3 Conclusion

From the challenge results we can see that our method gives performance comparable to other entries, while employing a method designed to give fast inference time. We provide two scores for the REGED dataset; the first one represents the initial submission where our method failed to find causes but due to the bias in the scoring statistic, this did quite well, the second represents a score obtained by the committee with the voting threshold decreased. For the benchmark datasets we find that the quality of the committee decisions is close to the best committee member in each case.

Results obtained for applying PC to whole datasets without localization generally indicate a lower accuracy rate than either PC with localization or the causal discovery committee. In principle to increase precision (at the expense of recall) we can increase the voting threshold upwards towards the unanimous voting level. Conversely it is also possible to increase the recall rate by altering the voting threshold in the opposite fashion. For applications where a causal relationship needs to be established with high precision, a unanimous voting scheme may be used though we have not so far analysed the accuracy of this approach.

The confusion matrices in Table 7.3 indicate that the committee generally obtains more true positives (higher recall - Table 7.2) than the corresponding committee member with the best average edit distance score. However the committee also generally obtains more false positives (lower precision) which accounts for the committee score not being as good as that of the best algorithm in each case.

Currently our method does not explicitly handle conflict of orientation so it is possible to have a situation where we find that $A \rightarrow B$ and also $B \rightarrow A$. The output is therefore not a DAG. We conjecture that finding bi-directional causes $B \leftrightarrow A$ may indicate the presence of a hidden variable which influences both A and B.

We have used our local committee framework with particular structure learning algorithms, but anticipate that other algorithms can be used in future work. Future research will also look at weighting the committee members based on derived properties of the dataset.

Chapter 8

Committee-based Independence Testing

Abstract

Statistical independence testing is an important operation in its own right and as a component of tasks such as constraint-based causal structure learning. We investigate the efficacy and improvement obtained from combining a battery of (conditional) independence tests for understanding the associative relationships between variables. We look at two approaches of combining the tests. In the first approach, denoted the hard-decision approach, the tests are combined with weights that are a function of features of the data to be tested. We train these weight functions using natural datasets from the UCI data repository, and show that our resulting adaptive committee independence test, Dependable Dependence 1 (DD1)*, has superior performance to any individual test. In this approach each test has to make the hard decision of dependence or independence given some dataset based on some p-value or significance level. We also demonstrate that this approach improves the performance of constraint-based causal structure learning algorithms. In the second approach denoted the soft-decision approach, each test returns a test statistic that can be related to the signal to noise ratio of the data which in turn is related to the strength of the dependence relationship. The committee test* Dependable Dependence 2 (DD2) *formed as a result of combining the different tests, returns a score related to the coefficient of determination (R^2) given by a regression model. Because the score is returned from a regression model we call this the soft-decision approach. This score is related to the nature of the dependence/independence relationship in the data. We also show that with this method,* DD2*, as well, superior performance over any one independence test in the committee is achieved.*

8.1 Introduction

Testing for independence between a set of observed variables is a fundamental operation in statistics, where the result of the test is of direct interest, and in machine learning, where it is the basis of tasks such as feature selection and constraint-based causal structure learning.

If we have two variables A and B of interest, then given i.i.d. samples $\mathcal{D}_{AB} \equiv \{a_i, b_i\}_{i=1}^{N}$ drawn from the joint distribution $P(A, B)$ there are many methods for estimating whether those variables are statistically independent. Each of these methods tends to have unique strengths and weaknesses. Some tests are based on a parametric prediction of A given B, in which case a dependency of a different parametric class might not be detected. Other tests are non-parametric, e.g. based on mutual information which requires that the densities of $P(A)$, $P(B)$ and $P(A, B)$ are estimated. More data is typically required to reach a conclusion than for a parametric test. This can be a limitation when testing for conditional dependence, particularly with multiple conditioning variables, as there may be only a few data points for each set of conditioning variables. In all dependence tests we additionally have to make the hard decision of setting a threshold of significance, and this can be a somewhat arbitrary choice.

Intuitively we try to choose the test which best matches the characteristics of the data. This is often practically carried out with rules of thumb, for example that rank correlation tests are suitable for small datasets, or that Pearson's test is useful if the data appears normally distributed. This can be inadequate for three reasons. First, in tasks such as constraint based structure learning there may be a large number of variables to be tested against each other, making it impossible to assess the most suitable test manually in each case. That is why it is typical in such problems to specify a single dependence test and significance threshold for all sets of variables; even though those variables may have very different properties. Second, in applications such as genotype-phenotype analysis, drug discovery, or the analysis of EEG data, datasets may have complex dependency types which are outside the scope of the classical tests and rules of thumb. In these cases we may have to consider more flexible methods such as kernel independence tests, but lack an automated method for making this choice. Third, some of the classical prescriptions can be too vague for the automation required in machine learning applications: "For small samples, we measure the correlation using the Spearman rank correlation coefficient" (Wasserman 2003, §16). How small is "small"?

To select an appropriate test, we propose to learn a mapping between the properties of a dataset and the reliability of different dependence tests on that dataset. We do this in order to automate the judgement about which tests are best suited for a particular dataset. We also propose a new dependence test that consists of a committee of several tests that are weighted based on their appropriateness for the dataset. The output of the committee is obtained as a vote over all the committee members.

We employ two approaches: a hard-decision based approach and a soft-decision based approach. In the so called hard-decision approach we use 504 sets of bivariate real data obtained from the UCI repository with known relationships (independence or dependence) and use these to learn weights for each test by calculating the probability that, given a particular dataset, the test will give the correct output (dependence or independence). These weights are aggregated in a weighted voting scheme in the *Dependable Dependence 1 (DD1)* test to output an independence or dependence relationship classification given a *new* bivariate dataset. This process requires that we make a hard decision on what significance thresholds the independence tests should employ and in this sense presents some limitations given the uncertainty in selecting the right threshold.

In the second approach we simulate different functional data and apply varying amounts of noise to these datasets to create a pool of training data. The amount of noise in each dataset can be expressed as the coefficient of determination (R^2). Increasing the noise in the functional data can be thought of as decreasing the dependence in the data. The test in this case, *Dependable Dependence 2 (DD2)*, is defined by a regression function that predicts the noise level or the R^2 score given some bivariate data. The regression algorithm is trained using data obtained from calculating properties of the synthetic datasets and results from running the individual tests on these datasets.

8.2 Hard-decision approach, DD1

8.2.1 Prediction of test performance

We begin by trying to relate the probability that a test gives the correct answer to features of the data to be tested. Independence between two variables $A \perp\!\!\!\perp B$ can be characterized as the joint distribution being equivalent to the products of the marginals,

$$P(A,B) = P(A)P(B)\ . \tag{8.1}$$

Factorizing the joint distribution, (8.1) yields a consequence of independence,

$$P(A|B) = P(A),\ \ P(B|A) = P(B)\ . \tag{8.2}$$

These two conditions lead to two different types of independence tests. A mutual information test can be seen in this context as assessing the uniformity of the ratio of the left-hand-side (lhs) and right-hand-side (rhs) of (8.1). As we only have samples drawn from A and B and no knowledge of their actual distributions, this requires density estimation of $P(A)$, $P(B)$ and $P(A,B)$. Extensions of this basic test include the least-squares independence test (LSIT) (Sugiyama 2011), which incorporates a cross-validated model selection procedure for density estimation.

A simple test based on equation (8.2) is the Pearson correlation coefficient, which evaluates whether there is a linear dependency between a pair of variables. In general, any regression or classification method (depending on the domains of A and B) can be used to test (8.2); if knowledge of B increases the predictability of A, or vice versa, then clearly (8.2) does not hold.

We now make two observations about independence testing. First, each test makes particular assumptions, either related to unsupervised learning for those testing condition (8.1), or related to supervised learning such as the possible parametric forms of dependencies for tests of (8.2). Second, choosing a single test before seeing the data loses information. Given assumptions about which types of dependencies might exist, a certain test has varying reliability at different points in the space of the data $\mathcal{A} \times \mathcal{B} \times N$. If we decide on a test t before knowing where $\mathcal{D}$ lies, and then use the data only to calculate the test output $t(\mathcal{D})$, the information about whether $\mathcal{D}$ lies in a region where the test performs favourably has been lost.

Consider an independence test $t : \mathcal{A} \times \mathcal{B} \times N \rightarrow \{0, 1\}$, and observed data $\mathcal{D}_{AB}$ drawn from some $P(A, B)$. Assume we know whether there is really a dependency between A and B, and this information is contained in the variable i_{AB} such that $i_{AB} = 1$ if $A \perp\!\!\!\perp B$ and $i_{AB} = 0$ otherwise. Assume we also have a set of features $x(\mathcal{D}) = \{x_1, \ldots, x_{d_x}\}$ summarising different characteristics of the data. We are interested first in calculating the probability that a test will give the correct answer given the dataset features:

$$p\left(t(\mathcal{D}_{AB}) = i_{AB} | x(\mathcal{D})\right) \ . \tag{8.3}$$

This can be done as follows:

1. Select a set of training instances, each of which comprises of observed data $\mathcal{D}_{AB}$ drawn from different variables A, B and a label i_{AB} denoting whether $A \perp\!\!\!\perp B$.

2. For each dataset $\mathcal{D}_{AB}$, generate a set of features $x = \{x_1, \ldots, x_{d_x}\}$ summarising its properties (e.g. dimensionality and measure of Gaussianity).

3. Run the independence test t on every dataset, and record whether $t(\mathcal{D}_{AB}) = i_{AB}$, that is, whether the test was correct.

4. For each independence test, train a classifier to predict whether the test will return the correct answer given features of the data.

In the experiments that follow, we use a Gaussian process classifier for step 4.

For our battery of tests, we attempted to select those based on different assumptions. The tests consist of distribution-bound tests which include the Pearson χ^2 test which assumes normality in the data and is useful for detecting linear dependencies, it also happens to be the most commonly used test in practice, used in constraint-based structure learning and Fishers test which is similar to the χ^2 but is more suitable for continuous data. We also employ distribution-free tests which include the kernel-based Hilbert-Schmidt independence criterion (HSIC) test which is a non-parametric test that utilizes cross-covariance operators on universal reproducing kernel Hilbert spaces (RKHSs). HSIC can detect non-linear dependency and works best with large sample sizes and small conditioning sets (Fukumizu et al. 2008, Gretton et al. 2008). Another distribution-free test we used is the Kernel-based Conditional Independence test (*KCI-test*) (Zhang et al. 2011) which is similar to the HSIC test but is more suited for small sample sizes. KCI-test can handle larger conditioning sets in the conditional independence (CI) case, and is similar to (Gretton et al. 2008) in the unconditional independence case. We also include tests based on partial correlation and conditional correlation (Baba et al. 2004). We list the tests used in this work in Table 8.1 with the corresponding significance thresholds used.

A total of 34 properties/features were extracted for each dataset. We assume that each dataset $\mathcal{D}$ is composed of two columns of data (many samples from $P(A, B)$). For each variable in a dataset the following features were extracted: measures of Gaussianity from the Jarque-Bera test, the Lilliefors test, and the Kolmogorov-Smirnov test, the kurtosis, the mean, variance, the type of variable (binary, multi) and the balance (a measure of the uniqueness of the variables in the data) ; a total of 11×2 features. From the whole dataset, we took the number of samples, the seven test statistics in Table 8.1 as well as the correlation coefficient as possible informative features for the regression. We also calculated the two principal components of the dataset and from each component we extracted the kurtosis, mean, variance and three measures of Gaussianity (Jarque-Bera, Lilliefors, and Kolmogorov-Smirnov). The type of features to extract was decided upon heuristically and since no feature selection was applied all features were used for the experiments.

We note that the use of meta-data to predict the performance of a portfolio of algorithms has been used for many settings other than independence testing. For example, similar studies have been done for algorithm selection in solving satisfiability problems (Xu et al. 2008).

8.2.2 Combining independence tests

We now look at combining all tests into a single adaptive test, which we call *Dependable Dependence 1 or DD1*. For a new dataset, we extract the 34 features and use

Dependence measure	Significance thresholds
χ^2	0.05
Fishers Exact	0.05
KCI-test	0.01
HSIC Kernel	0.01
Partial Correlation Ind Test	0.05
Pearson Correlation Test	0.05
Spearman Correlation Test	0.05

Table 8.1: Set of independence tests considered.

them to calculate the probability, for each test, of returning the correct dependence relationship output for $\mathcal{D}_{AB}$. These probabilities are normalised to give a set of weights. Weighted voting of the tests is done to obtain the overall output. For this, the weighted sum of individual test outputs is thresholded (e.g. at a significance level of 0.5) to provide a dependence estimate for the test dataset. This procedure is illustrated in Algorithm 6.

```
Algorithm 6: Dependable Dependence Test
Input : Data $\mathcal{D}_{AB}$, extracted features of Data, $x(\mathcal{D}_{AB})$
foreach Independence test $j \in \{, \ldots N_j\}$ tests do
    Calculate probability of test success: ;
    $p_j = p\left(t_j(\mathcal{D}_{AB}) = i_{AB} | x(\mathcal{D}_{AB})\right)$
end
Calculate normalised weights: $w_j = \frac{p_j}{\sum_k p_k}$ ;
Calculate weighted test output: $t = \sum_j w_j t_j(\mathcal{D})$;
Return: 0 if $t < threshold$, 1 otherwise.
```

This method for independence testing is equivalent to the "mixtures of local experts" approach in classification, the idea being to assign greater weight to the algorithms in our committee that we predict have a better chance of giving the correct answer. Committee machines, similar to ensemble methods such as bagging and boosting, tend to give superior performance to any of the algorithms that constitute them. This can be explained in terms of the bias and variance of each of the component tests. By decomposing the overall squared error of each test into the square of the bias plus the variance, when averaging many tests it can be shown that the bias terms (all positive) average and the variance terms (positive and negative) cancel out. The extent to which the variance terms cancel, and therefore the increase in accuracy that averaging brings about, depends on how uncorrelated the different tests are.

8.3 Soft-decision based approach, DD2

One of the main problems in dependence testing is that, while one can reliably reject independence, one can never fully reject dependence, because dependence can be arbitrarily weak. Making a hard decision on the relationship between two variables based on a p-value or some commonly used significance level adds additional uncertainty to the result. This is the limitation of the first approach. In the second approach the result of the committee test over a set of datasets can be viewed as a continuum over a range from dependence to independence. If the hard decision needs to be made, it can be an application specific threshold where one can choose a threshold based on some acceptable level of specificity or sensitivity. For example for some medical diagnoses where one may want to have a high specificity, an appropriately high threshold can be selected.

The dependence test in this case, *Dependable Dependence 2 or DD2* outputs a score related to the coefficient of determination (R^2) relative to a regression over the outputs from the different individual independence tests and the properties of the data. This score can be viewed as a measure of the signal to noise ratio in the dataset and by extension a measure of dependence or independence. For example if we assume the variables A, B are related by a function composed of a deterministic component and a stochastic component, R^2 is a score that quantifies the expected ratio of the magnitude of the deterministic part to the stochastic part.

For a measure of dependence to be considered sound it should fulfil the properties of generality and equitability (Reshef et al. 2011). Generality refers to the property that the test should capture a wide range of associations in data not limited to specific functional types. One advantage of the committee of tests is that it evens out any reliance on any specific functional types. The second property of equitability ensures that a test gives similar scores to different relationships possibly of different functional types but with equal noise levels in the data. An equitable test should give similar scores to functional relationships with similar R^2 (Reshef et al. 2011). For example noiseless functional relationships should have $R^2 = 1$. Adding noise to data can be looked at as *reducing the dependency* in the data.

We used simulated functional data in this approach to come up with training data that relates different functional data with different noise levels and used this in *DD2* to output a score related to the R^2. This was done as follows:

1. Simulate a set of training instances, each comprising of synthetic data, $\mathcal{D}_{AB}$

generated from different functional relationships, $f_{AB} : A \times B$, and a label R^2_{AB} denoting the noise level in the data.

