Author:
Fearn, James A

Title:
Novel Methods for Approximate Bayesian Inference of Independent and Evolutionarily Dependent Data

Novel Methods for Approximate Bayesian Inference of Independent and Evolutionarily Dependent Data

By

JAMES FEARN

Department of Engineering Mathematics
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

NOVEMBER 2021

Word count: 19981

Abstract

In this thesis we present three new methods for probabilistic machine learning which extend widely used algorithms for approximate Bayesian inference and phylogenetic comparative methods. The first is a modification of Expectation Propagation (EP), called γ-EP, which incorporates a bias term into the approximate factors. The advantage of doing this becomes apparent when we adjust the coefficient γ of the bias term to maximise the evidence, as γ-EP is able to converge to solutions which make the data more probable.

The γ-EP method also provides an efficient algorithm for training sparse Bayesian linear classifiers. This makes it applicable to classification with repeated data points, which EP cannot handle robustly. It is simple to implement as it only requires a few modifications to the canonical EP algorithm. The γ-EP algorithm is extended to use kernel matrices and applied to oncogenic single nucleotide variant (SNV) classification.

The second method is a new phylogenetic regression model called Phylogenetic Relevance Vector Machine (PhyRVM). We present the first analytical solution for the phylogenetic signal λ and show the PhyRVM outperforms the widely used maximum likelihood approach Phylogenetic Least Squares (PGLS) on a simulated dataset and on the problem of predicting optimal growth temperature of archaea. We pursue this application further with the RVM as we investigate whether we can learn scientifically meaningful genomic correlates using the most relevant features. Our trained RVM model achieves state-of-the-art performance for archaeal OGT prediction and predicts a hyperthermophilic last universal common ancestor. The final method we present is a new phylogenetic dimensionality reduction technique called Phylogenetic Probabilistic Principal Components Analysis (P3CA). The advantage of P3CA is that it is a probabilistic model so it can optimise the phylogenetic signal λ by maximising the likelihood.

DEDICATION AND ACKNOWLEDGEMENTS

I would like to thank Colin Campbell and Tom Gaunt for giving me the freedom and encouragement to follow my own research interests. I would like to thank Tom Williams for introducing me to the fascinating world of phylogenetics and for many interesting discussions on the mini milestones that made up this work.

I would like to thank Edmund Moody who first introduced me to the concept of 'phylogenetic signal' and who gathered the data and built the phylogenies necessary to test the phylogenetic comparative methods developed in this thesis. I would like to thank Mark Rogers for sharing with me the data for 'CScape'. I would also like to thank Bastien Boussau for hosting me in his lab at CNRS in Lyon and sharing with me his code for ancestral sequence reconstruction in RevBayes. I would also like to thank Paul Kirk for giving me the time and encouragement to finish any corrections while I was working at the Biostatistics Unit in Cambridge.

It was a pleasure to work in the Buncaer and to be able to exchange ideas with somebody from a completely different field while overlooking a picturesque view of Bristol.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ... [redacted] DATE: ..... 9/NOV./2021 .....

Table of Contents

List of Tables

Table **Page**

List of Figures

Figure **Page**

CHAPTER 1

INTRODUCTION

"Solve the problem of interest, do not solve more general problem as intermediate one... In this paper I give up this imperative."

— Vladimir. N. Vapnik

1.1 Bayesian Model Comparison

A central task in the empirical sciences is to make statistical inferences from experimental data. Given a statistical model which has some free parameters, fitting the model to the data (often called *learning* or *training*) involves inferring the values of these parameters. The parameters $\mathbf{w}$ can be inferred from the posterior distribution given the observed data D using Bayes' Theorem:

$$p(\mathbf{w}|D) = \frac{p(\mathbf{w},D)}{p(D)} = \frac{p(D|\mathbf{w})\,p(\mathbf{w})}{\int p(\mathbf{w},D)\,d\mathbf{w}} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}} \tag{1.1}$$

The *integral* in the denominator of (1.1) is called the *evidence*, *marginal likelihood* or *normalising constant* and it is usually analytically intractable. The bulk of the computation is therefore spent on sampling from the posterior which can lead to Bayesian inference seeming too impractical for many high-dimensional problems [4]. In this thesis, we will focus on ways of approximating Bayesian inference, which means reformulating the learning task as an *optimization* problem, by approximating this integral with a simpler one which is analytically tractable and iteratively updating the approximations. This leads to learning algorithms which are fast, accurate and deterministic.

The method of maximum likelihood is a common choice for making inferences about parameters from the available training data [35]. The problem with fitting models to data by maximising the likelihood function $p(D|\mathbf{w})$ in (1.1) is that it suggests to use the most complex[1] models out of those under consideration. Overly complex models (such as high order polynomials) will fit the training data much better than simple models but can vary significantly between sub-samples of training data. In statistical parlance, we say that the overly complex models have low bias and high variance or *overfit* the training data [5]. On the other hand, overly simple models may not fit well but will not vary much either - they have a high bias and low variance or *underfit* the training data. The best model should minimise the trade-off between bias and variance or overfitting and underfitting [5]. A straight-forward solution to select the best model is to use out-of-sample prediction methods like *cross-validation*. The downside of these approaches is that none of the models are trained on the full training data and cross validation does not always select the best model [69].

An alternative solution is to use the evidence[2] in Bayes' Theorem to inform model comparison. We will define the *model* as the functional form and set of all parameters in the likelihood and prior. We will compute the posterior over one parameter (in this case $\mathbf{w}$); and we will call the other parameters in the model the *hypothesis* $\mathscr{H}$. Given a set of candidate hypotheses $\mathscr{H}_i$ (from a *hypothesis space*), the posterior probability for each hypothesis is [58]:

$$p(\mathscr{H}_i|D) \propto p(D|\mathscr{H}_i)\, p(\mathscr{H}_i) \tag{1.2}$$

The *prior* $p(\mathscr{H}_i)$ expresses our prior belief in the plausibility of each hypothesis. Assuming we have no reason to favour one hypothesis over any other (we say 'assuming a *flat prior*'), we can rank the hypotheses solely using the evidence $p(D|\mathscr{H}_i)$. The evidence naturally incorporates Occam's razor which states that unnecessarily complex models should not be preferred to simple ones. Bayesian model comparison allows several models to be compared using the full training data in a consistent framework. The concept of Bayesian model comparison can be stated simply as: *the model with the largest evidence will make the observed data most probable.*

Note, equation (1.2) is not normalised. Therefore, the objective of Bayesian model comparison is most unlike cross-validation. It is not to pick the most adequate model from a

[1] Not to be confused with *algorithmic complexity* which measures the run-time of a model.

[2] Suppose for the moment that computing the evidence does not pose a significant hurdle.

finite hypothesis space, but to find the true model by continually trying different models and comparing them using the evidence. The process works because the true model will *on average* have the maximum evidence (see proof in appendix A.1 due to MacKay [58]).

1.2 Approximate Bayesian Inference

In this thesis, we will consider two frameworks, Expectation Propagation (EP) [63] and Evidence Approximation (EA) [58], to approximate posteriors using Gaussians. In EA learning, we will assume a Gaussian likelihood and Gaussian prior, which is a *conjugate prior* as it gives rise to a Gaussian posterior. We also assume a flat prior over hypotheses $p(\mathscr{H})$, and at each iteration, the hypothesis is updated by the current approximation to the posterior in order to maximise the evidence. This means EA explicitly performs Bayesian model comparison to prevent overfitting. In EP learning, the algorithm converges to the maximum of a particular (negative) energy function [63], which is not the evidence. Therefore, each iteration (over the entire dataset) does not necessarily improve the model. The EM-EP [50] algorithm provides a general framework for combining EP with Bayesian model comparison to select $\mathscr{H}$ by maximising a lower bound on the log evidence. EP has an advantage over Gaussian EA when the problem requires a non-Gaussian likelihood. However, EP makes a strict assumption that the samples are *independent* which is not appropriate for all types of data (e.g. biological species).

1.2.1 Evidence Approximation

Traditionally, the Evidence Approximation (EA) has only been used to model data assumed to have been drawn *independently* from a data generating distribution. However, sometimes the independence assumption is a nuisance. One example, which we look at in chapter 5 is *phylogenetic comparative methods* [38]. Here, the data represent species which are assumed to share a common evolutionary history. The comparative method is used to test whether a continuous phenotypic characteristic (or *trait*) of the species also shares an evolutionary history. Recently, phylogenetic comparative methods have been used to study relationships between feeding behaviour and brain volume in Neotropical bats [97] and body mass and behavioural dominance in hummingbirds [10]. To do phylogenetic regression in an evidence approximation framework, we express the evolutionary tree of the data (often called a *phylogeny*) as a covariance matrix [32] and

maximise the evidence to determine the value of a parameter λ, due to Pagel [78], which measures how much the phylogeny should be used (often called *phylogenetic signal* [92]) in the predictive model of the trait.

1.2.2 Expectation Propagation

To apply Expectation Propagation (EP), we assume that we already know the true values of $\mathscr{H}$ which we want to use to approximate $p(\mathbf{w}|D,\mathscr{H})$ using Bayes' Theorem (1.1). This is not as straightforward as it sounds as it still requires that we evaluate the evidence which is often an intractable integral. Even by approximating the likelihood with a product of independent distributions and using a Gaussian prior, there are too many integrals to be analytically tractable. The solution to this problem became known as *assumed-density filtering* (ADF), which appeared independently in the statistics [55], control theory [60] and machine learning [8] [76] literature, and was later extended to EP by Minka[3] [64]. The idea is to update the posterior approximation for each data point sequentially which only requires solving a single integral at a time. The only requirement is that this integral is analytically tractable which is the case for the examples considered. Nevertheless, assumed-density filtering is an online algorithm so it is necessarily biased by the arbitrary order in which the data points are processed. The EP algorithm extends ADF to make multiple passes through the data which can iteratively refine the posterior approximations until convergence. In chapters 3 and 4, we apply the EP framework on two different likelihoods, a mixture of signal and noise Gaussians called the clutter problem and a sigmoidal confidence measure for binary classification, and show how it can be improved by reformulating EP as a more general method which performs approximate inference of approximate posteriors, where exact inference as in EP is a special case. Minka [64] has shown that EP provides an approximation to the evidence at every iteration. We can use the evidence to select models which approach and converge to different local optima of a modified EP energy function.

1.3 Road map

Chapter 2 provides background material on the Bayesian Occam's razor and other frameworks for learning statistical models (maximum likelihood, VC dimension and minimum description length). There is also an introduction to the exponential family of

[3] A very similar algorithm was also presented by Opper and Winther [75] derived using techniques from statistical mechanics.

distributions and the Kullback-Leibler divergence for approximate inference.

Chapter 3 introduces a new parameter γ into the (Gaussian) canonical EP algorithm which can be optimised with the evidence and applies the new approach, which we call γ-EP, to the problem of separating Gaussian signal from Gaussian noise (called the *clutter problem*) resulting in improved performance.