2. For each dataset $\mathcal{D}_{AB}$, generate a set of features $x = \{x_1, \ldots, x_{d_x}\}$ summarising its properties. Features extracted also include results of standard independence tests.

3. Run independence test t on every dataset and record the label $t(\mathcal{D}_{AB}) = t_{\mathcal{D}_{AB}}$ the test statistic of that test.

4. Formulate the complete training dataset by appending $t_{\mathcal{D}_{AB}}$ to the properties of the data, x and a label R^2_{AB}.

5. Given the training data generated from (4.) above, train a regression algorithm to predict R^2 in a new dataset given its properties.

8.3.1 Generation of synthetic data

To obtain the training data, functional data was generated in a similar manner as (Reshef et al. 2011). The different functions used are listed in Table 8.2. For each function f in Table 8.2, we generated a set of 20 datasets $\mathcal{D}_i^{f_{AB}}$, $i = 1, \ldots, 20$. The 20 datasets per function were created by adding incrementally larger amounts of uniform noise to $\mathcal{D}_0^{f_{AB}}$ the dataset created from the noiseless functional relationship. The sample size of each dataset was uniformly random in the range [1, 1500]. For each dataset the coefficient of determination, R^2 was also calculated. A total of 320 datasets were created from this process. This data represents a wide scope of functional relationships and because we determine the amount of noise in each dataset we can use this data as a reliable training set for our independence test.

8.3.2 Formulating the DD2 test

For each of the 320 datasets $\mathcal{D}_i^{f_{AB}}$, a battery of 9 individual independence tests was applied. The tests included those in Table 8.1 with an additional four tests; the standard Kendall statistical hypothesis test, two tests based on the L1 statistic (Gretton and Gyorfi 2010) and the maximal information coefficient (MIC) test (Reshef et al. 2011) based on the maximal information-based non-parametric exploration (MINE) statistics. The test statistic for each of the 9 tests was recorded for each dataset. To create the training data, several data properties were extracted from each of the synthetic 320 datasets as explained in section 8.2.1. The results of the 9 tests were appended to the 34 features extracted from each dataset to form the complete training set.

Function (*f*)	**Function Description ($x \in \{0, 1\}$)**
Linear	$y = x$
Parabolic	$y = 4(x - \frac{1}{2})^2$
Cubic	$y = 128(x - \frac{1}{3})^3 - 48(x - \frac{1}{3})^2 - 12(x - \frac{1}{3}) + 2$
Exponential	$y = 10^{10x} - 1$
Linear/Periodic	$y = sin(10\pi x) + x$
Sinusoidal, Fourier Freq	$y = sin(16\pi x)$
Sinusoidal, non-Fourier Freq	$y = sin(13\pi x)$
Sinusoidal, Varying Freq	$y = sin(7\pi x(1 + x))$
Categorical	Pts ∈:$\{(1, 0.287), (2, 0.796), (3, 0.290), (5, 0.717)\}$
Random	y = rand()
Linear + Periodic, Low Freq	$y = \frac{1}{5} sin(4(2x - 1)) + 1.1(2x - 1)$
Linear + Periodic, Medium Freq	$y = sin(10\pi x) + x$
Linear + Periodic, High Freq	$y = \frac{1}{10} sin(10.6 * (2x - 1)) + 1.1(2x - 1)$
Linear + Periodic, High Freq 2	$y = \frac{1}{5} sin(10.6 * (2x - 1)) + 1.1(2x - 1)$
Non-Fourier	$y = cos(7\pi x)$
Cosine, High Freq	$y = cos(14\pi x)$

Table 8.2: Functions used to create the deterministic component of the training data.

The *DD2* test thus works as follows: given some bivariate data $\mathcal{D}_{AB}$ whose dependence relationship is sought, 34 features are extracted from the data that represent the meta properties of this data (e.g. dimensionality, measure of Gaussianity). An additional 9 features are generated from the test statistics of 9 individual independence tests applied to the data. A regression is then done using the training data derived from the synthetic data to determine a score related to the coefficient of determination R^2 which gives an idea of how significant the noise component of the functional relationship is to the stochastic component and in converse how dependent the variables A and B are on each other. Because it is a regression the output is a real number and depending on the domain application and the confidence level one requires, a threshold can be determined to categorise the relationship as independent or not. For the experiments we used a regression tree to do the regression.

We now describe experiments to compare the performance of our methods with the individual independence tests and with each other.

8.4 Experiments

In order to learn the mapping between dataset properties/features and the suitability of different independence tests to apply to that dataset, we need adequate training data. We use both synthetic data generated from different functional relationships and real-world data from the UCI repository (Asuncion et al. 1998) for this purpose. Training with synthetic data tends to be risky because relationships learnt are not necessarily representative of real-world data, however they provide a higher certainty about the dependence relationship represented in the data. The real-world data from the UCI repository is more representative of the real-world but we are also less certain about the relationships we assign to this data since we assign them intuitively based on some knowledge we have of the domain of the data.

For the real-world data, we selected datasets from the UCI repository for which we had sufficient domain knowledge to determine dependence relationships between the features in the data intuitively. To generate samples from the UCI datasets in which the dependence relation $A \not\perp\!\!\!\perp B$ holds, we defined adjacency matrices of directed graphs for each dataset. From these adjacency matrices we were able to calculate all pairs of unconditionally dependent variables and for each pair create two-column datasets of different lengths for all the datasets. In total 252 bivariate datasets that represent the dependent relationship, $A \not\perp\!\!\!\perp B$, were used.

Sample datasets that exhibit the independence relation $A \perp\!\!\!\perp B$ were obtained by randomly permuting one column of the datasets in the previous step. To increase our certainty of the independence relationship we again created separate independent datasets by swapping different columns of different datasets randomly and truncating them to uniform size. This resulted in a further 252 natural bivariate datasets that entail the (unconditional) independence relationship.
From the UCI repository the following datasets were used:

- *Abalone*, which describes features of an edible sea snail. This dataset was previously used for classification tasks where the task was to predict the age of the snails given information about the length, diameter of their shells, weight, height etc.

- *Diabetes*, Pima Indian database used in the prediction of whether a particular individual has diabetes or not.

- *Forest*, used in the prediction of what area of forest is burnt given data on the rainfall, temperature, rain, etc.

- *Heart*, relates the presence of heart disease in a patient to several measureable physical attributes of the patient such as age, sex, blood pressure, cholesterol levels, etc.

- *Car*, measures car acceptability relative to qualities of the car like number of doors, boot size, capacity, maintenance costs, etc.
- *House*, a dataset on USA congressional voting, relating affiliation of congress members (democrat, republican) to issues for which they may vote, for example immigration, crime, education spending.
- *Contraception*, relates a married womans contraceptive method choice based on their demographic and socio-economic characteristics.
- *Nursery*, used as a basis for obtaining the ranking of nursery school applications. Attributes are derived broadly from families' structure and financial standing, occupation of the parents and the social and health picture of the family.
- *Segment*, models image segmentation of several outdoor images and relates this to pixel attributes for example intensity, means of blue, red and green colors, etc.
- *Functional data*, used for functional learning and contains 352 bivariate numerical data sets collected from diverse sources.

With the 504 resulting bivariate datasets and accompanying labels of dependence or independence, we were able to test the accuracy of our adaptive independence testing. For the hard-decision approach, first we assessed such tests in isolation and without conditioning variables, then in a constraint-based causal structure learning problem, using the *DD1* test in place of a conventional (non-adaptive) dependence test. For the soft-decision approach, we used this real-world data as a test set for the *DD2* test and also compared the performance of this test to that of the *DD1* test.

8.4.1 Approach 1, DD1 experimental results

Independence testing

Given labelled datasets, training of classifiers to predict the probability of success of each test given features of the data $x(\mathcal{D})$ as described in section 8.2.1 was done. As a baseline for comparison, each of the independence tests in Table 8.1 was run for every dataset. The accuracy of each of the tests on the 504 datasets is shown in Table 8.3.

These results show a high performance in the tests based on correlation as a measure of independence (Pearson and Spearman). This may indicate that many

χ^2	*Fisher*	*KCI-test*	*HSIC*
0.713	0.537	0.818	0.874

Partial Corr	*Pearson*	*Spearman*	***DD1***
0.871	0.880	0.883	**0.888**

Table 8.3: Accuracy of 7 independence tests compared with leave-one-dataset-out cross validated accuracy of the DD1 test.

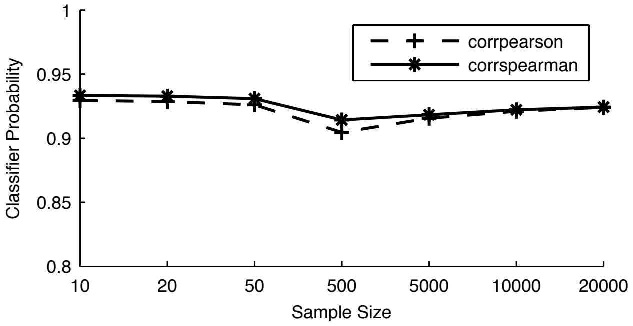
of the dependencies in the natural datasets were of an uncomplicated linear nature for which correlation based tests respond well. The KCI-test also performs well possibly because it is robust for small datasets which form the majority of the 504 datasets.

We evaluate the performance of our adaptive test, *Dependable Dependence 1, (DD1)* by training on data from all but one of the seven UCI datasets, then testing on the remaining dataset. This leave-one-dataset-out cross validation is repeated until each dataset has been tested. We avoid standard n-fold cross validation in order not to test the dependence of variables A, B having already seen the answer for A, C and B, C in the training data. The performance of DD1 on the 504 datasets is also shown in Table 8.3, superior to all the other tests, albeit marginally.

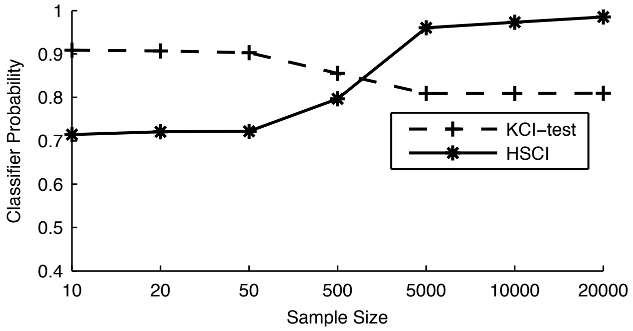
Figure 8.1 shows how different features influence the performance of our classifier for different independence tests. The classifier outputs the probability of the test obtaining the right output (independence or dependence) given the features of the data. To see the effect of only the sample size we hold all other features constant and vary the dimensionality ratio. From the Figure 8.1 (b)we observe that the classifier weights the HSIC and KCI-test algorithms based on their strengths; KCI-test tends to work best for small sample sizes while HSIC test is unreliable for small samples but performance improves significantly with increasing sample size. Interestingly, we see that despite the perceived wisdom that Spearman should be used for small sample sizes and Pearson for large sample sizes, we find that when our classifier is trained there is little difference between the two as shown in Figure 8.1 (a).

Constraint-based structure learning

The performance of any constraint-based structure learning algorithm is limited by the accuracy of the conditional independence test that it employs. Recently different versions of PC algorithms have been proposed that use different conditional independence tests to overcome the problems of the previously used tests for example (Sun et al. 2007), (Tillman et al. 2009) and (Zhang et al. 2011) that use kernel-



(a) Pearson Vs Spearman



(b) HSIC Vs KCI-test

Figure 8.1: Classifier response predicting the chance of tests being correct as a function of sample size (other features held constant).

based independence methods that are robust to non-linear, non-Gaussian data. Also (Margaritis 2005) has recently proposed a non-parametric recursive-median based test. Chapter 5 explains causal structure learning in greater depth; for our purposes here we focus on constraint algorithms that employ a conditional independence test for causal structure discovery. In particular we review modifications to the PC algorithm.

Ordinarily, in most causal structure discovery algorithms, one type of conditional independence (CI) test is selected and is assumed to be sufficient to test all pairs of variables in the dataset. Our work moves a step further to adapting the CI test for each different pair of variables within the same dataset that has to be tested. To the best of our knowledge this is the only work that has attempted to do this in the field of constraint-based causal structure learning.

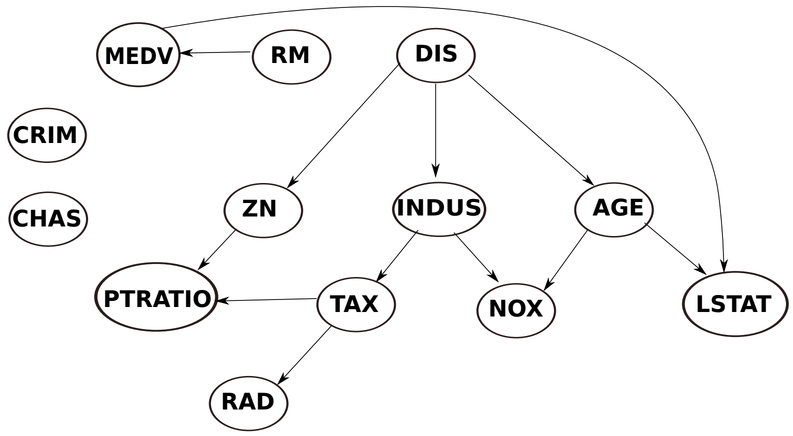


Figure 8.2: Discovered graph from the Boston-housing dataset using the *DD1* independence test in the PC algorithm.

Experimental work was based on the work of (Zhang et al. 2011) and (Margaritis 2005) where the PC algorithm was modified to use a different conditional independence test. Both these works tested their methods on the Boston-housing data from the UCI repository as an example of a real-world dataset. For comparison, we elected to do the same in our work. While (Zhang et al. 2011) chose to use only 12 of the variables (that were continuous), we chose to look at all the 14 variables. A description of the variables is available with the data.

The Boston-housing data was originally employed in a hedonic price model, based on the premise that the price of the property is determined by structural attributes (such as size (RM) and age (AGE) as well as neighborhood attributes (such as crime rate (CRIM), accessibility (DIS, RAD) and environmental factors (NOX)).

Figure 8.2 presents the discovered graph from the dataset. There is no standard ground truth for this data and so we compare our results with the results presented in the work of (Zhang et al. 2011) and (Margaritis 2005). Our method manages to discover the plausible links identified by (Zhang et al. 2011): the link between the average number of rooms (RM) and the median value of homes (MEDV), the link between percentage of the non-retail business acres-a proxy for industry (INDUS) and Nitrogen oxide concentrations (NOX). (Margaritis 2005) and the original work identify the absence of a link between NOX and MEDV and our method finds this relationship in the data as well.

While the intepretation of the ground truth of this graph is subjective, we find it quite intuitive that links are obtained for example between distance to major employment centers (DIS) and whether the land is used for industrial purposes (IN-

DUS), the age of the housing units (AGE) and the residential land zoned for lots (ZN). We also find that the strongest predictor of the value of a house (MEDV) is the size of the house (RM) which is also quite plausible. Both the other works find the link between the crime rate (CRIM) and the median value of houses (MEDV), a link we do not discover in our graph. This appears to be a false negative in our results. We find the link between the price of houses (MEDV) and residential zones present (ZN) quite plausible, despite not being found in (Margaritis 2005) and not evaluated in (Zhang et al. 2011).

In the absence of a ground truth and within limits of intuitive plausibility, we propose that our discovered graph is a more faithful representation of the underlying relationships within the data than the discovered graphs in (Zhang et al. 2011) and (Margaritis 2005).

8.4.2 Approach 2, DD2 experimental results

In approach 2, we derived the *Dependable Dependence 2 (DD2)* test from synthetic data based on some functional relationships with noise added. To test this method we used the 504 real-world datasets from the previous sections. Ideally any independence test should give a range of test statistics with independent data at one end and dependent data at the other end of the range. For example the *DD2* test gives a score of close to 1 for dependent data and close to 0 for independent data. While the range and scores vary for each of the individual tests, the pattern is the same.

In Figure 8.3 we illustrate the performance of three of the individual independence tests compared with the *DD2* test. Each plot represents the ratio(quotient) of the score obtained from running the tests on the 252 bivariate datasets exhibiting the dependence relationship to the score obtained from running the same test on 252 bivariate datasets exhibiting the independence relationship. Because the independent data are created as permutations of the dependent data, this is feasible. For the test output to be considered correct, the log of the quotient necessarily has to be greater than 0 (the threshold line cutting across the graph). The higher the point on the curve, the better the test differentiates dependence and independence relationships. As is evident, most tests tend to follow the pattern with a relatively good separation between independent and dependent data, however the *DD2* test seems to show better performance with fewer points below the red line.

From Figure 8.3 there is some indication that the *DD2* test gives better separation than some of the other tests. To confirm this observations, we constructed Receiver Operator Characteristic (ROC) curves for three of the individual tests and compared them with the *DD2* ROC curve. Figure 8.4 illustrates these ROC curves. The *DD2* test can be seen to have a bigger Area Under the Curve (AUC) implying it does

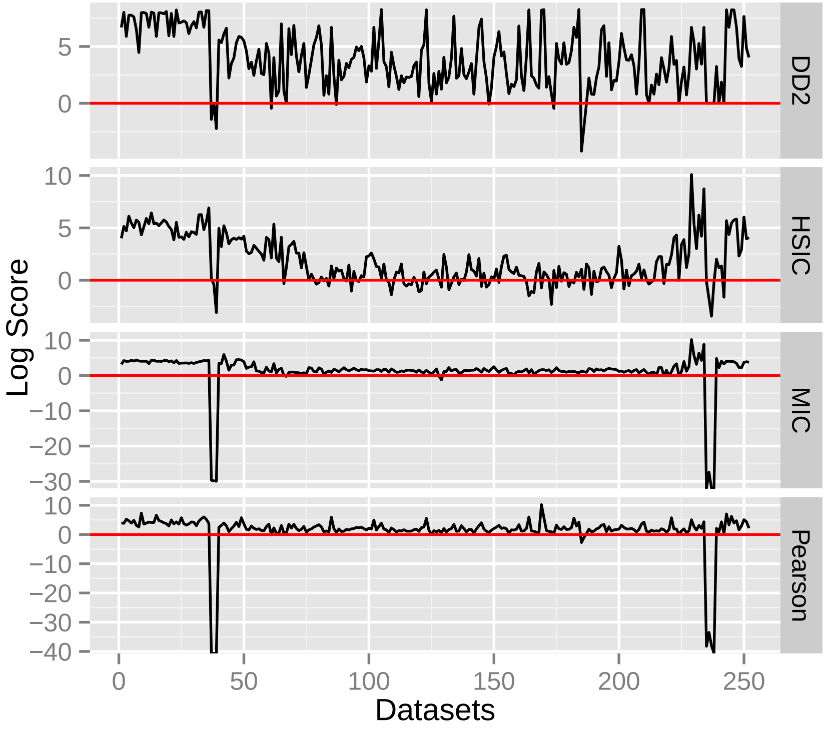


Figure 8.3: Comparative plots of results from a selection of four of the independence tests *Dependable Dependence 2*, (DD2), Kernel-based test, (HSIC), Maximal Information Coefficient (MIC) and the Pearson test, applied to the 252 dependent and independent bivariate datasets. The plot is of the ratio of the score obtained for dependence vs independence in all the 252 datasets. All points below the red line indicate errors in the test results of the particular test for those particular datasets i.e. a ratio below 0 on the log scale.

a better job at differentiating the independent and dependent datasets. The corresponding table of AUCs for all 10 tests and the DD1 test is shown in Table 8.4. The *DD2* also outperforms the *DD1* test albeit not significantly.

KCI-test	*HSIC*	*Partial Corr*	*L1*	*L1-Boot*
0.7403	0.7369	0.6229	0.8849	0.8849

Pearson	*Spearman*	*Kendall*	*MIC*	***DD1***	***DD2***
0.6229	0.6230	0.6219	0.8374	**0.8946**	**0.9010**

Table 8.4: AUCs for all the 10 tests and the *DD1* test obtained from running the tests on the 504 dependent and independent bivariate datasets.

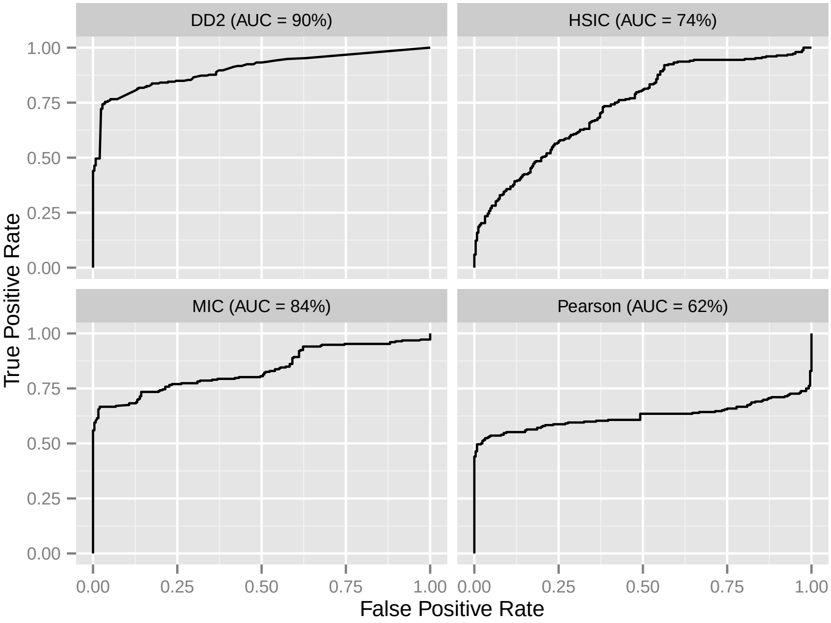


Figure 8.4: ROC curves corresponding to Figure 8.3 depicting Receiver Operator Characteristics (ROC) for three of the individual independence tests MIC, HSIC, Pearson and the *Dependable Dependence 2*, (DD2) test.

8.5 Conclusion

We have presented two methods, the hard-decision approach dependence test also called *DD1* and soft-decision approach test also called *DD2* of obtaining the independence status of two variables which adapts to the characteristics of the data being tested. For *DD1*, by learning which algorithms perform well in different regions of feature space, the tests in our committee can be appropriately weighted in order to give robust performance. In *DD2*, the committee of individual independence tests outputs a continuum of scores based on the amount of signal and noise in the data, that represent how (in)dependent two variables are. We have shown that indeed the *DD1* and *DD2* tests can perform better than the set of standard tests we looked at when tested on 504 real-world bivariate datasets and that the *DD2* test outperforms the *DD1* test albeit not too significantly. We have also shown that as a plug-in to a constraint-based causal discovery algorithm, more intuitive structures are discovered from a real-world dataset than when a fixed test is employed. We showed this for the *DD1* test; because the *DD2* slightly outperforms the *DD1* test, we expect that the results of applying the *DD2* test to the causal discovery problem will not be significantly different. This hypothesis may fail in the case that the two tests give similar results but get different test cases right or wrong. Theoretically however detecting dependence or dismissing dependence tends to be somewhat problematic because we have to make the hard decision about which threshold to use to differentiate dependence from independence.

Considering we are attempting to model the relationships between meta-data and test performance in real-world data, the amounts of data we used are relatively low, however we see superior performance even with these small numbers. Its our conviction that with more data and with the ever improving computing power, this can be a viable method particularly for fields like causal structure learning where causal relations for the most part are determined by the result of conditional independence tests. Another issue to be addressed in the current formulation concerns the weighting functions for approach 1; currently we use the normalised classification probabilities, though this might not be the optimal weighting for averaged voting at test time. Therefore another area of potential future analysis is to learn weights directly on the simplex.

Work based on:

E. Mwebaze, W. Okori, J.A. Quinn – *"Causal structure learning for famine prediction"* In Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI), Spring Symposium on AI for Development, 2010.

Chapter 9

Causal Structure Learning for Famine Prediction

Abstract

In this chapter, we discuss one practical application of causal discovery techniques in addressing the problem of food security. Food shortages are increasing in many areas of the world. Here we consider the problem of understanding the causal relationships between socio-economic factors in a developing-world household and the risk of experiencing famine (food insecurity) in that household. We analyse the extent to which it is possible to predict famine in a household based on these factors, looking at data collected from 5404 households in Uganda. To do this we use techniques discussed in chapter 7; a set of causal structure learning algorithms, employed as a committee that votes on the causal relationships between the variables. We contrast prediction accuracy of famine based on feature sets suggested by our prior knowledge and by the models we learn.

9.1 Background to the problem

Many inhabitants of developing countries are at risk of famine, and quantifying the risk under different circumstances is an important problem. One clear indicator of impending famine is the early stage of food shortage at the household level, known as *food insecurity*.

Household food security may be related to several socio-economic factors such as age, sex, marital status and education level of the household head, location, size of land, size of household, amount of labour available, possession of livestock, distance from the household to the main road, income and presence of agricultural shock (Okori et al. 2009, Webb and Yohannes 1999). In this study we analyze an extensive dataset of such variables collected from households in Uganda, and apply structure learning techniques in order to understand the causes of famine and predict which households are at risk.

To understand the causative relationships in the data, we employ a set of causal structure learning methods that are combined in a committee and vote on the causal relationships between the different variables. We present three candidate models for the causal relationships between the variables under study: 1) an *a priori* model, derived using our knowledge of the domain without reference to the data, 2) a model derived using statistical inference only, and 3) a model obtained by using both prior domain knowledge and statistical inference. We note that causal structure learning on natural datasets often makes most sense as an iterative process, combined with a structure prior from domain knowledge. Given the three models, we compare prediction accuracy of famine risk using corresponding feature sets.

Our work has implications for policy and intervention planning in food security, indicates how predictable food shortages are at the household level, and provides evidence that a combination of recently developed structure learning methods (which have mainly been analysed with synthetic data) are capable of producing intuitively plausible results from natural datasets.

9.2 Causal structure discovery

In Chapter 5, we reviewed structure learning algorithms and in Chapter 7 we presented a novel committee based causal structure discovery algorithm that we employ in this chapter to address the problem of food insecurity. The goal is to learn causal relationships from purely observational data, without being able to perform manipulations.

Learning causal relationships in a network of variables is not trivial, and the problem quickly becomes intractable for networks of more than a few variables (Robinson 1977). We simplify the task of searching through the large number of possible models by first finding an undirected graph between the variables (the skeleton), and then orientating the edges using the structure learning committee method described in Section 7.1.2.

The committee method combines the power of differently motivated structure discovery algorithms to determine the causal structure. In order to optimize the economics of execution time and accuracy, the committee method is initialized by skeleton discovery with feature reduction. We do this by first discovering the sets of parents and children for each variable in order to break the problem up into sets of tractable local neighbourhoods (skeleton discovery). We then apply a structure learning committee for orientating edges between the variables.

9.3 Famine data

The dataset used in this paper comprised of information collected by the Uganda Bureau of Statistics from 5404 households in four regions of Uganda (Central, Eastern, Northern and Western), spanning 57 districts of the approximately 80 districts of Uganda. The aim of collecting the data was to determine the degree to which households are susceptible to famine or food insecurity.

The original dataset has 24 features including household number, region, district, marital status, education level, occupation, sex, age of household head, household size, labour (estimate of annual labour per household), distance to the main road, distance to the garden(s), income, calories for season 1, calories for season 2, total calories, production for season 1, production for season 2, total production (quantity of agricultural produce for consumption by the family i.e. it excludes what gets sold), agricultural shock (includes pest attacks, weather e.g. floods, dry spells, irregular rains, and changes in the market e.g. sudden low prices), pest attack, livestock and food insecurity (famine propensity).

For this study several of the features were removed including household size, region, district, and income. Income was removed because it had several missing values. The calories and production for the two seasons were also removed and only the total calories/production maintained. Since food security is derived from total calories, total calories was redundant and also removed.

In all for this study a total of 13 features were used, listed in Table 9.1. Preprocessing reduced the data to 3094 households after deleting all the households with missing values in any of the 13 features. The preprocessed data was used in its entirety to discover the underlying causal structure around the target variable and later split into training and test data to test the prediction accuracy based on the whole dataset, and contrasted with prediction accuracy based on the derivative datasets from the derived causal graphs.

For individuals who are faced with food shortages, the caloric intake in their diet is reduced and this measure can be used as a proxy for food insecurity (IFPRI 1998, Salih 1994). The net caloric intake is calculated from the difference between the energy content (calories) of each agricultural food crop that is produced by a household and that utilized for different purposes like animal feeds and wastage within an agricultural year. For each household this net value is divided by the number of people in a household and number of days in a year to derive the dietary energy intake per person per day. Those households that fall short of a value of 1800 kilocalories per person per day in this study are considered as being food insecure.

Variable	**Type**	**Symbol**
Sex of the household head	*male/female*	Sex
Age of the household head	*years*	Age
Marital status of the household head	*married/divorced/ single/widowed*	MS
Size of household	*number of people*	HS
Size of land available to the household for farming	*acres*	LS
Amount of labour available for cultivation per year	*person-years*	La
Distance from household residence to the nearest main road	*km*	DR
Distance from household residence to farm land	*km*	DG
Total annual production of crops available for consumption by the household (excluding crops which are sold)	*kg*	TP
Agricultural shock (e.g. presence of flooding, drought, market fluctuation)	*true/false*	AS
Crops attacked by pests	*true/false*	PA
Ownership of livestock	*true/false*	Li
Household famine status (whether daily calorie intake per person in the household is above 1800 kCal)	*famine/not famine*	Fa

Table 9.1: Variables in the famine dataset describing each household surveyed.

9.4 Experiments and results

Experiments were carried out on three different configurations of the famine network.

9.4.1 *A Priori* graph

A graph was first constructed based on the authors' knowledge of the domain, without reference to the data. The structure is shown in Figure 9.1.

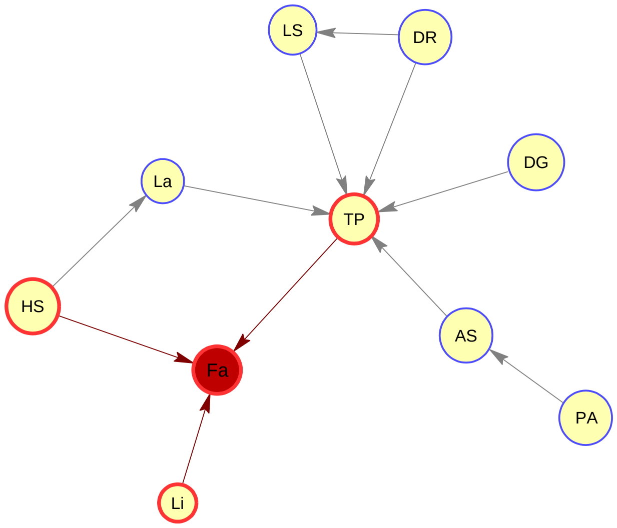


Figure 9.1: Intuitive *a priori* graph of causal relationships in the famine dataset. Features are represented as **HS**-Household Size, **LS**-Land Size, **La**-Labour, **DR**-Distance to main Road, **DG**-Distance to Garden, **TP**-Total Production, **AS**-Agricultural Shock, **PA**-Pest Attack, **Li**-Livestock, and **Fa**-Famine. The rest were assumed to be independent.