Chapter 4 applies γ-EP to a logistic-type probit likelihood used for binary classification, on which canonical EP has previously shown excellent predictive performance [63] [90]. Remarkably, by setting $\gamma > 0$, the linear classifier automatically ignores redundant training examples. The γ-EP approach is extended to non-linear classification and applied to classify oncogenic single nucleotide variants.

Chapter 5 applies Bayesian model comparison to phylogenetic comparative methods for continuous traits using the evidence approximation. The phylogenetic relevance vector machine (PhyRVM) is derived, including a new analytical update for Pagel's λ by maximising the evidence. A new method for kernel *dimensionality reduction* called phylogenetic probabilistic principal components analysis (P3CA) is developed with closed-form solutions. We predict the *optimal growth temperature* (OGT) of prokaryotes and reconstruct the OGT of the *last universal common ancestor* (LUCA).

Chapter 6 summarises the main results of chapters 3-5. By maximising the evidence to select γ or λ for independent or evolutionarily dependent data respectively, the methods developed in this thesis are able to determine whether exact or approximate inference of Gaussian approximate posteriors is preferable. We also suggest some directions for future work on the PhyRVM and sparse Bayesian classification using γ-EP.

CHAPTER 2

BACKGROUND

"Nothing is more practical than a good theory."

— Vladimir N. Vapnik

This chapter describes the Bayesian Occam's razor in more detail and compares it to other commonly used statistical frameworks: maximum likelihood, VC dimension and minimum description length. We also introduce the Kullback-Leibler divergence and exponential family of distributions which are central to approximate inference methods such as Expectation Propagation and Variational Inference.

Likelihood and Evidence: A central concept in Bayesian inference is *marginalisation*, in which auxiliary variables are integrated out of a joint probability density. In the evidence approximation, we assume a flat prior over hypotheses $\mathcal{H}$, which leaves a single scalar parameter w to be integrated out:

$$p(D|\mathcal{H}) = \int p(D,w|\mathcal{H})\, dw \tag{2.1}$$

$$= \int p(D|w,\mathcal{H})\, p(w|\mathcal{H})\, dw \tag{2.2}$$

The two terms in the integral are: the prior $p(w|\mathcal{H})$, which expresses prior belief in the value of w; and the likelihood $p(D|w,\mathcal{H})$, which measures the predictions the model makes about the observed data D for a particular value of w. As the number of samples increases, the Gaussian posterior $p(w|D,\mathcal{H}) \propto p(D|w,\mathcal{H})\, p(w|\mathcal{H})$ tends to be sharply peaked at the true value of w [4]. The value of w at the peak is called the *maximum a posteriori* estimate w_{MP}. The evidence is given by the area under the un-normalised

posterior which can be approximated by the height of the peak times its width $\Delta w_{\text{posterior}}$, which represents the posterior uncertainty in w [58]:

$$p(D|\mathcal{H}) \approx p(D|w_{MP},\mathcal{H})\, p(w_{MP}|\mathcal{H})\, \Delta w_{\text{posterior}} \tag{2.3}$$

which can be restated as the product of the best fit likelihood $p(D|w_{MP},\mathcal{H})$ and an *Occam factor* $p(w_{MP}|\mathcal{H})\, \Delta w_{\text{posterior}}$ [34]. By assuming that the prior $p(w|\mathcal{H})$ is uniform over a large interval Δw_{prior}, the Occam factor = $\Delta w_{\text{posterior}}/\Delta w_{\text{prior}}$ [58] and the evidence is:

$$p(D|\mathcal{H}) \approx p(D|w_{MP},\mathcal{H})\, \frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}} \tag{2.4}$$

The maximum likelihood estimate is achieved by using a flat prior, which maximises Δw_{prior}, minimising the Occam factor and potentially causing the model to overfit. Therefore, maximising the evidence is a trade-off between maximising the fit to the data (likelihood) and the model complexity (Occam factor).

Minimum Description Length and Evidence: The *minimum description length* (MDL) principle [94] approaches the model comparison problem from a completely different angle, and yet nevertheless, it is very similar to the Bayesian approach. Suppose, a sender wishes to send a dataset D to a receiver using the shortest message possible. The naive approach would be to just transmit the data suitably encoded. A better solution is to first transmit a model M for generating the data using a message of length $L(M)$, then a second message of length $L(D|M)$ to correct the mistakes made by the model after observing the data. If we measure the amount of information in a message using the logarithm to the base e[1], the total description length is [4]:

$$\text{description length} = L(D|M) + L(M) \tag{2.5}$$

$$= -\log[p(D|M)\, p(M)] \tag{2.6}$$

The description length has a very similar functional form as the (negative log) evidence (2.3) and similarly it embodies Occam's razor. The best fit likelihood is given by $L(D|M)$ and the Occam factor is $L(M)$. A very simple model will provide a poor description of the data leading to a large correction term; and a very complex model will require fewer corrections but more information in the model term. Therefore, minimising the description length is also a trade-off between fit to the data (likelihood) and complexity (Occam factor).

[1]The units of measurement are 'nats'.

Akaike's information criterion (AIC) [2] and the Bayesian information criterion (BIC) [100] can be thought of as simple approximations to the MDL given by $2d - 2\log p(D|\mathbf{w}, \mathscr{H})$ and $d\log n - 2\log p(D|\mathbf{w}, \mathscr{H})$ respectively (where d is the number of parameters and n is the number of samples) [58]. Both AIC and BIC resemble a simple multivariate extension to the approximation of the evidence (2.4) by assuming each parameter has the same ratio $\Delta w_{\text{posterior}}/\Delta w_{\text{prior}}$ [5]:

$$\log p(D|\mathscr{H}) \approx \log p(D|w_{MP}, \mathscr{H}) + d\log\left(\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}}\right) \tag{2.7}$$

In this simple approximation the Occam factor penalty increases linearly with the number of parameters. The correct expression for the multivariate Gaussian Occam factor is [58]:

$$\text{Occam factor} = p(\mathbf{w}_{\text{MP}}|\mathscr{H})\,(2\pi)^{d/2}\,|\mathbf{\Sigma}|^{-1/2} \tag{2.8}$$

where $\mathbf{\Sigma} = \nabla\nabla \log p(\mathbf{w}|D, \mathscr{H})$ is the Hessian matrix which is the inverse of the posterior covariance.

VC dimension and Evidence: An alternative to Bayesian model selection for binary classification was given by Vapnik and Chervonenkis [117]. Binary classifiers learn a mapping (often called a *decision boundary*) from n samples of d-dimensional data $\mathbf{x} \in \mathscr{R}$ to a binary label $y = \pm 1$. The VC dimension D_{VC} is defined as the largest set of data that the model can classify perfectly for any labelling (often called *shattering*). For a linear classifier $D_{VC} = d$ [73]. Vapnik and Chervonenkis showed that if the number of examples is much greater than D_{VC} then a large difference between training error (the classification error on the training set) and generalisation error (the probability of a trained classifier misclassifying a test example $\mathbf{x}^*$) is very unlikely [73]. This motivates increasing the amount of data and decreasing D_{VC} (without significantly increasing the training error). Furthermore, Vapnik [118] related D_{VC} to a quantity called the *margin*, ρ, which is the least distance from the decision boundary to the training data:

$$D_{VC} \leq \frac{D_S}{\rho^2} \tag{2.9}$$

where D_S is the diameter of the smallest sphere containing the training data. This bound motivated the development of an algorithm which explicitly maximises the margin called the *Support Vector Machine* (SVM) [18]. The SVM uses only the examples which

lie on the margin, called *support vectors*, to construct a classifier[2]. The non-support vectors do not contribute to the decision boundary. A link between the Bayesian and maximum margin formalism was found by Herbrich and Graepel [43] who built on the PAC (Probably Approximately Correct) Bayesian theorems of D. McAllester [61] and the work of Herbrich, Graepel and Campbell [44]. For $\delta \in (0,1]$ with probability at least $1-\delta$ over a random training sample, the generalisation error of the Bayes optimal classifier, $R[Bayes(\mathbf{x}^*)]$, is bounded above by [44]:

$$R[Bayes(\mathbf{x}^*)] \leq \frac{2}{n}\left(\ln\left(\frac{1}{p(V(D))}\right) + 2\ln(n) + \ln\left(\frac{1}{\delta}\right) + 1\right) \tag{2.10}$$

where $p(V(D))$ is the prior over classifiers which perfectly separate the training data. The space of perfect separators is called *version space*. When we use the 'PAC-likelihood', given by 1 for a correct classification and 0 otherwise, in Bayes' Theorem, $p(V(D))$ is equal to the evidence [40]. Therefore, by maximising the (negative) log marginal likelihood (evidence) the bound (2.10) is minimised. There is also a PAC-Bayes margin bound for the generalisation error. For this we will need to normalise the margin ρ by the norm of each training example, $\bar{\rho} = \rho/\|\mathbf{x}\|$ to keep the margin within version space. For $\delta \in (0,1]$ with probability at least $1-\delta$ over a random training sample and $h \in V(D)$, the generalisation error of a linear classifier h correctly classifying n samples with a positive normalised margin $\bar{\rho}$ is bounded above by [43]:

$$R[h] \leq \frac{2}{n}\left(m\ln\left(\frac{1}{1-\sqrt{1-\bar{\rho}^2}}\right) + 2\ln(n) + \ln\left(\frac{1}{\delta}\right) + 1 + \ln(2)\right) \tag{2.11}$$

where $m = \min(n,d)$. For maximum margin linear classifiers, $\bar{\rho} = 1$, the margin term in (2.11) vanishes and similarly for maximum evidence classifiers, the (negative) log evidence term in (2.10) becomes very small. Maximising the evidence or the margin leads to better generalisation.

These inequalities belie a key difference between the Bayesian and maximum margin formalism. If all of the models under consideration are ill-suited to the data, then the evidence may not be correlated with the generalisation error [58]. This sounds like a strange flaw in the Bayesian framework but actually it is one of its greatest strengths. Whereas, increasing the margin provably decreases the generalisation error [117], the SVM practitioner is unaware if a better model is needed. But the Bayesian, after checking

[2]Technically, this is called a hard-margin SVM. The soft-margin SVM also includes data points which lie within the margin and misclassifications as support vectors.

the value of the evidence, is aware of this failure of the model and is motivated to search for new models which predict the data with higher probability.