The intuition behind the causal relationships in Figure 9.1 is as follows:

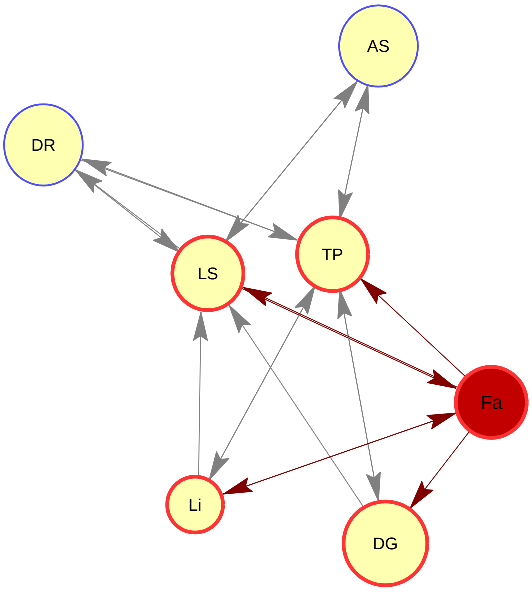
TP → **Fa :** The more food that a family produces, the more is available for direct consumption.

Li → **Fa :** Owning livestock is a direct mitigation of famine risk, e.g. consumption of milk and eggs increases calorie intake.

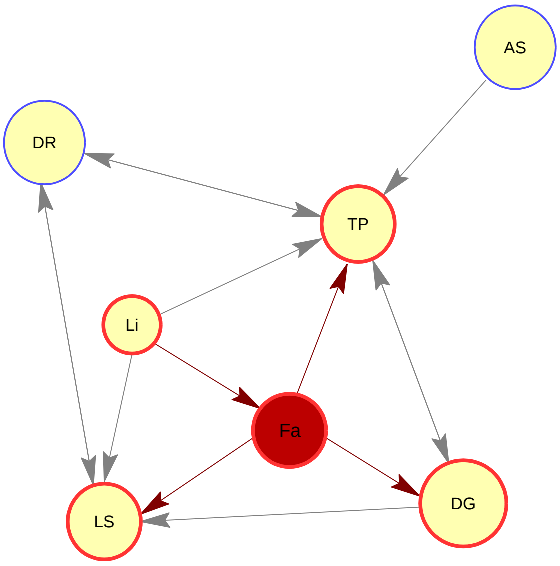
HS → **Fa :** The greater the size of the household, the smaller the share of consumable produce per person.

La → **TP :** The greater the size of the household, the more manpower is available for raising crops.

LS → **TP :** The more land available to a family for farming, the more potential they have for growing crops.



(a) Graph learnt with uninformative prior



(b) *A posteriori* graph

Figure 9.2: Inferred and *a posteriori* causal graphs depicting causal relationships in the famine data set. The highlighted nodes, with thicker edges, represent nodes directly connected to the target variable *famine/food insecurity*. Features are represented as **HS**-Household Size, **LS**-Land Size, **La**-Labour, **DR**-Distance to main Road, **DG**-Distance to Garden, **TP**-Total Production, **AS**-Agricultural Shock, **PA**-Pest Attack, **Li**-Livestock, and **Fa**-Famine.

DR → TP : Households closer to the road are more likely to sell produce rather than keep it for consumption.

DR → LS : Land closer to the roads is more expensive, making typical plots of land smaller.

DG → TP : More time required to travel and transporting goods to and from the farm means that less time is available for work on cultivation.

AS → TP : Agricultural shock (e.g. flooding, drought, market fluctuation) directly affects production.

PA → AS : Pest attack is part of agricultural shock by definition.

HS → La : The more people in the household, the more manpower is available for food cultivation.

9.4.2 Graph learnt with uninformative prior

The discovered graph was learned from the data using the committee method of causal discovery, without incorporating any prior domain knowledge. It is shown in Figure 9.2 (a). The highlighted nodes, with thicker edges, $\{LS, DG, TP, Li\}$ represent the direct causes and effects of our target variable *famine*$\{Fa\}$. It is interesting to note that the discovered graph has as its direct causes *landsize* and *livestock* as the major determinants of whether a household is likely to face famine in a given year or not, which are quite intuitively plausible causes.

Table 9.2 shows how each algorithm voted on the neighbours of the target variable. It represents how confident the committee was about each of the edges related to famine/food security. It is interesting to note that two algorithms EPC and K2 agree exactly on the same set of causes while LiNGAM that assumes non-Gaussian distributions does not find any causes. A voting threshold of 2 was used to obtain the uninformed graph depicted in Figure 9.2 (a).

9.4.3 *A Posteriori* graph

The *a posteriori* graph was obtained by including the *a priori* graph in to the committee and doing the voting again. The effect of this inclusion is to strengthen the vote on the prior causative links in the graph. A voting threshold of 3 was used for the *a posteriori* graph, and we double the weight of votes from the prior knowledge model (the weight of votes determines our confidence in the structure learning as opposed to our assessment of causes from prior knowledge). The graph is depicted in Figure 9.2 (b). Intuitive explanations for some of these causal relationships is as follows:

Edge	PC	EPC	MWST	GES	LiNGAM	K2
LS→Fa	1	0	0	1	0	0
Fa→LS	0	1	0	1	0	1
Fa→DG	0	1	1	0	0	1
Fa→TP	0	1	0	1	0	1
Li→Fa	1	0	1	0	0	0
Fa→Li	0	1	0	0	0	1

Table 9.2: Table showing relative voting strengths of the different committee algorithms for causative edges related to famine. The numbers represent votes per algorithm for each edge. The committee members are described in detail in Chapter 7.

Li → LS : The more livestock a household has the more likely they are to look for a larger piece of land, or conversely, the more land a household has, the more likely they are to have livestock.

Fa → TP : This is somewhat counter-intuitive, as we expect the reverse that low total production of crops leads to a state of famine. However this causal relationship does have a plausible interpretation. Total production is the amount of crops produced excluding those which are sold. During a shortage, families often cut down to one meal a day in order to conserve their resources. If they produce perishable crops then they are likely to sell the rest in order to build up a financial buffer.

Fa → LS,DG : In these cases we would also expect the most probable relationships would be the reverse. It is plausible however that some members of the committee may lay greater emphasis on the effects of famine than the causes for instance K2 is a causal structure algorithm that theoretically will orient edges differently based on the ordering of the features. It is hence likely that if the target $\{Fa\}$ is analysed first, the algorithm will give a stronger weight to the effectual relationship of consecutive features with the target.

Another interesting result is that we obtain several intuitive causes of variable $\{TP\}$, the total production. The fact that the two derived graphs depict some intuitive causative characteristics is interesting because it provides some support that the methods used are able to derive plausible causes from observational data. The bidirectional nature of some relationships arises because for a natural dataset like the one we used, the features intrinsically and intuitively tend to have bidirectional causal influence depending on which one manifests first. A bidirectional link can also indicate the presence of a hidden cause, which influences both of the observed variables.

Note that it is difficult to find relationships such as $PA \rightarrow AS$ statistically. As there is only one cause it is difficult to test with conditional independence.

9.5 Classification of famine risk

We further tested the accuracy of our true graph and derived graphs by splitting the data into training and testing data, training several models on a training set and measuring their prediction accuracy on the test set. Four datasets were used; the full dataset and three datasets derived from the direct causes and effects of the target based on the *a priori* graph, the inferred graph (uninformative graph) and the *a posteriori* graph (informed graph). Figure 9.3 shows the prediction accuracy measured as area under the curve (AUC) for various classifiers. The types of classifiers were chosen to comprise a wide scope of different classification techniques. Mostly standard parameters were used for the parametric classifiers.

Results from Figure 9.3 indicate that a reduced featureset consisting of parents of the target variable has implications in improving prediction accuracy. Results from a reduced featureset provide comparable classification/prediction accuracy relative to the whole featureset. Not only is this advantageous in the predictive/classification case, but it also has advantages in reducing the amount of data collected in the field for such a survey due to a reduced featureset. A reduced set of features, which we believe are direct causes and effects of the target variable, are also more robust in their performance under conditions when there may be interventions on some of the variables (e.g due to humanitarian relief or implementation of new agricultural policies). This has been investigated in Chapter 6.

9.6 Conclusion

This chapter presents results in the application of causal discovery to famine prediction in Uganda. The causal models we derive can be used to inform policy making with regards to household food security in a country like Uganda where food security or famine is a threat to many households, and indicate the extent to which household food shortages can be predicted based on socio-economic factors.

Our results provide some evidence in the understanding of famine specifically food insecurity in typical households in some districts of Uganda. We have also shown that prediction accuracy is maintained using the variables that we find to be direct causes or effects. This has implications for the number of variables which may

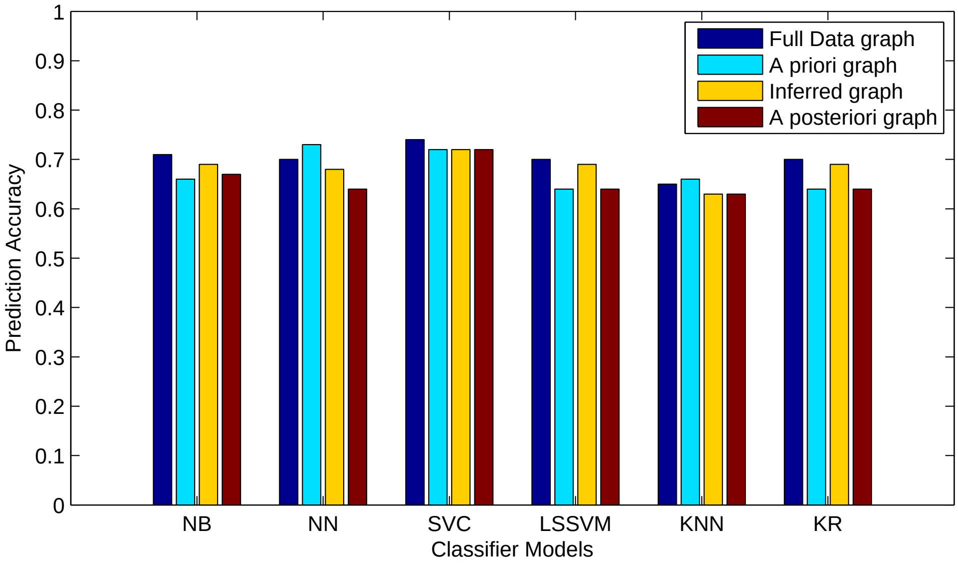


Figure 9.3: AUC scores of various standard classifiers with a complete featureset (Full Data graph) and reduced sets of causally related features with the target (*a priori* graph, inferred graph and *a posterior* graph). Algorithms used include: NB-Naive Bayes, NN-Neural Network, SVC-Support Vector Classifier with a linear kernel, LSSVM-Least Squares Support Vector Classifier, KNN-K-Nearest Neighbour and KR-Kernel Ridge regression classifier

need to be collected in future studies. Where we find bidirectional causes we postulate that there may be confounding variables and more data needs to be collected.

Concerning prediction performance we reason that inferences based on variables which have a direct causal link with the target variables are more robust under manipulations of the variables. A density modelling approach to prediction using the entire dataset may not be reliable if the data distribution is shifted due to some external intervention, for example due to a relief effort. Predictions based on true direct causes of a target variable can be expected to have the same reliability whether or not those causes have been manipulated. Chapter 6 discusses a novel algorithm whose classification accuracy is improved by consideration of causally relevant features.

Some key extensions to this work in the future include calculation of local district level causal graphs and making the comparisons between these different graphs across different districts, across different famine-prone and non famine-prone districts, different regions, etc. Also looking at district level confounding data to understand the dynamics of food insecurity at the different districts and regions will be a focus of any extensions to this work as well as including other data for example satellite observation data.

It would also be useful to formulate likelihood terms for the members in the causal committee, and produce a more sophisticated structure prior so that we can use Bayesian methods to infer the posterior model.

Work based on:

J.A. Quinn, K. Leyton-Brown, E. Mwebaze, – *"Modeling and monitoring crop disease in developing countries"*, In Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI), Conference on Computational Sustainability, 2011.

J.R. Aduwo, E. Mwebaze, J.A. Quinn – *"Automated vision based diagnosis of cassava mosaic disease"*, in Proceedings of ICDM Workshop on Data Mining in Agriculture, 2010.

Chapter 10

Implementation of a Crop Disease Surveillance System

Abstract

Information about the spread of crop disease is vital in developing countries, and as a result the governments of such countries devote scarce resources to gathering such data. Unfortunately, current surveys tend to be slow and expensive, and hence also tend to gather insufficient quantities of data. In this chapter we describe two general methods for improving the use of survey resources by performing data collection with mobile devices and by diagnosing crop diseases through the application of AI techniques. First, we describe how we implemented a mobile-based data collection surveillance system that enables real-time mapping of incidence counts as well as displaying a country wide severity map. Second, we demonstrate that the diagnosis of plant disease and vector counting can be automated using images taken by a camera phone, enabling data collection by survey workers with only basic training. We have applied our methods to the specific challenge of viral cassava disease monitoring in Uganda, for which we have implemented a real-time mobile survey system that was piloted at the National Crop Resources Research Institute in Uganda.

10.1 The problem

Cassava is the third largest source of carbohydrates for human consumption worldwide, providing more food calories per cultivated acre than any other staple crop. It is an extremely robust plant which tolerates drought and low quality soil and is a source of food to a large portion of people within the East African region. The foremost cause of yield loss for this crop is viral disease (Otim-Nape et al. 2005), a major factor keeping East African farmers trapped in poverty (The Economist, (Anonymous 2011)).

The economies of many developing countries are dominated by an agricultural sector in which small-scale and subsistence farmers are responsible for most production, utilizing relatively low levels of agricultural technology. As a result, disease among staple crops presents a serious risk, with the potential for devastating consequences. It is therefore critical to monitor the spread of crop disease, allowing targeted interventions and foreknowledge of famine risk.

Currently, teams of trained agriculturalists are sent to visit areas of cultivation across the country and make assessments of crop health. A combination of factors conspire to make this process expensive, untimely and inadequate, including the scarcity of suitably trained staff, the logistical difficulty of transport, and the time required to coordinate paper reports. The problems can be expressed under three broad categories.

First, the surveillance visits to remote districts in the country are constrained by available funds for the activity, the large geographical area and limited skilled labour. Uganda presently has approximately 100 districts with about 95 % engaged in small scale cassava farming. Normally the survey is conducted by a set of teams of experts, usually just under 10 teams and spans a period of 3 weeks. The allocated budget for this annual survey is approximately USD 60,000. Coupled with other constraints for example poor road infrastructure in the remote areas of the country, the surveyors only sample a few districts within the four regions of the country. The results from these four regions are used to generalize over the whole country. Even in the districts that are selected for the survey, only a portion of the gardens sampled along the main road through the district are surveyed.

Second, the relatively highly specialized task of visually diagnosing plants in remote fields is made especially more problematic because of the limited skilled personnel. Reliable automatic methods for performing surveys therefore offer the possibility of extending the scope of disease surveillance. The ubiquity of camera phones in even the most rural parts of many developing countries introduces the possibility of survey by non-expert workers, who submit images of crops that are then automatically classified. We investigate computer vision techniques for using camera-enabled mobile devices to make disease diagnoses directly, allowing reliance on survey workers with lower levels of training, and hence reducing survey costs. Specifically, given expert-annotated images of single cassava leaves, we demonstrate classification based on color and shape information detailed in Chapters 3 and 4.