Exponential Family: The exponential family is a set of probability distributions $p(\mathbf{x})$ which can be written in a particular form given by:

$$p(\mathbf{x}) = \frac{1}{Z(\boldsymbol{\eta})}\exp\left(\mathrm{Tr}[\boldsymbol{\eta}T(\mathbf{x})]\right) \tag{2.12}$$

where the *trace* (Tr) is included in the exponential for multivariate distributions. We call $T(\mathbf{x})$ the *natural statistic* of $\mathbf{x}$, $\boldsymbol{\eta}$ is the *natural parameter* and $Z(\boldsymbol{\eta}) = \int \exp\left(\mathrm{Tr}[\boldsymbol{\eta}T(\mathbf{x})]\right) d\mathbf{x}$ is the normalising constant. The exponential family includes the Gaussian distribution among many other standard distributions. Consider the multivariate Gaussian distribution as an example:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\left(-\frac{1}{2}\mathrm{Tr}[(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})]\right) \tag{2.13}$$

$$= \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\left(-\frac{1}{2}\mathrm{Tr}[\boldsymbol{\Sigma}^{-1}\mathbf{x}\mathbf{x}^T - 2\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu}]\right) \tag{2.14}$$

$$= \frac{1}{Z(\boldsymbol{\eta})}\exp\left(\mathrm{Tr}[\boldsymbol{\eta}T(\mathbf{x})]\right) \tag{2.15}$$

where $T(\mathbf{x}) = (\mathbf{x}, \mathbf{x}\mathbf{x}^\mathbf{T})$ and $\boldsymbol{\eta} = (\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}, -\frac{1}{2}\boldsymbol{\Sigma}^{-1})$ and $Z(\boldsymbol{\eta}) = (2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}\exp\left(\frac{1}{2}\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu}\right)$. The expectation of the natural statistic is related to the normalising constant through the formula [42]:

$$\nabla_{\boldsymbol{\eta}}\log(Z(\boldsymbol{\eta})) = \frac{\int\left[\nabla_{\boldsymbol{\eta}}\exp\left(\mathrm{Tr}[\boldsymbol{\eta}T(\mathbf{x})]\right)\right] d\mathbf{x}}{Z(\boldsymbol{\eta})} = E[T(\mathbf{x})] \tag{2.16}$$

Kullback-Leibler Divergence: Suppose we have any arbitrary distribution $p(\mathbf{x})$ and we want to find the best Gaussian approximation $q(\mathbf{x})$. To do this we will need to minimise the dissimilarity between probability densities. The *Kullback-Leibler divergence* (KL-divergence) [53] is a convenient quantity to use to measure dissimilarity because it is always greater than or equal to zero ($KL(p||q) = 0$ if $p = q$):

$$KL(p||q) = \int p(\mathbf{x})\log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) d\mathbf{x} \tag{2.17}$$

The minimum of $KL(p||q)$ where $q(\mathbf{x})$ is in the exponential family is given by matching expected natural statistics of $q(\mathbf{x})$ to $p(\mathbf{x})$ (often called *moment matching*) (see proof in appendix A.2 [42]). Assumed-density filtering (ADF) performs moment matching sequentially one data point at a time. Expectation Propagation (EP) extends ADF by

iterating through the data points multiple times until convergence.

It is important to note that $KL(p||q) \neq KL(q||p)$ and minimising $KL(q||p)$ will lead to a different family of algorithms called *variational inference* [82]. The log evidence $\log(p(\mathbf{x}))$ can be decomposed into a lower bound $\mathscr{L}(q)$ and a KL divergence term:

$$\log(p(\mathbf{x})) = \mathscr{L}(q) + \mathrm{KL}(q||p) \geq \mathscr{L}(q) \tag{2.18}$$

Variational inference maximises this lower bound (often called *evidence lower bound* (ELBO) [6]) which is equivalent to minimising the $KL(q||p)$ in (2.18). In Expectation Propagation, it is a strict requirement that $q(\mathbf{x})$ is an exponential family distribution and that it *factorises over the data* (independence assumption). On the other hand, variational inference does not have to make any strict requirements on $q(\mathbf{x})$ [5] but often it is assumed that it *factorises over the variables* (called *mean-field theory* approximations [79]) and that $q(\mathbf{x})$ is an exponential family distribution to make the optimisation have simple analytical solutions. In practice, EP approximates the posterior moments well (by moment matching), but it can be misled by multi-modal distributions as it tries to average across all modes [5]. Mean-field variational inference provides a complimentary alternative by capturing any individual posterior mode well but the approximations will underestimate the posterior variance [6]. Finally, the KL-divergence is a member of a more general family of *alpha divergences* [66]. The other divergences in this family may be more difficult to optimise but may also improve accuracy [65].

CHAPTER 3

EXTENSIONS OF GAUSSIAN EXPECTATION PROPAGATION

"The worth of an algorithm is not always what it seems."

— Thomas P. Minka

This chapter develops a proposed modification of the canonical Expectation Propagation (EP) algorithm by including a new bias term in the approximate factors. The modified EP algorithm, called γ-EP, is equivalent to canonical EP when the coefficient of the bias term γ is set to -1. The value of γ can be tuned by maximising the evidence to achieve superior accuracy to canonical EP on the clutter problem.

3.1 The Clutter Problem

Empirical data gathered from sensor readings or laboratory experiments is the lifeblood of the scientific enterprise. It is crucial that these measurements are made accurately and consistently, because without that most statistical tools will be of little benefit, but in certain special cases even a considerable amount of noise can be tolerated if it can be modelled properly. One example which we will look at in this chapter is called the *clutter problem* [64]. The clutter problem assumes that the *signal* (or distribution of interest) is distributed by a Gaussian with mean $\boldsymbol{\theta}$ and the noise is distributed by a separate Gaussian such that the observed data is assumed to be distributed by a *mixture* of both Gaussians. That is to say, the noise Gaussian is weighted by the known proportion

$w \in [0,1]$ of zero-mean background clutter and the signal by one minus this proportion:

$$p(\mathbf{x}|\boldsymbol{\theta}) = (1-w)\mathcal{N}(\mathbf{x}|\boldsymbol{\theta},\mathbf{I}) + w\mathcal{N}(\mathbf{x}|\mathbf{0},a\mathbf{I}) \tag{3.1}$$

A random variable $\mathbf{x} \in \mathscr{R}^d$ distributed by a multivariate Gaussian $\mathcal{N}(\mathbf{x}|\mathbf{m},\mathbf{V})$ is defined entirely by its mean vector $\mathbf{m}$ and covariance matrix $\mathbf{V}$ through the probability density function:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{m},\boldsymbol{V}) = \frac{\exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{m})^T\boldsymbol{V}^{-1}(\mathbf{x}-\boldsymbol{m})}{|2\pi\boldsymbol{V}|^{\frac{1}{2}}} \tag{3.2}$$

The classical result for the maximum likelihood estimate of $\mathbf{m}$ is the sample mean (denoted by $\bar{\mathbf{x}}$):

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \tag{3.3}$$

If we try to estimate θ with the sample mean we will find that it gets worse and worse as $w \to 1$ as it includes a greater proportion of noise to signal. A Bayesian treatment of the problem will not only give an estimate of $\boldsymbol{\theta}$, but also the variance or uncertainty in the estimates. A necessary addition in the Bayesian treatment is the prior which we choose for convenience[1] to be Gaussian over the d-dimensional mean vector $\boldsymbol{\theta}$:

$$p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, b\mathbf{I}) \tag{3.4}$$

Typically, Bayesian inference is performed by building up an empirical approximation to an exact posterior in such a way that the approximation gets more accurate over time and exact in the infinite limit. These are known as Monte Carlo methods [5]. While they have the theoretical advantage of asymptotic exactness, they are slow and the approximations are non-deterministic.

Expectation Propagation (EP), which grew out of a method called assumed-density filtering (ADF), is an approximate Bayesian inference technique that is fast and deterministic. It is deterministic because the posterior approximations can be calculated analytically and it is fast because it converges very quickly to a fixed-point[2] (of which there can be multiple). The first iteration of EP is typically initialised to be equivalent to assumed-density filtering (ADF) and further iterations can refine the posterior approximations. However, these further iterations are not guaranteed to improve the original

[1] The prior will be used as the initialisation of the posterior in ADF and EP.

[2] A fixed-point is reached when the input is equal to the output of each iteration.

ADF approximation. They are not backed-up by Bayesian model comparison, for instance. In this chapter, we develop a method called γ-EP which encourages the EP iterations to find other fixed-points which make the data more probable by maximising the evidence.

In section 3.2, we introduce spherical Gaussian assumed-density filtering, Expectation Propagation and a new EP update called 'reuse' and demonstrate its potential to improve performance. In section 3.3, we demonstrate that EP iterations can lead to a worse solution than ADF and derive the γ-EP variant of EP (which is equivalent to EP with $\gamma = -1$). In section 3.3.1, we compare ADF, EP and γ-EP for various levels of background clutter. In section 3.3.2, we reformulate γ-EP as a constrained optimization problem to explain how it works by maximising the evidence and, using Lagrange multipliers, introduce a new method called $\lambda\gamma$-EP which gives statistically significant performance improvements over ADF and EP in low clutter levels.

3.2 Spherical Gaussian Expectation Propagation

We will introduce the Expectation Propagation algorithm for the clutter problem using a *spherical* Gaussian posterior approximation:

$$q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{m}_\theta, v_\theta \mathbf{I}) \tag{3.5}$$

The advantage of using a spherical Gaussian instead of a diagonal or full covariance Gaussian is shorter run-time. Although a richer approximating distribution can capture more of the posterior probability mass, if we can get performance which is as good by only a single variance parameter, then we can make significant computational savings when the posterior is very high-dimensional.

We assume the observed data, $D = (\boldsymbol{x_1}, ..., \boldsymbol{x_n})$ are independent so the likelihood is given by a product of (3.1) for every data point and the joint distribution of D and $\boldsymbol{\theta}$ is given by the prior times the likelihood:

$$p(D, \boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{i=1}^{n} p(\mathbf{x}_i|\boldsymbol{\theta}) \tag{3.6}$$

We can also consider writing $p(D, \boldsymbol{\theta})$ as a product of independent factors:

$$p(D, \boldsymbol{\theta}) = \prod_{i=0}^{n} t_i(\boldsymbol{\theta}) \tag{3.7}$$

where $t_0(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$ and $t_i(\boldsymbol{\theta}) = p(\mathbf{x}_i|\boldsymbol{\theta})$ for $i = 1,...,n$. We will assume $a = 10$ and $b = 100$ as in [63] for direct comparison of our proposed modification to Minka's canonical formulation of EP.

Assumed Density Filtering (ADF) [64] makes one forward pass through the data and sequentially incorporates each factor t_i step-by-step into a Gaussian approximation to the posterior. The 'new' posterior at each step forms the prior (or 'old' posterior $q^{\setminus i}(\boldsymbol{\theta}|m_\theta^{\setminus i}, v_\theta^{\setminus i})$) for the next step. The 'new' posterior is updated by minimising $\mathrm{KL}(q^*||q^{\setminus i})$ which is called *moment matching*, where the exact posterior is given by:

$$q^*(\boldsymbol{\theta}) = \frac{t_i(\boldsymbol{\theta})q^{\setminus i}(\boldsymbol{\theta})}{\int t_i(\boldsymbol{\theta})q^{\setminus i}(\boldsymbol{\theta})d\boldsymbol{\theta}} \tag{3.8}$$

The denominator of (3.8) is the normalising constant for each step which has an analytical solution for Gaussian posteriors [64]:

$$Z_i = (1-w)\mathcal{N}(\mathbf{x}_i|m_\theta^{\setminus i}, (v_\theta^{\setminus i}+1)\mathbf{I}) + w\mathcal{N}(\mathbf{x}_i|\mathbf{0}, 10\mathbf{I}) \tag{3.9}$$

The moment matching update equations for the posterior mean and variance are given by (see proof in appendix A.3 [41]):

$$m_\theta = m_\theta^{\setminus i} + v_\theta^{\setminus i}\frac{\partial \log Z_i}{\partial m_\theta^{\setminus i}} \tag{3.10}$$

$$v_\theta = v_\theta^{\setminus i} d - (v_\theta^{\setminus i})^2\left[\left|\frac{\partial \log Z_i}{\partial m_\theta^{\setminus i}}\right|^2 - 2\frac{\partial \log Z_i}{\partial v_\theta^{\setminus i}}\right] \tag{3.11}$$

Moment matching with a full Gaussian covariance requires a matrix inversion, so ADF takes $O(nd^3)$ time to solve the clutter problem. Whereas, using a spherical Gaussian instead, ADF takes $O(nd)$ time. There is no risk of overfitting by estimating the full covariance matrix because it only improves the accuracy of the approximations for the given class of posterior distribution (e.g. multivariate Gaussian). ADF is limited by treating each factor t_i exactly as it can only make one pass through the data before the full posterior approximation is finished. Therefore, it is biased by the arbitrary ordering of the data. A more powerful technique, called Expectation Propagation (EP) [64], begins by approximating each of the factors $t_i(\boldsymbol{\theta})$ with a Gaussian *approximate factor* $\tilde{t}_i(\boldsymbol{\theta})$:

$$q(\boldsymbol{\theta}) = \frac{\prod_i \tilde{t}_i(\boldsymbol{\theta})}{\int \prod_i \tilde{t}_i(\boldsymbol{\theta})\, d\boldsymbol{\theta}} \tag{3.12}$$

While it is not possible to determine the approximating distribution $q(\boldsymbol{\theta})$ which minimises $KL(q^*||q)$ as it would require averaging with respect to the true distribution [5], EP can

make local approximations by updating the approximate factors sequentially. As each approximate factor is updated independently of the order they appear, the algorithm can iterate over all of the data points multiple times, refining the posterior estimates until convergence.

The EP algorithm is initialised such that $\tilde{t}_i(\boldsymbol{\theta}) = 1$ for $i = 1,...,n$. It then proceeds by iterating the following three steps until all of the approximate factors converge.
For every data point i:
(1) Remove an approximate factor from the posterior to get an 'old' posterior:

$$q^{\setminus i}(\boldsymbol{\theta}) \propto \frac{q(\boldsymbol{\theta})}{\tilde{t}_i(\boldsymbol{\theta})} \tag{3.13}$$

(2) Compute 'new' posterior $q(\boldsymbol{\theta})$ via (3.10) and (3.11) by minimising $KL(q^*||q^{\setminus i})$.
(3) Update the approximate factor:

$$\tilde{t}_i(\boldsymbol{\theta}) = Z_i \frac{q(\boldsymbol{\theta})}{q^{\setminus i}(\boldsymbol{\theta})} \tag{3.14}$$

We suppose that by reusing the posterior mean as the approximate factor update, we can achieve better accuracy. We call this update: 'reuse'. As a motivating example, consider the simple problem of estimating the mean of a stream of univariate Gaussian distributed data without clutter. For this example we will focus on ADF. By defining approximate factor updates as in canonical EP [63]:

$$v_i^{-1} = v_\theta^{-1} - \left(v_\theta^{\setminus i}\right)^{-1} \tag{3.15}$$

$$m_i = m_\theta^{\setminus i} + (v_i + v_\theta^{\setminus i})\left(v_\theta^{\setminus i}\right)^{-1}\left(m_\theta - m_\theta^{\setminus i}\right) \tag{3.16}$$

The factor approximation is given by:

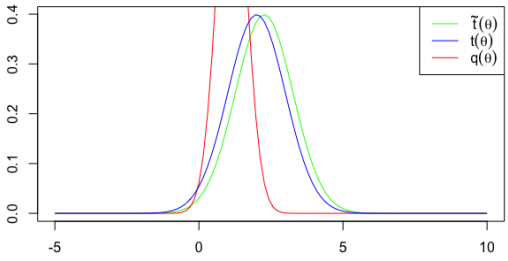
$$\tilde{t}_i(\theta) = \frac{Z_i(m_\theta^{\setminus i}, v_\theta^{\setminus i})}{\mathcal{N}(m_i|m_\theta^{\setminus i}, (v_i + v_\theta^{\setminus i}))}\left(\frac{v_i + v_\theta^{\setminus i}}{v_i}\right)^{\frac{d-1}{2}} \mathcal{N}(\theta|m_i, v_i\mathbf{I}) \tag{3.17}$$

The factor approximation is not a proper Gaussian, it is scaled by a constant and the variance, v_i, can be negative. Now, suppose we remove the v_i term from (3.16), then we find the individual approximate factor mean is equivalent to the approximate posterior mean, giving the 'reuse' update:

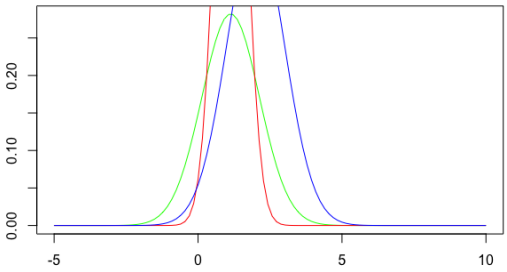
$$m_i = m_\theta \tag{3.18}$$

In this case the factor approximation becomes:

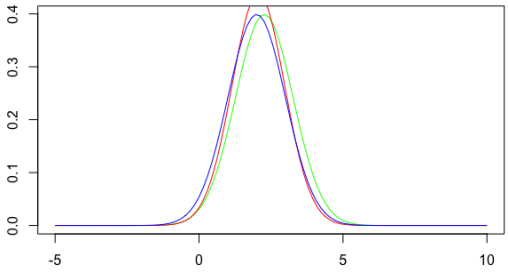
$$\tilde{t}_i(\theta) = \frac{Z_i(m_\theta^{\setminus i}, v_\theta^{\setminus i})}{\mathcal{N}(m_i|m_\theta^{\setminus i}, v_\theta^{\setminus i})}\left(\frac{v_i + v_\theta^{\setminus i}}{v_i}\right)^{\frac{d}{2}}\left(\frac{v_i}{v_i^{\setminus i}}\right)^{\frac{1}{2}} \mathcal{N}(\theta|m_i, v_i\mathbf{I}) \tag{3.19}$$



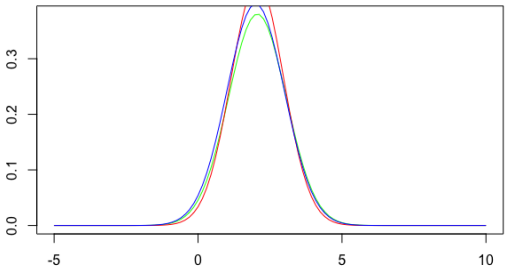
(a) $m_\theta^{\setminus i} = 0$, $v_\theta^{\setminus i} = 1, m_i = 2.27$



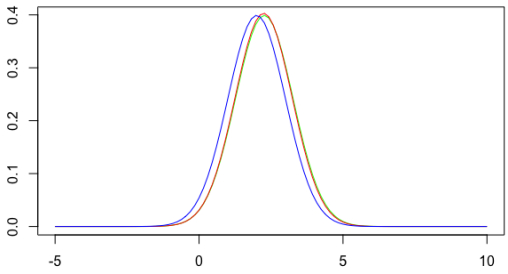
(b) $m_\theta^{\setminus i} = 0$, $v_\theta^{\setminus i} = 1, m_i = 1.13$



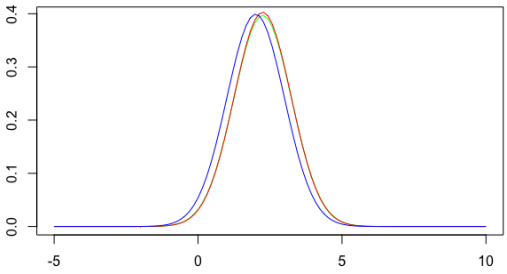
(c) $m_\theta^{\setminus i} = 0$, $v_\theta^{\setminus i} = 10, m_i = 2.27$



(d) $m_\theta^{\setminus i} = 0$, $v_\theta^{\setminus i} = 10, m_i = 2.06$



(e) $m_\theta^{\setminus i} = 0$, $v_\theta^{\setminus i} = 100, m_i = 2.27$



(f) $m_\theta^{\setminus i} = 0$, $v_\theta^{\setminus i} = 100, m_i = 2.24$

Figure 3.1: Examples of fitting a single factor using ADF with the canonical update (left) and the 'reuse' update (right) with $\theta = 2$, $m_i^{\setminus i} = 0$ and $v_i^{\setminus i} \in \{1, 10, 100\}$

Notice, the variance of the Gaussian in the denominator of (3.19) does not include v_i. If we hold $m_i^{\setminus i}$ and $v_i^{\setminus i}$ constant and run ADF, then (3.17), which includes v_i, will compensate for the error in $v_i^{\setminus i}$ and $\tilde{t}_i(\theta)$ will remain the same for various settings of $v_i^{\setminus i}$, as shown in Figure 3.1 (a), (c) and (e). Whereas, the approximate factor (3.19) which does not include v_i, can vary significantly depending on the value of $v_i^{\setminus i}$, as shown in Figure 3.1 (b), (d) and (f). As $v_i^{\setminus i}$ increases, the canonical EP update (3.16) gets closer to the 'reuse' update (3.18) and therefore $q(\theta)$ is attracted to $\tilde{t}_i(\theta)$ so Figures 3.1 (e) and (f) are very similar. The 'reuse' update can lead to catastrophic approximations if the posterior does not approximate $t_i(\theta)$ very well, as shown in Figure 3.1 (b). However, when $v_i^{\setminus i} = 10$, $q(\theta)$ passes through a very good approximation to $t_i(\theta)$ in Figure 3.1 (d).