Third, the data eventually obtained from the survey is presented as a static map that under-represents the collected data. Roughly four main cassava diseases are investigated during the survey. For each of these diseases the incidence (presence) of the disease is recorded and a severity measure (1 - 5) is recorded as well. Corre-

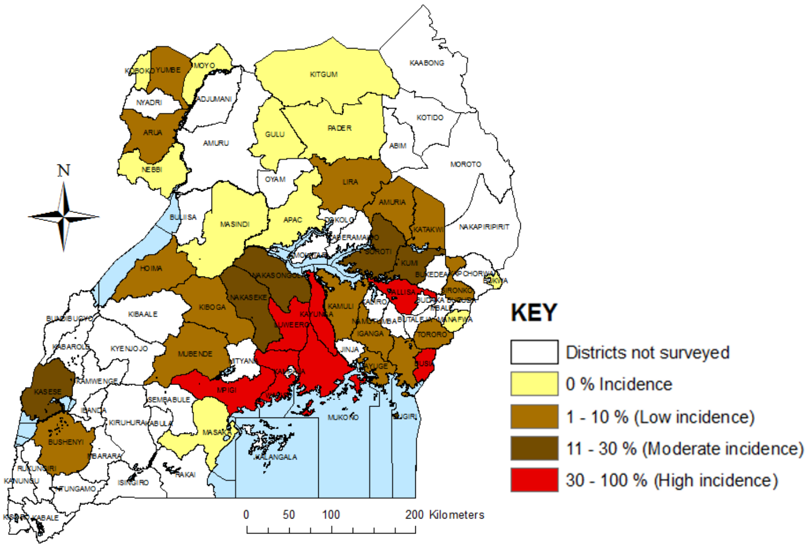


Figure 10.1: Incidence map of cassava brown streak disease in Uganda, 2009 (NACCRI).

sponding geo-coordinates are also collected for each record. An example of such a map is shown in Figure 10.1 where the incidence of Cassava Brown Streak disease (CBSD) for the year 2009 is portrayed. The map helps the concerned agricultural institutions in the country plan targeted interventions to specific areas in the country to control the spread of a possible epidemic or salvage farmers yield. Unfortunately the time it takes to get this data transcribed from paper forms and input into a GIS system to produce a map like this normally renders any interventions based on this data fruitless.

10.2 Improved survey system design

We designed and implemented a surveillance system based on mobile data collection with real-time mapping with the added component of automated diagnosis of crop diseases. Figure 10.2 illustrates how the system operates. An enumerator or an extension worker in the field takes a photo of the plants in the garden, auto-diagnosis on the phone or server is done and the image is shown on a map in real-time.

The transition from the current system to the improved surveillance system was executed as follows:

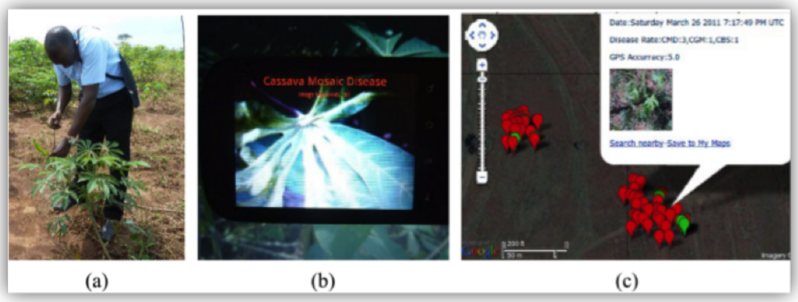

Cassava Mosaic Disease
Date:Saturday March 26 2011 7:17:49 PM UTC
Disease Rate:CMD:3,CGM:1,CBS:1
GPS Accuracy:5.0
Search nearby - Save to My Maps
(a)
(b)
(c)

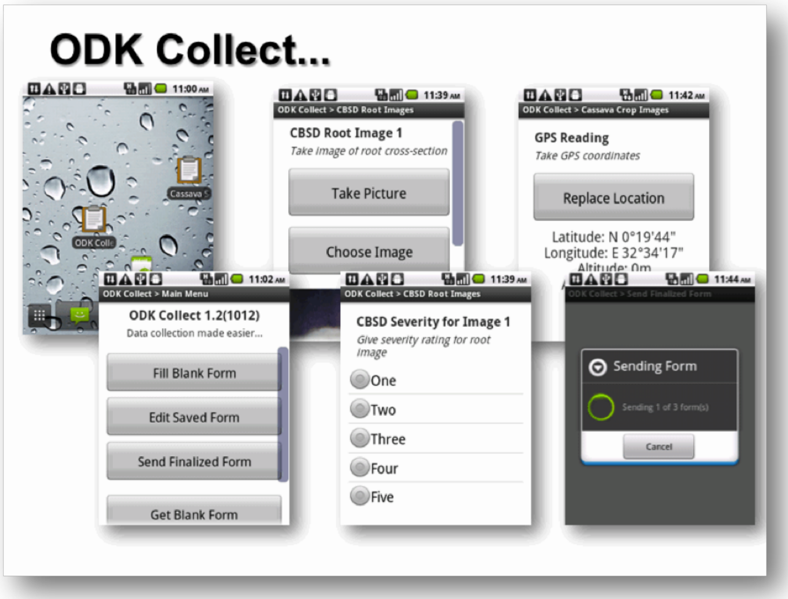
Figure 10.2: Surveillance System depicting (a) an extension worker taking a photo of crops in the field, (b) automated diagnosis being done by the software on the phone and (c) real-time mapping of the disease incidence on an online map.

1. Replace the paper forms with low-cost mobile phones running Android OS.
2. Using the existing telecommunications network get geo-tagged survey data directly on to a map in real time from data collectors in the field.
3. Using the available processing power on the phones, automate the cumbersome tasks of whitefly count and severity scoring on the phone so experts need not be the ones to carry out the survey and can use their limited time doing something else.

10.3 Mobile data collection and mapping

We developed a mobile-based survey system that allows surveyors to collect geo-tagged images in addition to other survey parameters and seamlessly present the data to an online map in real time. A version of the system can be seen at `http://cropmonitoring.appspot.com`. We expect that our system will be used in the up-coming crop survey by National Crop Resources Research Institute (NACRRI).

The system implementation was based on the Open Data Kit (ODK) (Anokwa et al. 2009) suite of applications, the Google AppEngine, the Google Map API and Google Fusion Tables. We used the *ODK-build* component of ODK to build forms usable on the mobile devices (converting the paper form to an appropriate mobile format). *ODK-Collect* is a component of ODK that is used on the mobile device to enable data collection, upload and download of new electronic forms. It was used to collect the data on the mobile devices.



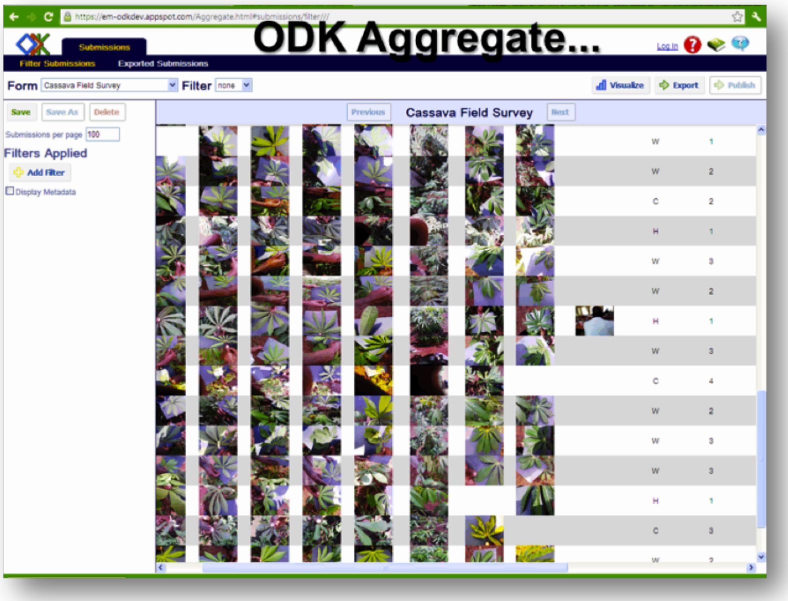
Latitude: N 0°19'44"
Longitude: E 32°34'17"
Altitude: 0m

Figure 10.3: Mobile phone implementation of the surveillance tool used for collecting geo-coded incidence data from the field

ODK is presently only compatible with devices running Android OS. Figure 10.3 illustrates some of the mobile phone screens from the conversion of the paper form for the mobile electronic version. ODK makes it easy to integrate images, geo-coordinates and other data into one data record.

The Google suite of tools were used to implement the backend server side of the system (Google AppEngine + *ODK-Aggregate*), and the front-end mapping system (Google Maps API + Fusion Tables).

To test the usability of the system, surveyors from NaCRRI collected some test data from several cassava fields. Figure 10.4 shows the Google AppEngine server side of the data collected by the surveyors. As is evident, images were taken in the gardens with a lot of clutter. We explain how we did the auto-diagnosis from such images in the next section. The real-time mapping of the data was done using Google Map API with Fusion Tables. Figure 10.5 illustrates some of the data as it came in. Clicking on any one blob in the map brings a pop-up that shows the image associated with that blob as well as the incidence and severity information.



Form Cassava Field Survey
Filter none
Submissions per page 100

W	1
W	2
C	2
H	1
W	3
W	2
H	1
W	3
C	4
W	2
W	3
W	3
H	1
C	3
W	2

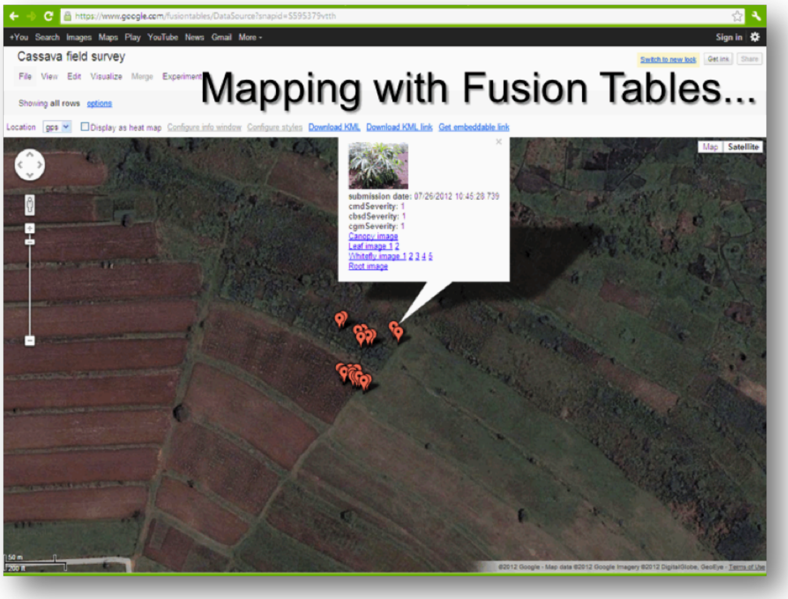
Figure 10.4: Mobile phone implementation of the surveillance tool used for collecting geo-coded incidence data from the field

10.4 Image-based diagnosis of crop disease

Diagnosing diseases from plant images requires that appropriate features are extracted from the images that represent each disease uniquely. Once features are extracted an appropriate classifier needs to be identified that can be trained on the features such that it can diagnose new leaf images accurately.

We employ a prototype-based classification scheme that identifies representative prototypes for each class of disease in the image data. During the training phase, the prototypes are updated to optimize an objective function that minimizes the distance of a particular prototype from examples of the same class and maximizes the distance to example data from a different class of disease.

Because of the complexity of identifying multiple diseases in this case, we extract different features and combine them using different distance measures in a combined-distance Learning Vector Quantization scheme that also optimizes a matrix of correlations between the different distance measures. These methods are explained in Chapters 3 and 4. For our purposes here we give a concise explanation of the practical aspects of using these methods including data collection, feature



submission date: 07/26/2012 10:45:28.739
cmdSeverity: 1
cbsdSeverity: 1
cgmSeverity: 1
Canopy image
Leaf image 1 2
Whitefly image 1 2 3 4 5
Root image

Figure 10.5: Mobile phone implementation of the surveillance tool used for collecting geo-coded incidence data from the field

extraction and how we deployed the methods on mobile devices.

10.4.1 Data collection

Image samples of cassava leaves used in the training of the classifier were captured from the National Crops Resources Research Institute, Uganda. We collected sample leaves from three different plantations, placed each leaf on a light box and captured images with a standard digital camera, at a resolution of 3072×2304. Leaf images were captured from 92 healthy plants and 101 plants infected with cassava mosaic disease. Examples of these images are shown in Figure 10.6.

10.4.2 Feature extraction

The images taken in the lab in this case have one uniform light background. With a light background, it is therefore straightforward to remove the background from the image by looking at intensity values. In Chapter 4 we also discuss some work done with natural images taken *in situ* with varied backgrounds.

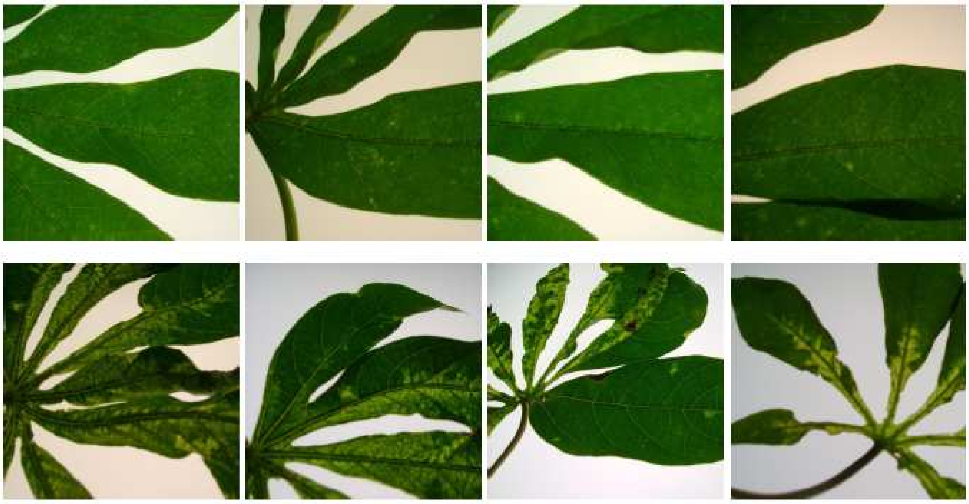
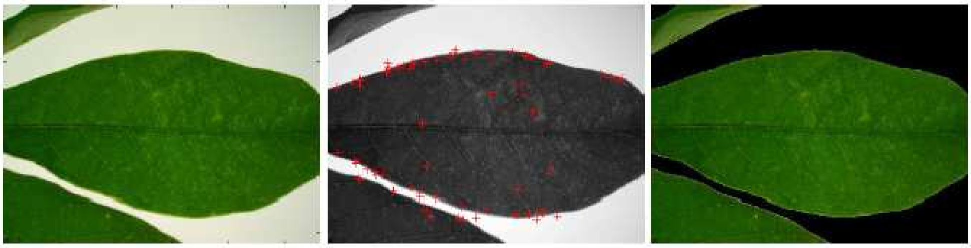


Figure 10.6: Examples of healthy leaves (top) and those infected with cassava mosaic disease (bottom).

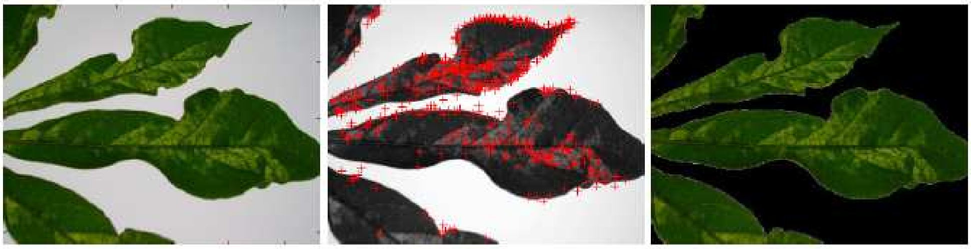
Three image processing techniques were employed to obtain representative feature data from the leaf images of the healthy plants and from those with Cassava Mosaic Disease (CMD).

In the first technique, we obtained a normalised histogram of the hues of pixels, taken by converting the image to HSV colour space. In the second we used SURF (Speeded Up Robust Features) (Bay et al. 2008); a scale and rotation invariant interest point detector and descriptor to obtain representative features. In the third we used SIFT (Scale Invariant Feature Transformation) (Lowe 2004) to obtain shape features corresponding to a 4×4 grid of histograms around each keypoint location. All these three methods are differently motivated and part of our investigation was to understand how classification performance changes with the use of different features.