It is worth noting that the approximate factor updates can be written in a slightly simpler form using *natural parameters* (η_i, τ_i). The natural parameters of a univariate Gaussian are $(\eta_i = m_i v_i^{-1}, \tau_i = -\frac{1}{2} v_i^{-1})$. The τ_i update is identical to the v_i update (3.15) and the η_i update is [90]:

$$\eta_i = m_\theta v_\theta^{-1} - \eta_\theta^{\setminus i} \tag{3.20}$$

Similarly, we can also write 'old' posterior updates in terms of natural parameters $(\eta_\theta^{\setminus i}, \tau_\theta^{\setminus i})$. However, the 'reuse' update cannot be seen in (3.20) and the 'new' posterior updates (3.10) and (3.11) must be computed in the mean and variance representation. Therefore, instead of swapping back and forth between expected natural statistics and natural parameters, we will stick to using the mean and variance representation throughout.

3.3 Extensions of Gaussian Expectation Propagation

So far we have introduced a new way to update the approximate factor means so that they all converge to the posterior mean. Now, we will demonstrate the benefit of this. The EP algorithm can be reformulated as a min-max optimization of a particular energy function [63]. However, each iteration of canonical EP is not guaranteed to decrease this energy function [5]. As spherical Gaussian EP is limited by sharing one variance parameter across all dimensions, then depending on the sample, the EP iterations can improve or ruin a good first iteration (or ADF approximation).

Consider the ADF and EP estimation of the distribution of interest (with mean $\boldsymbol{\theta}$)

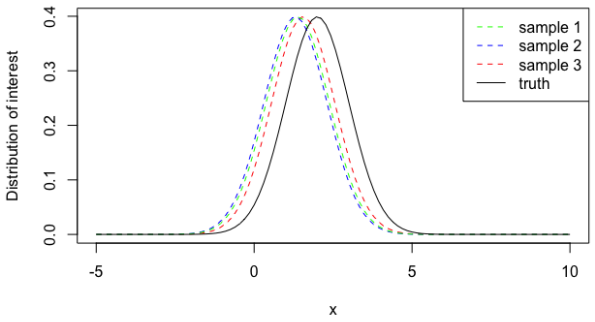


Figure 3.2: ADF approximation.

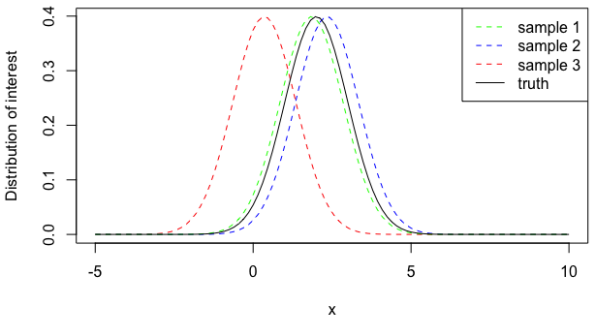


Figure 3.3: EP approximation.

for three different random samples shown in Figures 3.2 and 3.3 respectively. The parameters used are $w = 0.7$, $\theta = 2$ and $n = 20$. We have chosen three samples for which the ADF estimates are all quite similar, fairly good and below θ ($m_\theta^{red} = 1.55, m_\theta^{blue} = 1.31, m_\theta^{green} = 1.38$). Further EP iterations show improvements for the 'blue' and 'green' samples but the 'red' sample, which was the best for ADF, is now the worst for EP. The sample means are: $\bar{x}^{red} = 0.924$, $\bar{x}^{blue} = 2.85, \bar{x}^{green} = 3.43$. Here, m_θ^{EP} improves on m_θ^{ADF} for samples with $\bar{x} > m_\theta^{ADF}$ and worsens m_θ^{ADF} for samples with $\bar{x} < m_\theta^{ADF}$, as shown in Figure 3.4 for 5 different orderings of the 'red' sample corresponding to 5 different ADF estimates which all converge to the same EP fixed-point.

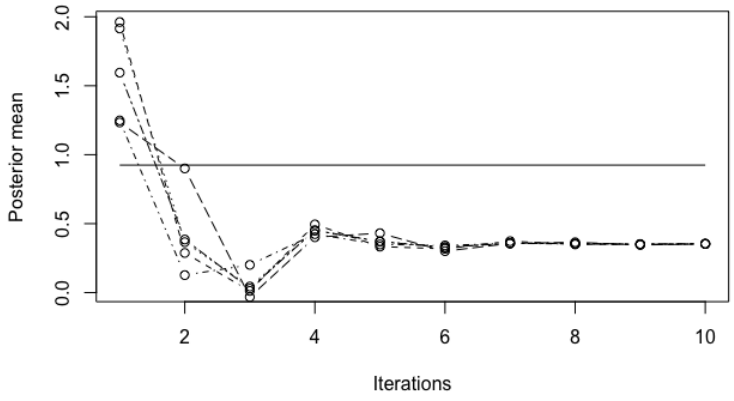


Figure 3.4: A plot of the m_θ trajectories for 5 different orderings of the 'red' sample. The sample mean is given by the solid line. The true value of the mean is 2. In all examples EP significantly worsens a good first iteration (ADF approximation).

Canonical EP makes a Gaussian approximation $\tilde{t}_i$ to t_i and then computes an exact posterior including $\tilde{t}_i$:

(3.21) 'new' posterior = 'old' posterior + approximate factor

The new method, called γ-EP, makes a Gaussian approximation $\tilde{t}_i$ to t_i and then computes an *approximate* posterior including $\tilde{t}_i$ with the 'reuse' update (3.18), where canonical EP is a special case in which the 'old' posterior incorporates the bias term to recover (3.21):

(3.22) 'new' posterior = 'old' posterior + approximate factor + bias

We can force EP to find other fixed-points which make the data more probable by multiplying the bias term by a parameter γ and maximising the evidence to choose its value.

The 'old' posterior mean and variance in canonical EP are:

$$(v_\theta^{\setminus i})^{-1} = v_\theta^{-1} - v_i^{-1} \tag{3.23}$$

$$m_\theta^{\setminus i} = v_\theta^{\setminus i}(v_\theta^{-1} m_\theta - v_i^{-1} m_i) \tag{3.24}$$

$$= m_\theta + v_\theta^{\setminus i} v_i^{-1}(m_\theta - m_i) \tag{3.25}$$

By rearranging (3.24) we get a mathematical expression for (3.21):

$$v_\theta^{-1} m_\theta = (v_\theta^{\setminus i})^{-1} m_\theta^{\setminus i} + v_i^{-1} m_i \tag{3.26}$$

The moment matching updates for the clutter problem used in ADF and EP are:

$$r_i = 1 - \frac{1}{Z_i} w \mathcal{N}(x_i | 0, 10\mathbf{I}) \tag{3.27}$$

$$\nabla_m = r_i \frac{x_i - m_\theta^{\setminus i}}{v_\theta^{\setminus i} + 1} \tag{3.28}$$

$$m_\theta = m_\theta^{\setminus i} + v_\theta^{\setminus i} \nabla_m \tag{3.29}$$

$$v_\theta = v_\theta^{\setminus i} - \frac{1}{d}(v_\theta^{\setminus i})^2 \Gamma \tag{3.30}$$

where $\nabla_m = \frac{\partial \log Z_i}{\partial m_\theta^{\setminus i}}$ and $\nabla_v = \frac{\partial \log Z_i}{\partial v_\theta^{\setminus i}}$ and $\Gamma = \nabla_m^T \nabla_m - 2\nabla_v$. By multiplying v_θ by the d-dimensional identity matrix, we can use the Woodbury identity [81] to find its inverse:

$$v_\theta^{-1}\mathbf{I} = (v_\theta^{\setminus i})^{-1}\mathbf{I} + (\Gamma^{-1}\mathbf{I} - v_\theta^{\setminus i}\mathbf{I})^{-1} \tag{3.31}$$

Multiplying $v_\theta^{-1}\mathbf{I}$ by m_θ gives:

$$v_\theta^{-1}\mathbf{I} m_\theta = (v_\theta^{\setminus i})^{-1}\mathbf{I} m_\theta^{\setminus i} + (\Gamma^{-1}\mathbf{I} - v_\theta^{\setminus i}\mathbf{I})^{-1}\left[m_\theta^{\setminus i} + v_\theta^{\setminus i}\nabla_m\right] + \nabla_m \tag{3.32}$$

$$= (v_\theta^{\setminus i})^{-1}\mathbf{I} m_\theta^{\setminus i} + (v_i)^{-1}\mathbf{I}\left[m_\theta^{\setminus i} + v_\theta^{\setminus i}\nabla_m + v_i \nabla_m\right] \tag{3.33}$$

By comparing (3.33) with (3.26) we recover the canonical EP update equations (3.15) and (3.16). By comparing (3.32) and (3.26), we recover the 'reuse' update (3.18) instead of (3.16). This new update does not fully factorise. There is now an additive ∇_m bias term. It is interesting to note that ∇_m is a function of r_i (3.27), which is the probability of the ith data point not being clutter. This means data points with high probability of being clutter do not significantly contribute to the posterior approximations. To derive a new expression for the 'old' posterior mean consistent with canonical EP, we start with (3.25) and plug in the previous EP updates until we have it in terms of the 'reuse' update m_i^{reuse}.

Plugging (3.16) into (3.25) gives:

$$(m_\theta^{\setminus i})^{new} = m_\theta^{new} + v_\theta^{\setminus i} v_i^{-1}(m_\theta^{new} - m_i) \tag{3.34}$$

$$= m_\theta^{new} + v_\theta^{\setminus i} v_i^{-1}(m_\theta^{new} - m_\theta^{\setminus i} + (v_i + v_\theta^{\setminus i})(v_\theta^{\setminus i})^{-1}(m_\theta - m_\theta^{\setminus i})) \tag{3.35}$$

$$= m_\theta^{new} + v_\theta^{\setminus i} v_i^{-1}(m_\theta^{new} - m_\theta^{old} + v_i(v_\theta^{\setminus i})^{-1}(m_\theta - m_\theta^{\setminus i})) \tag{3.36}$$

Plugging (3.29) into (3.36) gives:

$$(m_\theta^{\setminus i})^{new} = m_\theta^{new} + v_\theta^{\setminus i} v_i^{-1}(m_\theta^{new} - m_\theta^{old} - v_i \nabla_m) \tag{3.37}$$

$$= m_\theta^{new} + v_\theta^{\setminus i} v_i^{-1}(m_\theta^{new} - m_i^{reuse} - v_i \nabla_m) \tag{3.38}$$

Therefore, (3.38) and the 'reuse' update (3.18) is equivalent to using (3.25) and the canonical update (3.16). We call this algorithm γ-EP because we multiply the bias term $v_i \nabla_m$ by a parameter γ. Whenever we include the bias term, m_i^{reuse} will be shortened to m_i. In section 3.2, we demonstrated that the 'reuse' update can outperform the canonical update on a particular factor with an appropriate choice of $v_\theta^{\setminus i}$. The same is true for ∇_m (3.28) which is a function of $v_\theta^{\setminus i}$.

In order to derive (3.38) we have to remove an approximate factor $\tilde{t}_i$ from the posterior to get an 'old' posterior, where $\tilde{t}_i$ is given by:

$$\tilde{t}_i(\theta) = s_i \exp\left(-\frac{1}{2v_i}(\theta - m_i + \gamma v_i \nabla_m)^T(\theta - m_i + \gamma v_i \nabla_m)\right) \tag{3.39}$$

We will show γ-EP with $\gamma = -1$ is equivalent to canonical EP. By 'completing the square' in the exponential for θ we get expressions for $(m_\theta^{\setminus i}, v_\theta^{\setminus i})$:

$$\frac{1}{2}\left((\theta - m_\theta)^T v_\theta^{-1}\mathbf{I}(\theta - m_\theta) - (\theta - m_i + \gamma v_i \nabla_m)^T v_i^{-1}\mathbf{I}(\theta - m_i + \gamma v_i \nabla_m)\right) \tag{3.40}$$

=

$$\frac{1}{2}\left((\theta - m_\theta^{\setminus i})^T (v_\theta^{\setminus i})^{-1}\mathbf{I}(\theta - m_\theta^{\setminus i}) - (m_\theta^{\setminus i})^T (v_\theta^{\setminus i})^{-1}\mathbf{I}m_\theta^{\setminus i} + m_\theta^T v_\theta^{-1}\boldsymbol{I}m_\theta - (v_i\nabla_m - \gamma m_i)^T v_i^{-1}\mathbf{I}(v_i\nabla_m - \gamma m_i)\right)$$

where the 'old' posterior mean is given by:

$$m_\theta^{\setminus i} = v_\theta^{\setminus i}(v_\theta^{-1}m_\theta - v_i^{-1}m_i + \gamma\nabla_m) \tag{3.41}$$

$$= m_\theta + v_\theta^{\setminus i}v_i^{-1}(m_\theta - m_i + \gamma v_i\nabla_m) \tag{3.42}$$

which is equivalent to (3.38) with $\gamma = -1$. The variance update $v_\theta^{\setminus i}$ is the same as canonical EP and is given by (3.23). By 'completing the square' in the exponential for $v_i\nabla_m$ and using (3.23), we can express (3.40) entirely in terms of $m_\theta^{\setminus i}$:

$$-(m_\theta^{\setminus i})^T (v_\theta^{\setminus i})^{-1}\mathbf{I}m_\theta^{\setminus i} + m_\theta^T v_\theta^{-1}\mathbf{I}m_\theta - (v_i\nabla_m - \gamma m_i)^T v_i^{-1}\mathbf{I}(v_i\nabla_m - \gamma m_i)$$

$$=$$

$$(m_\theta - m_\theta^{\setminus i})^T v_\theta^{-1}\mathbf{I}(m_\theta - m_\theta^{\setminus i}) + (v_i\nabla_m - \gamma(m_\theta^{\setminus i} - m_i))^T v_i^{-1}\mathbf{I}(v_i\nabla_m - \gamma(m_\theta^{\setminus i} - m_i))$$

The 'new' posterior mean m_θ and θ are centred on the 'old' posterior mean $m_\theta^{\setminus i}$. By matching m_θ with $m_\theta^{\setminus i}$, θ is centred on m_θ. Furthermore, $m_\theta^{\setminus i}$ is centred on m_i, so the posterior can be refined iteratively through the approximate factor updates. Minka [64] suggested forcing negative v_i's to some very large value (10^8) and setting $v_\theta = v_\theta^{\setminus i}$ to improve canonical EP convergence and called this 'restricted' EP. He reported that 'restricted' EP leads to inaccurate posteriors however often it is necessary for convergence, shown in Figures 3.5 (left). We find that convergence and accuracy with $\gamma \geq 0$ can be improved by using Minka's EP restrictions on every data point, which we will call 'restricted' γ-EP, shown in Figures 3.5 (right).

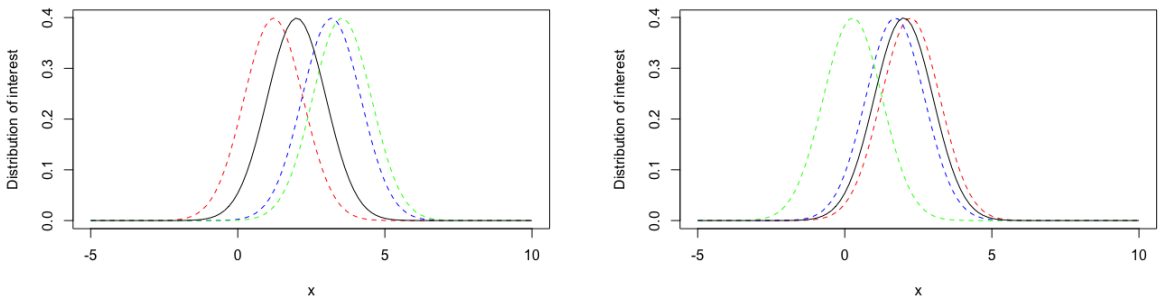


Figure 3.5: Distributions of interest for the same three samples (red, blue, green) approximating the true distribution (black) using γ-EP with $\gamma = -1$ (left) and $\gamma = 1$ (right).

The 'restricted' γ-EP algorithm for the clutter problem (with $a = 10, b = 100$).

1. Initialise: $v_i = \infty, m_i = 0, s_i = 1, m_\theta = 0, v_\theta = 100, \nabla_m = 0$.

2. Until (m_i, v_i) converges (change less than 10^{-4}).

For i = 1,...,n:

Compute 'old' posterior:

$$(v_\theta^{\setminus i})^{-1} = v_\theta^{-1} - v_i^{-1}$$

$$m_\theta^{\setminus i} = m_\theta + v_\theta^{\setminus i} v_i^{-1}(m_\theta - m_i + \gamma v_i \nabla_m)$$

Update 'new' posterior:

$$Z_i = (1-w)\mathcal{N}(x_i | m_\theta^{\setminus i}, (v_\theta^{\setminus i} + 1)) + w\mathcal{N}(x_i | 0, 10)$$

$$r_i = 1 - \frac{1}{Z_i} w\mathcal{N}(x_i | 0, 10)$$

$$m_\theta = m_\theta^{\setminus i} + v_\theta^{\setminus i} \nabla_m$$

$$v_\theta = v_\theta^{\setminus i} - \frac{1}{d}(v_\theta^{\setminus i})^2 \Gamma$$

Update approximate factor:

$$m_i = m_\theta$$

$$v_i = v_\theta^{-1} - (v_\theta^{\setminus i})^{-1}$$

if $\gamma \geq 0$

$$v_\theta = v_\theta^{\setminus i} \text{ and } v_i = 10^8$$

if $v_i < 0$

$$v_\theta = v_\theta^{\setminus i} \text{ and } v_i = 10^8$$

$$s_i = \frac{Z_i}{(2\pi v_i)^{d/2} \mathcal{N}(m_i | m_\theta^{\setminus i} + \gamma v_i \nabla_m, (v_i + v_\theta^{\setminus i})\boldsymbol{I})}$$

Compute the evidence:

$$B = \frac{m_\theta^T m_\theta}{v_\theta} - \sum_i \frac{m_i^T m_i}{v_i}$$

$$p(D) \approx (2\pi v_x)^{d/2} \exp(B/2) \prod_{i=1}^{n} \left(s_i (2\pi v_i)^{-d/2} \right)$$

where $\nabla_m = r_i \frac{x_i - m_\theta^{\setminus i}}{v_\theta^{\setminus i} + 1}$, $\nabla_v = -\frac{r_i d}{2(v_\theta^{\setminus i} + 1)} + \frac{r_i (x_i - m_\theta^{\setminus i})^T (x_i - m_\theta^{\setminus i})}{2(v_\theta^{\setminus i} + 1)^2}$,

$\Gamma = \nabla_m^T \nabla_m - 2\nabla_v = \frac{r_i d}{(v_\theta^{\setminus i} + 1)} - r_i(1 - r_i) \frac{(x_i - m_\theta^{\setminus i})^T (x_i - m_\theta^{\setminus i})}{(v_\theta^{\setminus i} + 1)^2}$

3.3.1 Bayesian Model Comparison

By inspecting the EP fixed-point, we can solve for s_i to get the update including the bias term $v_i \nabla_m$:

$$\int \tilde{t}_i(\boldsymbol{\theta}) \mathcal{N}(\boldsymbol{\theta}|\mathbf{m}_\theta^{\setminus i}, v_\theta^{\setminus i})\, d\boldsymbol{\theta} = \int t_i(\boldsymbol{\theta}) \mathcal{N}(\boldsymbol{\theta}|\mathbf{m}_\theta^{\setminus i}, v_\theta^{\setminus i})\, d\boldsymbol{\theta} \tag{3.43}$$

$$s_i = \frac{Z_i}{(2\pi v_i)^{d/2} \mathcal{N}(\mathbf{m}_i|\mathbf{m}_\theta^{\setminus i} + \gamma v_i \nabla_m, (v_i + v_\theta^{\setminus i})\boldsymbol{I})} \tag{3.44}$$

We can then compute the evidence as in canonical EP [5]:

$$P(D) \approx \int \prod_i^n \tilde{t}_i(\boldsymbol{\theta})\, d\boldsymbol{\theta} = (2\pi v_\theta)^{d/2} \exp(B/2) \prod_i^n \left(s_i (2\pi v_i)^{-d/2} \right) \tag{3.45}$$

where $B = \mathbf{m}_\theta^T \mathbf{m}_\theta / v_\theta - \sum_i \left(\mathbf{m}_i^T \mathbf{m}_i / v_i \right)$. Table 3.1 compares the average posterior mean m_θ given by 'restricted' ADF (with $\gamma = -1$), 'restricted' EP[3] and 'restricted' γ-EP[4] ($\gamma^* \in [-1, 1]$ was chosen using Brent's method [9] to maximise the evidence (3.45)) for $w \in \{0.2, 0.4, 0.6, 0.8\}$ over 50 samples of $n = 20$ data points where $\theta = 2$. Except for $w = 0.4$, 'restricted' γ-EP outperforms 'restricted' ADF on average. However, we tested the statistical significance of these results with a Wilcoxon paired signed rank test and found no statistical significance at the 1% significance level. We repeated the experiment with $n = 200$, shown in Table 3.2. Although, only $w = 0.4$ showed a statistically significant improvement between 'restricted' γ-EP and 'restricted' ADF, 'restricted' γ-EP has the lowest average error for all clutter levels. However, the standard deviations are too high for any of the other results to be significant. In particular, it is interesting to see that the maximum evidence value of γ is 1 for all clutter levels.