The hue distribution was calculated for each image using 50 histogram bins, and was then normalised. The SURF and SIFT schemes identify points of interest on each image of a leaf and output a range of descriptors per image. For these two datasets we averaged out the descriptors to obtain a representative prototype for each image. Intuitively, such an averaged feature descriptor gives an overall description of the shape characteristics in the image. Figure 10.7 shows the locations of SURF interest points in two training images. Example hue histograms are also illustrated in Figure 10.8. The feature sets used included the three sets of features:



(a) Healthy leaf (b) SURF keypoints (c) Leaf only



(d) Leaf with CMD (e) SURF keypoints (f) Leaf only

Figure 10.7: Examples of cassava leaf images, the locations of extracted SURF keypoints, and the results of background filtering (top row: healthy leaf, bottom row: CMD)

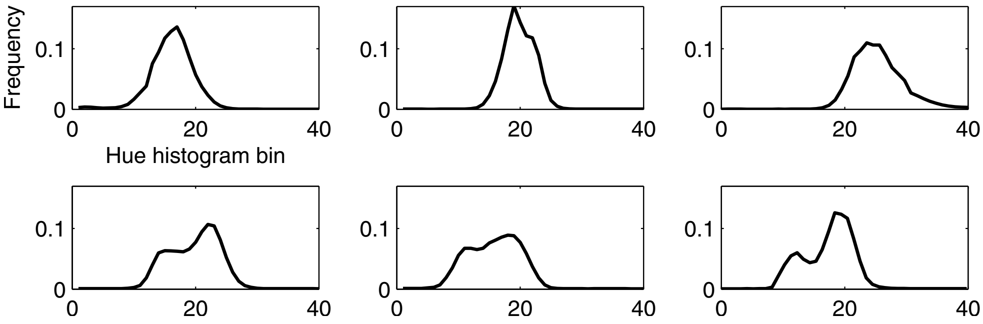


Figure 10.8: Normalised hue histograms of the leaf images (calculated from the corresponding images in Fig. 10.6), with healthy plants on the top row, and those with CMD on the bottom row. Note that the CMD leaves tend to have a bimodal hue distribution, where parts of the leaf affected by chlorosis add to the yellow range of the spectrum.

(i) color/hue histograms (HSV space), (ii) SURF features and (iii) SIFT features.

Classifier	HSV	SURF	SIFT
NB	0.7455 ± 0.0791	0.9111 ± 0.0474	0.9455 ± 0.0474
NN	0.8545 ± 0.0822	0.9000 ± 0.0707	0.9727 ± 0.0474
SVC	0.8727 ± 0.0725	0.8889 ± 0.0707	0.9273 ± 0.0643
k-NN	0.9455 ± 0.0474	0.9889 ± 0.0474	**0.9909 ± 0.0433**
DLVQ	0.8789 ± 0.0539	N/A	**0.9786 ± 0.0985**

Table 10.1: Classification accuracy area under the receiver operating characteristic curve (AUC) performance of different classifiers for the three base datasets, HSV, SURF and SIFT.

10.4.3 Classification

The data being conveniently represented as histograms, we used Divergence-based LVQ (DLVQ) for the classification. As aforementioned in Chapter 3, DLVQ exploits the fact that our features are represented in the form of normalised distributions. We used the Cauchy-Schwarz divergence as the distance measure in DLVQ algorithm.

For comparative purposes, we also applied several standard classifiers. We applied naïve Bayes (NB), a two layer multi-layer perceptron neural network, (NN) *Parameters: number of hidden neurons = 10, number of training epochs = 100, regularization* = 10^{-14}., a *2-norm* support vector classifier, (SVC) *Parameters:* C *= 0, degree = 1,* γ *= 0, regularization* = 10^{-14}, and a k-nearest neighbour classifier, (*k*-NN); *we used* $k = 10$ *random splits that did not result in ties in the voting.*

10.4.4 Empirical results

Classification results for the three datasets: HSV histogram data, SURF feature data and SIFT feature data are shown in Table 10.1. Results for standard algorithms were obtained as 100-fold cross validated scores while for DLVQ results were obtained as an average over 100 randomized splits of the data after 1000 epochs.

For the normalized HSV colour histogram data and the normalized SIFT data, we observe the DLVQ classifier providing a comparable accuracy to *k*-NN. Analysis of colour histograms by use of divergence measures has the potential to give good classification performance because histograms are more naturally represented as distributions. However high accuracy is also observed for the standard implementation of the other standard classifiers especially *k*-NN. As mentioned in Chapter 2, however, an algorithm like *k*-NN is much more expensive to run than DLVQ for example. *k*-NN calculates distances of each point from every other point while for DLVQ the distances to the prototypes is calculated in the classification step. This has critical implications for deployment of a solution such as this. We expound on the deployment in the Section 10.4.5.

For the SURF dataset, DLVQ is not applicable since SURF introduces negative non-normalised data. In Chapter 4 we propose a solution for this. We present an LVQ system that uses a global distance measure that combines different heterogeneous distance measures relative to the different representations of the data. This enables the LVQ system to make use of this additional information and thus improve classification performance.

10.4.5 Deployment on mobile devices

In many developing countries, the relatively highly specialized task of visually diagnosing plants in remote fields is made especially problematic because of the limited skilled personnel. Reliable automatic methods for diagnosis therefore offer the possibility of extending the scope of disease surveillance. The ubiquity of camera phones in even the most rural parts of many developing countries introduces the possibility of surveillance by non-expert workers. By implementing some intelligence on these mobile phones, immediate diagnosis of disease plants can be attained or submission of crop images can be done to some central server where automated classification is done. We investigated computer vision techniques for using camera-enabled mobile devices to make disease diagnoses directly, allowing survey workers with lower levels of training to make the diagnoses, and hence reducing survey costs.

Specifically, from the training process described in the previous section (and in Chapter 3), we were able to obtain trained prototypes. The advantage of using DLVQ or any LVQ scheme is that once prototypes have been trained, then classification of new images is reduced to a feature extraction followed by a comparison step where the distance between the representation of the new image (in either color histograms or sift features) and the different prototypes is calculated. The diagnosis accorded to the new image is the class of the closest prototype.

For implementation purposes, we used a $ 100 Huawei mobile phone running Android 2.2. The way the system works is once the application is activated on the phone, the user points the phone camera at a plant and within a minute, features have been extracted from the image captured by the phone camera and a classification has been output on the phone screen. Figure 10.9 gives an illustration of this.

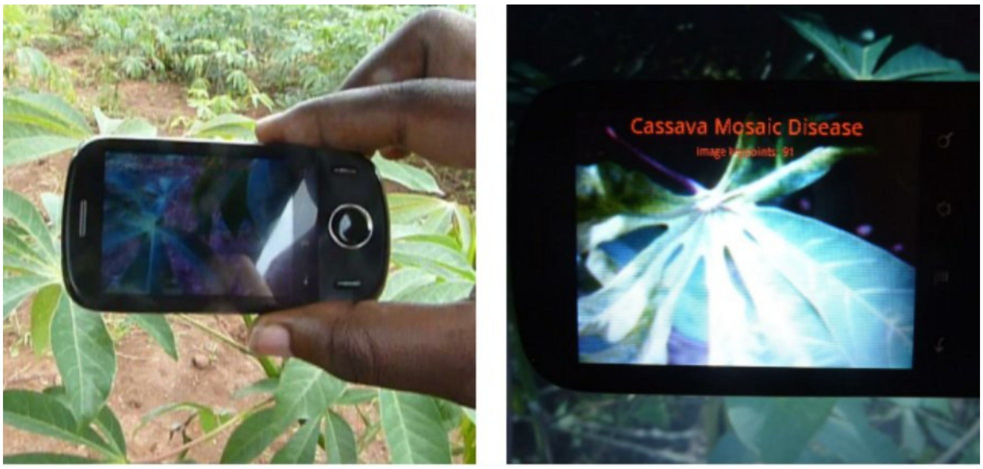


Figure 10.9: Deployment and operation of automated diagnosis on a mobile device.

10.4.6 Automated whitefly count

Whiteflies are the major vector for transmission of cassava mosaic disease and other cassava related diseases. To understand the spread of disease, experts are required to count the number of whifeflies on a collection of leaves in each garden during surveillance. These whiteflies, because they are so tiny can be up to the order of 300 on a single plant. Figure 10.10 attempts to depict why this is not a trivial task; the small size and mobility of the whiteflies results in inaccurate whitefly counts. Whitefly counting has the added caveat that it has to be done in the early morning hours because sunlight perturbs the whiteflies making them even harder to count.

We built an Android application that uses the OpenCV implementation of Haar Cascades to segment out and count the whiteflies from an image taken of the leaf. With our system we get near 85 % accuracy. We found that this functionality was perceived as being particularly valuable by the surveyors we worked with. An added advantage is that since whiteflies transmit several other diseases that affect other plants, this same system can be a useful intervention that can be extended to other crops.

10.5 Practical considerations for system deployment

This section highlights some of the unforeseen challenges with actually deploying such a system. As is always the case the theory underestimates the praxis – this case was no different, several issues we thought were significant and would be prob-

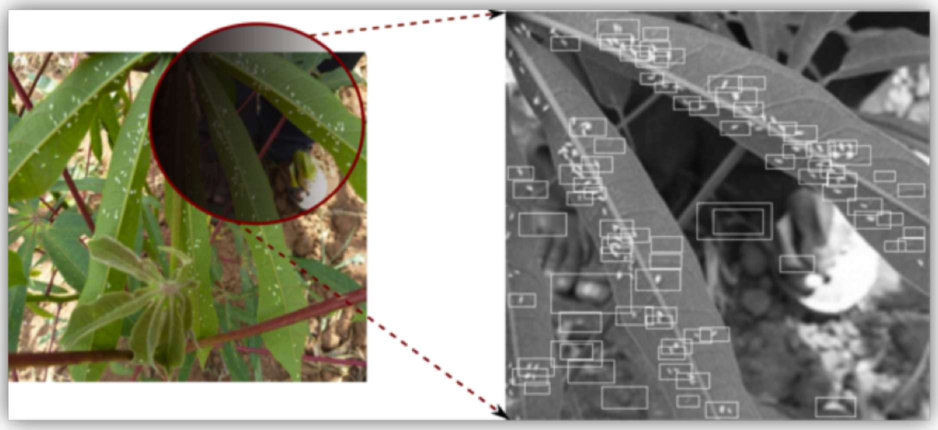


Figure 10.10: A Cassava leaf with an infestation of whiteflies. Manually counting these whiteflies is time consuming and error-prone.

lematic actually turned out to be non-issues; the more difficult instances were the non-issues that became issues. We detail several of these here spanning from the preparation, training and the deployment phases of the apps on the phone and what was observed on two separate field test visits with the actual survey experts. For this test deployment we used 8 phones running Android and about 16 surveyors, 2 per phone.

1. Space/storage considerations on the phone – the image capture application uses the native phone camera application. We found that the default setting for the size and resolution of the images taken with the camera had to be changed from a 3M high-resolution image to a 1 M normal resolution image. The size of the images affects how much bandwidth will be used in uploading the images to the server and the capacity of the memory card to use in the phone. For the real survey we expect each phone to take a minimum of 1000 images.

2. Screen/keyboard size of phone – during the training of the surveyors, the issue of usage of the phone keyboard came up. We were using an android phone; the Huawei Gaga U8180 which has a display size of 240 X 320 px. Issues of how to use the touch screen with dirty fingers (from digging up roots, etc.) also came up.

3. Sun glare – this happened to be one of the most daunting issues. Collecting data in the gardens when the sun is all out was problematic because of diffi-

culty in seeing the phone screen and hence difficulty in navigating the form. One solution round this was to set the brightness setting to auto on the phone. Some improvement is realized but this is something to take into consideration when buying devices.

4. Background clutter – for one of the field visits we tried to use a cardboard paper that was put behind each leaf before the photo of the leaf was taken. This was a very cumbersome procedure especially when trying to take an image of the whiteflies on the leaves, more so because for the whitefly shot, one needs to turn the leaf over as the whiteflies reside on the underside of the leaf. Taking the shot with the background required at least two people to execute. This was rejected and so images had to be taken with a cluttered background and the filtering done at the server end.

5. Image capture – To score the severity of disease manifested by a plant accurately, some diseases require that the examination of the whole plant is done, while for others this can be done from looking at the canopy of the plant. A 2D image of a plant unfortunately does not provide all the information for accurately determining the severity score of a plant, we thus had to settle with taking two representative images of the plant – one full plant image and a close-up canopy/leaf shot.

6. Power/charging issues – the Huawei Gaga like most touch screen phones has about 1 day lifespan on the battery under heavy usage. We had to provide car chargers to the surveyors so the phones could be charged in the cars as they move from one garden to the next.

10.6 Potential impact and lessons learnt

With the pilot we achieved enough positives to warranty that our survey system be used in the forth coming national survey along side the ordinary paper-based system. Because not all components of the system have been tested fully, only a small portion of the system that does the actual collection of data using mobile phones will be deployed. Data will be automatically mapped in real time and some processing done to provide a prediction of diagnosis that can be compared with the experts diagnosis from the field.

One point to note: the success of the whitefly count application – seemingly trivial problem from the machine learning point of view – seems to have accrued more enthusiasm than the more complicated auto diagnosis task. A good lesson in this case is that solving the less complicated problems first and getting quick results initially can be extremely critical in getting a buy-in from domain collaborators when thinking about deployment of a system.

Conclusion

We have discussed the problems of crop surveillance in detail and given a machine learning solution to the problems. When dealing with real-world problems, it appears imperative to carry the whole bag of tools to tackle the problem. We demonstrated how several machine learning tools are integrated to solve the set of problems. Importantly we showed how LVQ can be used in a real-world problem and offer superior performance to other possible tools. This chapter has generally not dwelt on the actual implementation of the different interventions. This was intentional – alot of the scientific theory behind these interventions has been discussed in previous chapters besides. The goal of this chapter was to discuss a real-world problem and show how the different components fit together to solve the problem. It also put some emphasis on what practical considerations to take when thinking of deploying such a system. We believe this may be more beneficial to someone deploying a similar system.

Chapter 11

Summary

This thesis is a two-part thesis. In the first part we discuss extensions of LVQ prototype-based classifiers that use information theoretic measures as distance measures. We also present work on the use of different representations of data, in this case histograms of images, SIFT and SURF features in the same LVQ system. We show the possibility of formulating one combined distance measure for the heterogeneous dataset formed by a combination of their individual representative distance measures.

In the second part we delve into causal structure discovery and its application to real world problems. We also presented a first attempt at leveraging some of the techniques of causal learning and applying them to feature relevance learning in LVQ. We also present some deployment examples of some of the techniques. A detailed summary of some of the overarching ideas in the thesis follows.

We begin by motivating the use of divergences as a possible similarity (dissimilarity) measure between prototypes and data points in Chapter 2 and 3. The use of divergences as distance measures is, in principle, possible for all data sets that contain non-negative feature values. It appears particularly suitable for the classification of histograms, spectra, or similar data structures for which divergences are the natural representation. We further present a novel algorithm *Divergence-LVQ*, formulated as a GLVQ cost function based scheme that uses divergences as a distance measure. As a specific example of this versatile framework we consider the family of γ-divergences which contains the so-called Cauchy-Schwarz divergence as a special case and approaches the well-known Kullback-Leibler divergence in the limit $\gamma \to 0$. The corresponding training schemes are applied to three different real world data sets and we show that DLVQ can yield superior classification accuracies and Receiver Operating Characteristics. A key feature of this scheme is that we can swap out the specific divergence measure with any one of a large number of differentiable measures seamlessly.

Real world problems in some cases present us with data that can be represented in several ways. For example image data can be represented as RGB histograms or as numerical data from SIFT feature extraction. In Chapter 4, we discuss a variant of DLVQ, *Combined-distances LVQ*, that combines different representations of the data and trains the combined distance measure with a matrix that represents the component-wise correlations of the different data representations. The formulation of *Combined-distances LVQ* is similar to the GMLVQ formulation but with the dimensions representing the different components of the same (heterogeneous) data. The matrix lambda (Λ) is re-defined as a matrix that defines the correlations of the component-wise distance measures. We discuss two ways of combining the sub-distances; linearly and matrix-wise and show the superiority of the matrix-based combination over the linear combination of the sub-distances. While performance over other implementations using a single uniform distance measure for example, was not our concern here, from our experiments, we still observe superior performance over previous implementations with the same dataset in Chapter 3.

In Chapters 5 and 6, we change our focus to the field of causal structure learning and inference and motivate the marriage of the field of causality and LVQ learning. Chapter 5 presents a treatise of the field of causal structure learning from purely observational data and explains the foundational concepts that underlie causal structure discovery. The causal faithfulness and sufficiency assumptions are discussed as well. In causal discovery we try to calculate causes or causal relationships from observational data. Causes are important because they remain predictive even when the data is manipulated. We extend this concept to the field of LVQ in Chapter 6. We present a novel algorithm called *Causal-RLVQ* that extends Relevance LVQ (RLVQ); a classification scheme that learns a relevance profile for the classification task. *Causal-RLVQ* is made robust to outside interventions by identifying causes in the training data, since causes of a target variable remain predictive even when the data is manipulated. The causality - LVQ union of the algorithm is specified by a parameter alpha (α) that allows us to adjust the bias towards causative features, where $\alpha = 0$ yields standard RLVQ. The goal is to assure robust classification when it is not certain that the test set has the same distribution as the training set which is a very common scenario in praxis.

In several techniques in the field of machine learning, improvement in performance of several algorithms has resulted from combining several weak algorithms or tests into a committee for example boosting and bagging are techniques that benefit from this. In Chapters 7 and 8 we looked at committees of causal structure learners and committees of conditional independence tests. In both cases we find that the committee performs better than any individual algorithm/test. In Chapter 7 we develop and use a committee of *weak* structure learners to do causal structure discovery from observational data. We also present a novel causal discovery algorithm *Expected Partial Correlation (EPC)* that uses partial correlation in the discovery of causal relationships between variables. The ensemble/committee method for causal discovery uses a committee of differently oriented causal algorithms to vote on the causal relationship between variables. The chapter is concluded with some experiments and results of a competition where this method gave superior performance.

In Chapter 8 we investigate the efficacy and improvement obtained from combining a battery of (conditional) independence tests for understanding the associative relationships between variables. We look at two approaches of combining the tests. In the first approach, coined hard-decision approach, the tests are combined with weights that are a function of features of the data to be tested. The resulting adaptive committee independence test, *Dependable Dependence 1 (DD1)*, provides superior performance to any individual test. We also demonstrate that this approach improves the performance of constraint-based causal structure learning algorithms as compared to using a fixed dependence test for all sets of variables in the dataset. In the second approach coined soft-decision approach, each test returns a test statistic that can be related to the amount of noise in the data. The committee test; *Dependable Dependence 2 (DD2)* formed as a result of combining the different tests, returns a score related to the coefficient of determination (R^2) of the data. Empirically *DD2* out performs *DD1* albeit marginally. Both tests out perform any of the individual tests in the committee however.

Chapters 9 and 10 detail some practical applications of these techniques to address real world problems; food insecurity prediction using causal structure discovery techniques and crop disease diagnosis and surveillance using techniques in LVQ discussed in Chapters 3 and 4. In Chapter 9, we calculate the causal relationships between socio-economic factors in a developing-world household and the risk of experiencing famine (food security) in that household. We show that it is possible to predict famine in a household based on these factors, looking at data collected from 5404 households in Uganda. Chapter 10 discusses the development and deployment of a crop disease surveillance system. We describe how we implemented a mobile-based data collection surveillance system that enables real-time mapping of incidence counts as well as displaying a country wide severity map. Further more we demonstrate that the diagnosis of plant disease (using LVQ techniques) and vector counting can be automated using images taken by a camera phone.

11.1 Future work

The scope of this work has been fairly wide and as such several options for future research are available. A conservative list of possible future work is presented.

- For most of the algorithms developed or extended as part of this work, we have attempted to validate them using real world data. However to concretize understanding of these algorithms it is important to apply them to bigger and more datasets both real and semantic. Particularly the DLVQ and the combined-distances LVQ classifiers could benefit from increased training data. From a practical point of view, we are trying to collect more leaf image data from the National Crop Resources Research Institute (NaCRRI) in Uganda as well as carrying out further testing of the phone diagnosis application in the field. Increased data and further tests will inform more research in the algorithms presented in this work.

- In Chapter 6, the Causal-RLVQ algorithm gave a proof of possibility on simulated data. Our formulation of the CRLVQ algorithm however, still has some limitations, particularly in the assessment of independence amongst features in the data. Future attempts will focus on employing improved versions of RLVQ with a better formulation of the assessment of independence for example using a committee. An interesting direction to address this could be to extend our current work to matrix relevance LVQ where a full matrix is adapted during the training to reflect the relevance of correlated features of the data drawing from the work done in (Schneider et al. 2009b).
- Chapters 7 and 8 dealt with the use of committees of weak algorithms and tests. Results show that the committees out perform any individual algorithm or test. Future work in this regard can be envisaged in formulating appropriate weighting mechanisms for the committee. A study of the different combination strategies of the committee members would be useful in quantifying the improvement in performance of a committee over the individual committee members. This is important in calculating the performance vs resource usage equation that one has to consider when dealing with committees.

Bibliography

Abramson, B., Brown, J., Edwards, W., Murphy, A. and Winkler, R. L.: 1996, Hailfinder: A bayesian system for forecasting severe weather, *International Journal of Forecasting* **12(1)**, 57–71.

Aliferis, C. F., Tsamardinos, I. and Statnikov, A.: 2003, HITON, A Novel Markov Blanket Algorithm for Optimal Variable Selection, *Proc. of the 2003 American Medical Informatics Association (AMIA) Annual Symposium*, pp. 21–25.

Amari, S. I. and Nagaoka, H.: 2000, Methods of information geometry, *Translations of Methematical Monographs*, Vol. 191, Oxford University Press, New York.

Anokwa, Y., Hartung, C., Brunette, W., Borriello, G. and Lerer, A.: 2009, Open source data collection in the developing world, *Computer* **42**(10), 97–99.
URL: *http://dx.doi.org/10.1109/MC.2009.328*

Anonymous: 2011, Second helpings of tapioca pudding: a crucial crop in new trouble, The Economist. Jan 27th.

Arah, O.: 2008, The role of causal reasoning in understanding simpsons paradox, lords paradox, and the suppression effect: Covariate selection in the analysis of observational studies, *Emerging Themes in Epidemiology* .

Asuncion, A., Newman, D. J., Hettich, S., Blake, C. L. and Merz, C. J.: 1998, UCI repository of machine learning databases, http://archive.ics.uci.edu/ml/.

Baba, K., Shibata, R. and Sibuya, M.: 2004, Partial correlation and conditional correlation as measures of conditional independence, *Australian and New Zealand Journal of Statistics* **46**(4), 657–664.

Banerjee, A., Merugu, S., Dhillon, I. and Ghosh, J.: 2005, Clustering with Bregman divergences, *Journal of Machine Learning Research* **6**, 1705–1749.

Bay, H., Ess, A., Tuytelaars, T. and Gool, L.: 2008, SURF: Speeded Up Robust Features, *Computer Vision and Image Understanding* **110(3)**, 346–359.

Beinlich, I., Suermondt, H., Chavez, R. and Cooper, G.: 1989, The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks, *Proceedings of the 2nd European Conference on Artificial Intelligence in Medicine*, Springer-Verlag, pp. 247–256.

Bennett, K. and Mangasarian, O.: 1992, Robust linear programming discrimination of two linearly inseparable sets, *Optimization Methods and Software* **1**, 23–34.

Biehl, M., Ghosh, A. and Hammer, B.: 2007, Dynamics and generalization ability of LVQ algorithms, *Journal of Machine Learning Research* **8**, 323–360.

Binder, J., Koller, D., Russell, S. and Kanazawa, K.: 1997, Adaptive probabilistic networks with hidden variables, *Machine Learning* **29(2-3)**, 213–244.

Bishop, C. M.: 2006, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Boelke, J., Gerhard, M., Schleif, F.-M., Decker, J., Kuhn, M., Elssner, T., Pusch, W. and Kostrzewa, M.: 2005, *ClinProTools 2.0 User Documentation*. Available in the ClinProt - ClinProTools 2.0 Software package.

Bojer, T., Hammer, B., Schunk, D. and von Toschanowitz, K. T.: 2001, Relevance determination in learning vector quantization, *in* M. Verleysen (ed.), *Proc. of the 9th European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, pp. 271–276.

Borboudakis, G., Triantafillou, S., Lagani, V. and Tsamardinos, I.: 2011, Constraint-based approach to incorporate prior knowledge in causal models, *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*.

Bunte, K., Hammer, B., Villmann, T., Biehl, M. and Wismüller, A.: 2010, Exploratory observation machine (XOM) with Kullback-Leibler divergence for dimensionality reduction and visualization, *in* M. Verleysen (ed.), *European Symposium on Artificial Neural Networks (ESANN 2010)*, d-side publishing, pp. 87–92.

Candela, J. Q., Sugiyama, M., Schwaighofer, A. and Lawrence, N.: 2009, Dataset shift in machine learning.

Cheng, J., Greiner, R., Kelly, J., Bell, D. and Liu, W.: 2002, Learning bayesian networks from data : An information-theory based approach, *Artificial Intelligence* **137(1-2)**, 43–90.

Chow, C. K. and Liu, C. N.: 1968, Approximating discrete probability distribution with dependence trees, *IEEE Transactions on Information Theory* **14(3)**, 462–467.

Cichocki, A., Zdunek, R., Phan, A. H. and Amari, S.-I.: 2009, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*, Hoboken, NJ: Wiley.

Cooper, G. F. and Herskovits, E. H.: 1992, The induction of probabilistic networks from data, *Machine Learning* **9(4)**, 309–347.

Cover, T. M. and Hart, P. E.: 1967, Nearest neighbor pattern classification, *IEEE Transactions on Information Theory* **13**(1), 21–27.

Duda, R. O., Hart, P. E. and Stork, D. G.: 2000, *Pattern Classification*, Wiley-Interscience Publication.

Fawcett, T.: 2006, An introduction to ROC analysis, *Patt. Rec. Lett.* **27**, 861–874.

Fujisawa, H. and Eguchi, S.: 2008, Robust parameter estimation with a small bias against heavy contamination, *Multivariate Analalysis* **99**(9), 2053–2081.

Fukumizu, K., Gretton, A., Sun, X. and Schölkopf, B.: 2008, Kernel measures of conditional dependence, *NIPS*, Vol. 20, Cambridge, MA, pp. 489–496.

Gretton, A., Fukumizu, K., Teo, C. H., Song, L., Schölkopf, B. and J.Smola, A.: 2008, A kernel statistical test of independence, *NIPS*, Vol. 20, Cambridge, MA, pp. 585–592.

Gretton, A. and Gyorfi, L.: 2010, Consistent nonparametric tests of independence, *Journal of Machine Learning Research* **11**, 1391–1423.

Grossberg, S.: 1976, Adaptive pattern classification and universal recoding: Part i: Parallel development and coding of neural feature detectors, *Biological Cybernetics.* **23**, 121–134.

Guyon, I.: 2009, Lung cancer simple model.
URL: *http://www.causality.inf.ethz.ch//data/LUCAS.html*

Guyon, I., Aliferis, C., Cooper, G., Elisseeff, A., Pellet, J.-P., Spirtes, P. and Statnikov, A.: 2008, Design and analysis of the causation and prediction challenge, *JMLR Workshop on Causality* **3**, 1–33.

Guyon, I., Aliferis, C., Cooper, G., Elisseeff, A., Pellet, J.-P., Spirtes, P. and Statnikov, A.: 2009, Datasets of the causation and prediction challenge, *Technical Report* .

Guyon, I., Aliferis, C. and Elisseeff, A.: 2007, Causal feature selection, *in* H.Liu and H.Motoda (eds), *Computational Methods of Feature Selection*, Data Mining and Knowledge Discovery, Chapman and Hall/CRC Press, Boca Raton, FL.

Haase, S.: 2014, *The Application of Divergences in Prototype Based Vector Quantization*, PhD thesis, University of Groningen, Faculty of Mathematics and Natural Sciences.

Hammer, B., Strickert, M. and Villmann, T.: 2005a, On the generalization ability of GRLVQ networks, *Neural Processing Letters* **21**(2), 109–120.

Hammer, B., Strickert, M. and Villmann, T.: 2005b, Supervised neural gas with general similarity measure, *Neural Processing Letters* **21**(1), 21–44.

Hammer, B. and Villmann, T.: 2002, Generalized relevance learning vector quantization, *Neural Networks* **15**(8-9), 1059–1068.

Heckerman, D.: 1998, Bayesian approach to learning causal networks, *Proc. Uncertainty in Artificial Intelligence (UAI)*.

Hoyer, P., Shimizu, S. and Kerminen, A.: 2006, Estimation of linear, non-gaussian causal models in the presence of confounding latent variable, *Proceedings of the Third European Workshop on Probabilistic Graphical Models (PGM'06)*, pp. 155–162.

IFPRI: 1998, Can FAO's Measure of Chronic Undernourishment be Strengthened, *Technical report*, International Food Policy Research Institute.

Jang, E., Fyfe, C. and Ko, H.: 2008, Bregman divergences and the self organising map, *in* C. Fyfe, D. Kim, S.-Y. Lee and H.Yin (eds), *Intelligent Data Engineering and Automated Learning IDEAL 2008*, Springer Lecture Notes in Computer Science 5323, pp. 452–458.

Jenssen, R., Principe, J. C., Erdogmus, D. and Eltoft, T.: 2006, The Cauchy-Schwarz divergence and Parzen windowing: Connections to graph theory and Mercer kernels, *Journal of the Franklin Institute* **343**(6), 614–629.

Kietzmann, T. C., Lange, S. and Riedmiller, M.: 2008, Incremental GRLVQ: Learning relevant features for 3D object recognition, *Neurocomputing* **71**(13-15), 2868–2879.

Kohonen, T.: 1986, Learning vector quantization for pattern recognition, *Technical Report TKK-F-A601*, Helsinki University of Technology, Espoo, Finland.

Kohonen, T.: 1990, Improved versions of learning vector quantization, *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, pp. 545–550 vol.1.

Kohonen, T., Schroeder, M. R. and Huang, T. S. (eds): 2001, *Self-Organizing Maps*, 3rd edn, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Kullback, S. and Leibler, R. A.: 1951, On information and sufficiency, *Annals of Mathematical Statistics* **22**, 49–86.

Langley, P.: 2011, The changing science of machine learning, *Machine Learning* **82**, 275–279.

Lauritzen, S. and Spiegelhalter, D.: 1998, Local computations with probabilities on graphical structures and their application to expert systems., *Journal of the Royal Statistical Society B* **50(2)**, 157–224.

Lowe, D.: 2004, Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision* **60(2)**, 91–110.

Margaritis, D.: 2005, Distribution-free learning of bayesian network structure in continuous domains., *Proc. of AAAI*, Pittsburgh, PA, pp. 825–830.

Mendenhall, M. J. and Merényi, E.: 2008, Relevance-based feature extraction for hyperspectral images, Vol. 19, pp. 658–672.

Mitchell, T. M.: 1997, *Machine Learning*, McGraw-Hill Series in Computer Science, WCB/McGraw-Hill, Boston, MA.

Mooij, J. M., Stegle, O., Janzing, D., Zhang, K. and Scholköpf, B.: 2010, A probabilistic latent variable models for distinguishing between cause and effect, *Proc. Neural Information Processing Systems (NIPS)*.