Table 3.1: Average m_θ ($\pm$ one standard deviation) for ADF, EP and γ-EP (with average maximum evidence γ) on 50 samples with $\theta = 2$ and $n = 20$ for various levels of background clutter.

w	0.2	0.4	0.6	0.8
ADF	1.511 ± 0.822	1.318 ± 2.164	0.195 ± 3.329	-0.601 ± 5.166
EP	1.555 ± 0.880	1.141 ± 2.234	0.448 ± 3.235	-0.377 ± 5.495
γ-EP	1.784 ± 1.106 $\bar{\gamma^*} = 0.164$	1.246 ± 2.100 $\bar{\gamma^*} = 0.357$	0.494 ± 3.725 $\bar{\gamma^*} = 0.147$	-0.534 ± 7.070 $\bar{\gamma^*} = -0.180$

[3] Canonical EP did not converge.

[4] The optimisation was improved by removing the if-statement for $\gamma \geq 0$.

Table 3.2: Average m_θ ($\pm$ one standard deviation) for ADF, EP and γ-EP (with average maximum evidence γ) on 50 samples with $\theta = 2$ and $n = 200$ for various levels of background clutter. Statistically significant results at the 1% level between γ-EP and ADF are shown in bold.

w	0.2	0.4	0.6	0.8
ADF	1.60 ± 0.281	0.877 ± 1.17	0.634 ± 2.05	0.524 ± 3.29
EP	1.62 ± 0.229	1.18 ± 1.12	0.519 ± 2.23	0.680 ± 3.69
γ-EP	1.97 ± 1.14	$\mathbf{1.75 \pm 1.64}$	1.47 ± 2.41	1.29 ± 3.64
	$\bar{\gamma}^* = 1$	$\bar{\gamma}^* = 1$	$\bar{\gamma}^* = 1$	$\bar{\gamma}^* = 1$

3.3.2 Basic Differential Multiplier Method

There is a marked increase in standard deviation between EP and γ-EP in Table 3.2 for $w = 0.2$. In this chapter, we will seek to constrain γ-EP to find a set of fixed-points with a smaller standard deviation. Notice, the similarity between the moment matching update for the Gaussian posterior mean and gradient ascent:

$$m_\theta = m_\theta^{\backslash i} + v_\theta^{\backslash i} \frac{\partial \log Z_i}{\partial m_\theta^{\backslash i}} \tag{3.46}$$

$$\dot{m}_\theta^{\backslash i} = \frac{\partial \log Z_i}{\partial m_\theta^{\backslash i}} = r_i \frac{x_i - m_\theta^{\backslash i}}{v_\theta^{\backslash i} + 1} \tag{3.47}$$

$$Z_i = \frac{\tilde{t}_i(\theta) q^{\backslash i}(\theta)}{q(\theta)} \tag{3.48}$$

The 'old' posterior mean moves in the direction of steepest ascent of $\log Z_i$ for every data point i, at a rate equal to the 'old' posterior variance, to give the 'new' posterior mean. Including the bias term in (3.46) yields:

$$m_\theta = (m_\theta^{\backslash i})^{EP} + v_\theta^{\backslash i} \left(\frac{\partial \log Z_i}{\partial m_\theta^{\backslash i}} + \lambda_i \gamma \frac{\partial \log Z_i^{old}}{\partial m_\theta^{\backslash i}} \right) \tag{3.49}$$

$$\dot{m}_\theta^{\backslash i} = \frac{\partial \log Z_i}{\partial m_\theta^{\backslash i}} + \lambda_i \gamma \frac{\partial \log Z_i^{old}}{\partial m_\theta^{\backslash i}} \tag{3.50}$$

where $(m_\theta^{\backslash i})^{EP}$ is the canonical EP update (3.25) and Z_i^{old} is the normalising constant from the previous iteration and λ_i is the Lagrange multiplier for the ith constraint $\log Z_i^{old} \leq 0$ for $i \in \{1, ..., n\}$ which implies $Z_i^{old} \leq 1$. The gradient ascent (3.49) is a solution to the *n constrained* optimisation problems:

$$\text{maximise } f_i(m_\theta^{\backslash i}) = \left(\log Z_i + \lambda_i \gamma \log Z_i^{old} \right) \tag{3.51}$$

$$\text{subject to } \log Z_i^{old} \leq 0 \tag{3.52}$$

To update the value of λ_i, an auxiliary differential equation is required:

$$\dot{\lambda}_i = \frac{\partial f_i}{\partial m_\theta^{\setminus i}} = \gamma \log Z_i^{old} \tag{3.53}$$

which performs gradient ascent on λ_i when $\gamma > 0$, though using the same sign for γ in (3.51) and (3.53) has been shown for general constrained optimisation problems to not work well because it tends to get stuck in saddle points [83]. It is also crucial to change the sign of the Lagrange multiplier in (3.51) because $\log Z_i$ will only be at a maximum if its gradient is oriented towards the constrained region $\log Z_i^{old} < 0$ so that $\partial \log Z_i = \lambda_i \partial \log Z_i^{old} \leq 0$. Expectation Propagation provides a simple solution to this problem by setting $\gamma = -1$ and $\lambda_i = 1$. When λ_i is not constant, the EP fixed-point does not change for any $\gamma < 0$ but it will take longer to converge as γ is decreased (and negative) [83].

There are two types of solutions according to whether the fixed-point lies in the region $\log Z_i^{old} < 0$ or on the boundary $\log Z_i^{old} = 0$ [5]. In the first case, the constraint is *inactive*, so $\lambda_i = 0$ and the stationary point is at $\partial \log Z_i = 0$ which implies $m_\theta^{\setminus i} = x_i$. This case is equivalent to γ-EP with $\gamma = 0$ and the posterior mean is equivalent to the sample mean at convergence. In the latter case, the constraint is *active*, so $\lambda \neq 0$ and the stationary point is at $\log Z_i^{old} = 0$ which implies $Z_i^{old} = 1$. Therefore, the solution of the optimisation problems (3.51) subject to (3.52) will have to satisfy the Karush-Kuhn-Tucker (KKT) [49] [52] conditions: $\log Z_i^{old} \leq 0, \lambda_i \geq 0, \lambda_i \log Z_i^{old} = 0$ for $i \in \{1, ..., n\}$.

The iterations of γ-EP may not converge with a poorly chosen positive value of γ in (3.51), but can be made to converge by maximising the evidence. The iterations move in the opposite direction to maximising $\log Z_i$, due to the positive sign of γ, but by choosing the value of γ to maximise the evidence, the iterations can be forced into a neighbourhood which maximises the objective (3.51) (because the log evidence is a function of $\sum_i \log Z_i$), which means on the next iteration the constraint $\log Z_i^{old}$ will be close to zero and the objective resembles canonical EP (with a small additive bias term).

Table 3.2 shows that the local maxima found by γ-EP with $\gamma = 1$ are consistently better than EP and ADF on average. Although the standard deviations are too large for the results to be statistically significant. In order to reduce the variance in posterior approximations, we can also optimise the value of λ_i (rather than setting $\lambda_i = 1$ in γ-EP). By swapping the sign in (3.53), we get a gradient *descent* update for λ_i which resembles the

Basic Differential Multiplier Method (BDMM) [83]. We will call this method $\lambda\gamma$-EP.

We performed a similar comparison of 'restricted' $\lambda\gamma$-EP, 'restricted' ADF and 'restricted' EP with $n = 200$ data points in Table 3.3. We restricted the comparison to low clutter levels $w \leq 0.25$ because in higher clutter levels, maximising the evidence failed to force the iterations towards any local maxima. In fact, the iterations were directed to local minima which catastrophically affected the approximations. The standard deviation for $\lambda\gamma$-EP with $w = 0.2$ is significantly lower than γ-EP in Table 3.2 and the value of $\bar{\gamma}^*$ decreases as the value of w increases. Again, we tested the statistical significance of these results with a Wilcoxon paired signed rank test and found $\lambda\gamma$-EP gives a statistically significant improvement over ADF and EP for $w \in \{0.1, 0.15, 0.2\}$.

Table 3.3: Average m_θ ($\pm$ one standard deviation) for ADF, EP and $\lambda\gamma$-EP on 50 samples with $\theta = 2$ and $n = 200$ in low clutter levels. Statistically significant results at the 1% level between γ-EP and ADF are shown in bold.

w	0.1	0.15	0.2	0.25
ADF	1.80 ± 0.095	1.70 ± 0.153	1.60 ± 0.281	1.48 ± 0.386
EP	1.80 ± 0.094	1.70 ± 0.144	1.62 ± 0.229	1.54 ± 0.322
$\lambda\gamma$-EP	$\mathbf{2.12 \pm 0.389}$	$\mathbf{2.08 \pm 0.450}$	$\mathbf{1.92 \pm 0.454}$	1.63 ± 0.396
	$\bar{\gamma}^* = 0.734$	$\bar{\gamma}^* = 0.669$	$\bar{\gamma}^* = 0.575$	$\bar{\gamma}^* = 0.525$

3.4 Concluding Remarks

In this chapter, we presented the γ-EP modification to canonical EP and its application on the clutter problem. We did not find that EP provides a statistically significant improvement over ADF on average. We did find that the local maxima achieved by maximising the evidence to select γ are superior to those found by canonical EP on average for the examples considered. We developed an extension to γ-EP using Lagrange multipliers which achieved statistically significant improvements over ADF and EP in low clutter levels. Here, we only looked at the one dimensional clutter problem. The γ-EP approach can be extended to d-dimensional $\boldsymbol{\theta}$ by using a d-dimensional $\boldsymbol{\gamma}$ and using a multi-dimensional optimisation method to maximise the evidence. The clutter problem is only one possible application of Expectation Propagation, which is a general tool for approximating posteriors with exponential family distributions. A thorough analysis of γ-EP will require more applications on different likelihood functions.

CHAPTER 4

BAYES POINT MACHINES AND ONCOGENIC SINGLE NUCLEOTIDE VARIANTS

"A failure of Bayesian prediction is an opportunity to learn."

— Edwin. T. Jaynes

This chapter applies the γ-EP algorithm to binary classification. The classifier, called a Bayes Point Machine (BPM), is an approximation to the Bayes optimal classifier [40] with a single average classifier. As in the previous chapter, we show $\gamma = -1$ recovers canonical EP. When $\gamma > 0$, the number of examples is reduced to a set of informative support vectors. Furthermore, we apply the BPM to a challenging task of classifying oncogenic (cancer causing) single nucleotide variants (SNVs) using a heterogeneous set of genomic features.

4.1 Bayes Point Machines

In this chapter, we consider the supervised learning problem. The goal of supervised learning is to learn a mapping from data to a target. Specifically, we consider binary classification for which the target is $y_i = \pm 1$. The Bayesian approaches to linear classification are competitive with the popular Support Vector Machine (SVM) [18]. The advantages of the SVM are its speed and sparsity. The SVM reduces the full training data (also called examples) to a smaller number of support vectors without sacrificing the quality of the classification output. The advantage of the Bayesian approach, called the *Bayes point*, is that it has been proven to be optimal [120] but computing the Bayes point exactly is in-

tractable as it requires solving an integral for every test input. Expectation Propagation (EP) can be used to efficiently compute the Bayes point. However, it cannot be used to remove redundant or noisy examples which can lead to EP failing spectacularly when the posterior puts most of its probability mass on examples which are clearly redundant (such as repeated examples). In this chapter, we present the γ-EP algorithm for training Bayes point machines, which is capable of learning sparse linear classifiers and can be extended to non-linear classification.