Munteanu, P. and Bendou, M.: 2002, The EQ framework for learning equivalence classes of Bayesian networks, *First IEEE International Conference on Data Mining (IEEE ICDM)*, San Jose.

Nielsen, F. and Nock, R.: 2009, Sided and symmetrized Bregman centroids, *IEEE Transactions on Information Theory* **55**, 2882–2904.

NNRC: 2002, Bibliography on the self-organizing map (SOM) and learning vector quantization (LVQ), Helsinki, University of Technology, available at: http://liinwww.ira.uka.de/bibliography/Neural/SOM.LVQ.html.

Okori, W., Obua, J. and Baryamureeba, V.: 2009, Famine Disaster Causes and Management Based on Local Community's perception in Northern Uganda, *Research Journal of Social Sciences* **4**, 21–32.

Otim-Nape, G., Alicai, T. and Thresh, J.: 2005, Changes in the incidence and severity of cassava mosaic virus disease, varietal diversity and cassava production in Uganda, *Annals of Applied Biology* **138(3)**, 313–327.

Pearl, J. (ed.): 2000, *Causality: Models, Reasoning, and Inference*, 1st edn, Cambridge University Press., Cambridge, MA.

Pearl, J. (ed.): 2009, *Causality: Models, Reasoning, and Inference*, 2rd edn, Cambridge University Press., Cambridge, MA.

Pellet, J.-P. and Elisseeff, A.: 2007, A partial correlation-based algorithm for causal structure discovery with continuous variables, *7th International Symposium on Intelligent Data Analysis*.

Principe, J. C., Xu, D. and Fisher III, J. W.: 2000, Information-theoretic learning, *in* S. Haykin (ed.), *Unsupervised Adaptive Filtering*, second edn, Vol. 1, Wiley, New York, chapter 7.

Rényi, A.: 1970, *Probability Theory*, North-Holland series in applied mathematics and mechanics, v. 10, Amsterdam.

Reshef, D. N., Reshef, Y. A., Finucane, H. K., Grossman, S. R., McVean, G., Turnbaugh, P. J., Lander, E. S., Mitzenmacher, M. and Sabeti, P. C.: 2011, Detecting novel associations in large data sets, *Science* **334**(6062), 1518–1524.
URL: *http://www.sciencemag.org/content/334/6062/1518.abstract*

Robinson, R. W.: 1977, Counting unlabeled acyclic digraphs, *in* C. Little (ed.), *Combinatorial Mathematics V*, Vol. 622 of *Lecture Notes in Mathematics*, Springer, Berlin.

Rumelhart, D. and Zipser, D.: 1985, Feature discovery by competitive learning, *Cognitive Science* **9**, 75–112.

Salih, S.: 1994, Food security in east and southern africa, *Nordic Journal of African Studies* **13**(1), 3–27.

Sato, A. S. and Yamada, K.: 1996, Generalized learning vector quantization, *in* M. C. M. D. S. Touretzky and M. E. Hasselmo (eds), *Advances in Neural Information Processing Systems (NIPS)*, Vol. 8, MIT Press, Cambridge, MA, USA, pp. 423–429.

Schleif, F.-M., Villmann, T. and Hammer, B.: 2008, Classification in clinical proteomics, *International Journal of Approximate Reasoning* **47**, 4–16.

Schneider, P., Biehl, M. and Hammer, B.: 2009a, Adaptive relevance matrices in learning vector quantization, *Neural Computation* **21**(12), 3532–3561.

Schneider, P., Biehl, M. and Hammer, B.: 2009b, Distance learning in discriminative vector quantization, *Neural Computation* **21**(10), 2942–2969.

Schneider, P., Biehl, M. and Hammer, B.: 2010, Hyperparameter learning in probabilistic prototype-based models, *Neurocomputing* **73**(7–9), 1117–1124.

Seo, S. and Obermayer, K.: 2003, Soft learning vector quantization, *Neural Computation* **15**(7), 1589–1604.

Shalev-Shwartz, S., Singer, Y. and Ng, A. Y.: 2004, Online and batch learning of pseudo-metrics, *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada.

Shimizu, S., Hoyer, P. O., Hyvarinen, A. and Kerminen, A.: 2006, A Linear Non-Gaussian Acyclic Model for Causal Discovery, *Machine Learning Research* **7**, 2003–2030.

Shoham, Y. (ed.): 1998, *Reasoning About Change: Time and Causation from the Standpoint of Artificial Intelligence*, MIT Press, Cambridge, MA.

Spirtes, P., Glymour, C. and Scheines, R.: 1993, *Causation, Prediction and Search*, Vol. 81, Springer Verlag, Berlin.

Spirtes, P., Glymour, C. and Scheines, R.: 2000, *Causation, Prediction and Search*, Cambridge University Press, Cambridge.

Strickert, M., Seiffert, U., Streenivasulu, N., Weschke, W., Villmann, T. and Hammer, B.: 2006, Generalized relevance LVQ (GRLVQ) with correlation measures for gene expression analysis, *Neurocomputing* **69**(7–9), 651–659.

Sugiyama, M.: 2011, Least-Squares Independence Test, *IEICE Transactions on Information and Systems* **6**, 1333–13336.

Sun, X., Janzing, D., Scholköpf, B. and Fukumizu, K.: 2007, A kernel-based causal learning algorithm, *Proceedings of the 24th International Conference on Machine Learning*.

Suppes, P. (ed.): 1998, *Probabilistic causality in space and time*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Tillman, R., Gretton, A. and Spirtes, P.: 2009, Nonlinear directed acyclic structure learning with weakly additive noise models, *Advances in NIPS*, Vol. 22, Vancouver, Canada.

Torrkola, K.: 2003, Feature extraction by non-parametric mutual infromation maximization, *Journal of Machine Learning Research* **3**, 1415–1438.

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

Villmann, T. and Haase, S.: 2009, Mathematical aspects of divergence based vector quantization using frechet-derivatives, *Technical Report MLR-03-2009*, Univ. Leipzig/Germany. ISSN:1865-3960 http://www.uni-leipzig.de/~compint/.

Villmann, T., Haase, S., Schleif, F.-M., Hammer, B. and Biehl, M.: 2010, The mathematics of divergence based online learning in Vector Quantization, *Proc. Fourth International Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR 2010)*, Vol. 5998 of *Springer Lecture Notes in Artificial Intelligence LNAI*, Springer, pp. 108–119.

Villmann, T., Hammer, B., Schleif, F.-M., Herrmann, W. and Cottrell, M.: 2008, Fuzzy classification using information theoretic learning vector quantization, *Neurocomputing* **71**, 3070–3076.

Wagstaff, K.: 2012, Machine learning that matters, *29th Proceedings of International Conference on Machine Learning*, Edinburgh, Scotland.

Wasserman, L.: 2003, *All of Statistics: A Concise Course in Statistical Inference*, Springer Verlag.

Webb, P. and Yohannes, Y.: 1999, Famine in Ethiopia : Policy Implications of Coping Failure at National and Household levels, *Technical Report 92*, International Food Policy Research Institute.

Weinberger, K. Q., Blitzer, J. and Saul, L. K.: 2006, Distance metric learning for large margin nearest neighbor classification, *Advances in Neural Information Processing Systems (NIPS)* **18**, 1473–1480.

Workbench, C.: 2010, Causality workbench data repository.
URL: *http://www.causality.inf.ethz.ch/repository.php*

Xu, L., Hutter, F., Hoos, H. H. and Leyton-Brown, K.: 2008, SATzilla: portfolio-based algorithm selection for SAT, *Journal of Artificial Intelligence Research* **32**, 562–606.

Zhang, K., Peters, J., Janzing, D. and Schölkopf, B.: 2011, Kernel-based Conditional Independence Test and Application in Causal Discovery, *Proc. UAI*, Vol. 27, Corvallis, OR, USA, pp. 804–813.

Zühlke, D., michael Schleif, F., Geweniger, T., Haase, S. and Villmann, T.: 2010, Learning vector quantization for heterogeneous structured data, *in* M. Verleysen (ed.), *European Symposium on Artificial Neural Networks (ESANN 2010)*, d-side publishing, pp. 271–276.

Samenvatting

Dit proefschrift bestaat uit twee delen. In het eerste deel beschrijven we hoe de op prototypen gebaseerde classificator LVQ uitgebreid kan worden door gebruik te maken van maten uit de informatie theorie. Daarnaast vergelijken we verschillende manieren van datarepresentatie in deze LVQ configuratie, in dit geval histogrammen van fotos, SIFT- en SURF-kenmerken. We tonen hoe hiervoor een enkele gecombineerde afstandsmaat kan worden geformuleerd, door de afzonderlijke afstandsmaten samen te nemen. In het tweede deel onderzoeken we het vinden van causale verbanden en toepassingen op problemen die uit het leven zijn gegrepen. Daarnaast verkennen we de combinatie met relevantie leren in LVQ en tonen we enkele toepassingen. Nu volgt een gedetailleerde samenvatting van de overkoepelende themas van dit proefschrift.

Allereerst motiveren we het gebruik van divergentiematen als mogelijke (on) gelijkheidsmaten tussen prototypes en data in hoofdstukken 2 en 3. Dit is, in principe, mogelijk voor alle soorten data zolang de waarden niet negatief zijn. Divergentiematen zouden daardoor vooral toepasbaar zijn op histogrammen, spectra, of vergelijkbare datastructuren waarvoor divergentiematen een natuurlijke representatie zijn. We presenteren een nieuw algoritme, genaamd *Divergentie-LVQ*, dat geformuleerd is als een schema dat op kostfuncties gebaseerd is, vergelijkbaar met GLVQ, echter gebruikmakend van divergentiematen. Als specifiek voorbeeld van dit omvangrijke raamwerk beschrijven we de familie van γ-divergenties, waaronder de zogenoemde Cauchy-Schwarz divergentie en, in de limiet $\gamma \to 0$, de Kullback-Leibler divergentie. De bijbehorende trainingschemas worden toegepast op drie dataverzamelingen uit het ware leven, waarbij we tonen dat DLVQ grotere accuratesse en ROC-characteristieken bereikt. Een belangrijke eigenschap van dit schema is dat eenvoudig gewisseld kan worden van divergentiemaat, zodat gekozen kan worden uit een groot aantal maten, mits differentieerbaar.

Werkelijke data kan vaak op verschillende manieren worden gerepresenteerd. Fotos kunnen, bijvoorbeeld, worden weergegeven als RGB histogrammen of als numerieke data van SIFT-kenmerken. In hoofdstuk 4 bediscussiren we gevarieerde-afstanden-LVQ, een variant van DLVQ, dat verschillende representaties van data samen neemt en leert door gebruik te maken van de gecombineerde afstandsmaat waarin een matrix aangeeft hoe de verschillende data representaties elementsgewijs correleren. De specificatie van gevarieerde-afstanden-LVQ lijkt veel op GMLVQ, echter worden meerdere representaties tezamen gebruikt als data dimensies, en wordt de matrix Λ geherdefinieerd zodat het de elementsgewijze correlaties weer geeft. We bediscussiren twee manieren van het samennemen van de subafstanden tot een gecombineerde afstand: lineair en middels een matrix, en tonen aan dat de laatste variant beter presteert dan de lineare variant. Ook zijn de prestaties beter dan eerdere experimenten met uniforme afstandsmaten in hoofdstuk 3, al moet gezegd worden dat een directe vergelijking niet ons primaire doel was.

In hoofdstukken 5 en 6 beschouwen we het vinden van causale verbanden en motiveren we hoe deze technieken met LVQ kunnen worden gecombineerd. Hoofdstuk 5 beschrijft hoe causale verbanden kunnen worden gevonden in puur observationele data en beschrijft de fundamentele concepten die ten grondslag liggen aan het vinden van causale verbanden evenals de aannames van causale geloofwaardig heid en sufficintie. Het vinden van oorzaken is belangrijk omdat ze voorspellende waarde hebben, ook al is de data gemanipuleerd. We breiden dit veld uit naar LVQ in hoofdstuk 6, waar we een nieuw algoritme genaamd *Causale-LVQ* presenteren, dat gebaseerd is op Relevantie-LVQ waarin een relevantieprofielen geleerd worden. Door een parameter α te introduceren kan causale-RLVQ in meer of mindere mate waarde hechten aan causale kenmerken. Merk op dat in het geval $\alpha = 0$, de definitie samenvalt met standaard RLVQ.

Een veelgebruikte techniek om betere prestaties te verkrijgen uit relatief slechte classificatoren is door deze samen te nemen in een ensemble middels technieken zoals boosting en bagging. In hoofdstukken 7 en 8 bestuderen we ensembles van leermethoden van causale verbanden, als ook ensembles van tests van conditionele afhankelijkheden. In beide gevallen presteren de ensembles beter dan individuele tests. In hoofdstuk 7 ontwikkelen we een ensemble van structuur-leermethoden met als doel het vinden van causale verbanden uit observationele data. We presenteren ook een nieuwe algoritme, Verwachte Gedeeltelijke Correlatie, dat partile correlatie gebruikt voor het ontdekken van causale verbanden tussen variabelen. De ensemble-methode gebruikt verschillende causale algoritmen en stemming om tot een besluit te komen over causaliteit tussen variabelen. Dit hoofdstuk wordt afgesloten met experimenten en resultaten van een competitie waarin deze methode de beste resultaten leverde.

In hoofdstuk 8 bestuderen we de prestaties en efficintie van het combineren van vele (conditionele) onafhankelijkheidstesten voor het begrijpen van associatieve relaties tussen variabelen. We vergelijken twee benaderingen voor het combineren van individuele tests. De eerste, gewogen harde beslissing, benadering combineert de testresultaten middels gewichten die gedefinieerd zijn als een functie van de kenmerken van de data die getest wordt. De resulterende adaptieve ensemble onafhankelijkheidstest, Afhankelijke Afhankelijkheid 1 (AA1), levert betere prestaties dan willekeurig welke individuele test. We tonen ook aan dat deze benadering de prestaties verbetert van gelimiteerde algoritmen voor het vinden causale verbanden in vergelijking met het gebruiken van een enkele afhankelijkheidstest voor alle verzamelingen van variabelen in de dataverzameling. In de tweede benadering, gewogen zachte beslissing, levert iedere test een waarde op die gerelateerd is aan de hoeveelheid ruis in de data. De ensemble-test, Afhankelijke Afhankelijkheid 2 (AA2), levert een waarde op die gerelateerd is aan de correlatiecoefficient R^2 van de data. Empirische tests wijzen uit dat DD2 iets beter presteert dan DD1. Beide ensemble-tests presteren echter beter dan individuele tests.

Hoofdstukken 9 en 10 beschrijven toepassingen van deze technieken in de praktijk: het voorspellen van onzekerheid van voedsel middels het zoeken naar causale verbanden, en het diagnosticeren en monitoren van ziektes in gewassen middels de LVQ technieken beschreven in hoofdstukken 3 en 4. In hoofdstuk 9 bepalen we de causale verbanden tussen socio-economische factoren in een huishouden uit een ontwikkelingsland en het risico op voedselschaarste in het huishouden. We tonen dat het mogelijk is om voedselschaarste voorspeld kan worden uit deze factoren op basis van een studie waarin data verzameld is van 5404 huishoudens in Uganda. Hoofdstuk 10 beschrijft de ontwikkeling en het inzetten van een monitoringssysteem voor ziekten van gewassen. We beschrijven hoe we dit hebben gemplementeerd in een systeem dat gebruikt maakt van mobiele telefoons waarmee het mogelijk is om op ieder moment incidentie kan worden getoond als ook een landkaart waarop de ernst getoond wordt. Tot slot demonstreren we dat de diagnose van plantenziekten en het tellen van voorkomens kan worden geautomatiseerd door gebruik te maken van LVQ technieken die toegepast worden op fotos genomen met de camera van een mobiele telefoon.