Given a set of d-dimensional independent & identically distributed training data and targets $D = \{(\mathbf{x_1}, y_1), ..., (\mathbf{x_n}, y_n)\}$, a linear classifier learns a d-dimensional vector $\mathbf{w}$ and a scalar offset b to classify each data point $\mathbf{x}_i$ using $\tilde{y} = \text{sign}(\mathbf{w}^T\mathbf{x}_i + b)$. We often incorporate the offset into the weight vector $\mathbf{w} = (b, w_1, w_2, ..., w_d)$, giving a simplified decision function $\tilde{y} = \text{sign}(\mathbf{w}^T\mathbf{x}_i)$, by appending a 1 to each data vector $\mathbf{x}_i = (1, x_{1,i}, x_{2,i}, ..., x_{d,i})$.

Model comparison is central to supervised learning, and classification is no exception. That is because for any particular mapping from training data to predicted class labels, there are infinitely many possible values of $\mathbf{w}$. Therefore, model selection criteria are necessary to sort through the infinite set of classifiers and select the one which will generalise best on out-of-sample test data.

In chapter 2, we introduced two complimentary model selection criteria, VC dimension and the evidence, and we showed that maximising either is optimal. The Bayesian generalisation error bound (2.10) can be rewritten in terms of the maximum margin generalisation error (2.11):

$$R[Bayes(\mathbf{x}^*)] \leq R[h] - \frac{\ln(4)}{n} \tag{4.1}$$

Thus, the improvement gained by maximising the evidence diminishes with increasing surface volume of version space [40]. The *Bayes optimal classifier* or *optimal perceptron* averages all classifiers in version space weighted by their posterior probabilities and for a new data point $\mathbf{x}^*$ it is [40] [64]:

$$Bayes(\mathbf{x}^*) = \text{sign} \int p(\mathbf{w}|y) p(\tilde{y}|\mathbf{x}^*, \mathbf{w})\, d\mathbf{w} = \text{sign}\left(E\left[\text{sign}(\mathbf{w}^T\mathbf{x}^*)\right]\right) \tag{4.2}$$

We can approximate the Bayes optimal classifier with the single average classifier by interchanging sign and expectation in (4.2) [40]:

$$\text{sign}\left(E\left[\text{sign}\left(\mathbf{w}^T\mathbf{x}^*\right)\right]\right) = \text{sign}\left(E[\mathbf{w}]^T\mathbf{x}^*\right) \tag{4.3}$$

The idea is that if version space is almost point-symmetric with respect to $E[\mathbf{w}]$, which is called the Bayes Point [64], then for each $\mathbf{w}$ in version space there exists another weight vector $\tilde{\mathbf{w}} = 2E[\mathbf{w}] - \mathbf{w}$, also in version space, so $\operatorname{sign}\left(\mathbf{w}^T\mathbf{x}^*\right) + \operatorname{sign}\left(\tilde{\mathbf{w}}^T\mathbf{x}^*\right) = 2\operatorname{sign}\left(E[\mathbf{w}]^T\mathbf{x}^*\right)$ [40].

The likelihood for Bayes point classification with additive noise $p(\xi) = \mathcal{N}(\xi|0,v)$ is:

$$p(\mathbf{y}|\mathbf{w},\xi) = \prod_{i=1}^{n} \Theta(y_i(\mathbf{w}^T\mathbf{x}_i + \xi)) = \prod_{i=1}^{n} \begin{cases} 1 & y_i(\mathbf{w}^T\mathbf{x}_i + \xi) > 0 \\ 0 & y_i(\mathbf{w}^T\mathbf{x}_i + \xi) < 0 \end{cases} \tag{4.4}$$

The noise can be averaged out by marginalisation which amounts to modifying the likelihood to a *probit regression* [77]:

$$p(\mathbf{y}|\mathbf{w}) = \prod_{i=1}^{n} \int p(\mathbf{y}|\mathbf{w},\xi)p(\xi)d\xi \tag{4.5}$$

$$= \prod_{i=1}^{n} \Phi\left(\frac{y_i\mathbf{w}^T\mathbf{x}_i}{\epsilon}\right) \tag{4.6}$$

$$Z_i = \Phi(z) = \int_{-\infty}^{z} \mathcal{N}(z|0,1)\,dz \tag{4.7}$$

where $Z_i = \int t_i(\mathbf{w})q^{\setminus i}(\mathbf{w})d\mathbf{w}$ is the ith normalising constant after removing the ith approximate factor, $\Phi(z)$ is the error-function and ϵ is the noise variance which can be chosen either by maximising the evidence or cross-validation. It is convenient to assume a spherical Gaussian prior:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0},\mathbf{I}) \tag{4.8}$$

We will use a full covariance Gaussian to give the best possible approximations to the posterior and as there is no matrix inversion required, computing the Bayes Point takes $O(nd^2)$ time (excluding computing the evidence which takes $O(n^3)$ time).

In section 4.2, we introduce and derive the γ-EP algorithm for training Bayes point machines (BPM). In section 4.2.1, we show that setting $\gamma > 0$ modifies the BPM loss function (similar to the 'hinge' loss used in the SVM) and causes the influence of redundant examples to become very small. In section 4.2.2 we extend the γ-EP BPM to non-linear classification. We demonstrate the effectiveness of γ-EP against canonical EP and the SVM on several benchmark classification datasets. In section 4.3 we combine the BPM with a composite kernel using several heterogeneous data sources to classify single nucleotide variants as oncogenic (cancer causing) or benign.

The γ-EP algorithm for the Bayes Point Machine.

To simplify notation we will write $y_i \mathbf{x}_i / \epsilon$ as $\mathbf{x}_i$.

1. Initialise $v_i = \infty, m_i = 0, s_i = 1, \mathbf{m}_w = \mathbf{0}, \mathbf{V}_w = \mathbf{I}, \alpha_i = 0$.

2. Until (m_i, v_i) converges (change less than 10^{-4}).

For i = 1,...,n:

Compute 'old' posterior:

$$\mathbf{V}_w^{\setminus i} = \mathbf{V}_w + \frac{(\mathbf{V}_w \mathbf{x}_i)(\mathbf{V}_w \mathbf{x}_i)^T}{v_i - \mathbf{x}_i^T \mathbf{V}_w \mathbf{x}_i}$$

$$\mathbf{m}_w^{\setminus i} = \mathbf{m}_w + (\mathbf{V}_w^{\setminus i} \mathbf{x}_i) v_i^{-1} (\mathbf{x}_i^T \mathbf{m}_w - m_i + \gamma v_i \alpha_i)$$

Compute 'new' posterior:

$$z_i = \frac{\mathbf{x}_i^T \mathbf{m}_w^{\setminus i}}{\sqrt{\mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i + 1}}$$

$$\alpha_i = \frac{1}{\sqrt{\mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i + 1}} \frac{\mathcal{N}(z_i | 0, 1)}{\Phi(z_i)}$$

$$\mathbf{m}_w = \mathbf{m}_w^{\setminus i} + \mathbf{V}_w^{\setminus i} \alpha_i \mathbf{x}_i$$

$$\begin{aligned} \mathbf{V}_w &= \mathbf{V}_w^{\setminus i} - \mathbf{V}_w^{\setminus i} \left(\nabla_m \nabla_m^T - 2 \nabla_v \right) \mathbf{V}_w^{\setminus i} \\ &= \mathbf{V}_w^{\setminus i} - (\mathbf{V}_w^{\setminus i} \mathbf{x}_i) \, \Gamma \, (\mathbf{V}_w^{\setminus i} \mathbf{x}_i)^T \end{aligned}$$

Update approximate factor:

$$v_i = \Gamma^{-1} - \mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i$$

$$m_i = \mathbf{x}_i^T \mathbf{m}_w^{\setminus i} + \mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i \alpha_i = \mathbf{x}_i^T \mathbf{m}_w$$

$$s_i = \frac{\Phi(z_i) \sqrt{1 + v_i^{-1} \mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i}}{\exp\left(-\frac{1}{2} \alpha_i^2 (\mathbf{x}_i^T \mathbf{V}_\mathbf{w}^{\setminus \mathbf{i}} \mathbf{x}_i - \gamma v_i)(\mathbf{x}_i^T \mathbf{V}_\mathbf{w}^{\setminus \mathbf{i}} \mathbf{x}_i + v_i)^{-1} (\mathbf{x}_i^T \mathbf{V}_\mathbf{w}^{\setminus \mathbf{i}} \mathbf{x}_i - \gamma v_i) \right)}$$

Compute the evidence:

$$B = \mathbf{m}_w^T \mathbf{V}_w^{-1} \mathbf{m}_w - \sum_i \frac{m_i^2}{v_i}$$

$$p(D) \approx |\mathbf{V}_w|^{\frac{1}{2}} \exp(B/2) \prod_i s_i$$

where $\nabla_m = \frac{\partial \log Z_i}{\partial \mathbf{m}_w^{\setminus i}} = \alpha_i \mathbf{x}_i$, $\nabla_v = \frac{\partial \log Z_i}{\partial \mathbf{V}_w^{\setminus i}} = -\frac{1}{2} \frac{\alpha_i \mathbf{x}_i^T \mathbf{m}_w^{\setminus i}}{\mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i + 1} \mathbf{x}_i \mathbf{x}_i^T$,

$\Gamma = \frac{\alpha_i \left((\mathbf{m}_w^{\setminus i})^T \mathbf{x}_i + \alpha_i (\mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i + 1) \right)}{\mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i + 1} = \frac{\alpha_i (\mathbf{x}_i^T \mathbf{m}_w + \alpha_i)}{\mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i + 1} = \frac{\alpha_i (m_i + \alpha_i)}{\mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i + 1}$

4.2 Extensions of Gaussian Expectation Propagation

Each iteration of Expectation Propagation produces an estimate of the leave-one-out error without any extra computation [87]. By using the 'old' posterior mean $\mathbf{m}_w^{\setminus i}$ to approximate a classifier trained on $(n-1)$ data points, we can predict a class label for the i^{th} data point and estimate the leave-one-out error E_{error}^{n-1}:

$$E_{error}^{n-1} = 1 - \frac{1}{n}\sum_{i=1}^{n} \Theta\left(y_i \mathbf{m}_w^{\setminus i} \mathbf{x}_i\right) \tag{4.9}$$

where $\Theta(z)$ is the step function defined by (4.4). We can use E_{error}^{n-1} for model selection. Qi et al. [87] tested this idea and found E_{error}^{n-1} to be a better model selection criterion than others such as: evidence, feature sparsity and the margin. This is not surprising as the leave-one-out error is an 'almost' unbiased estimate of the test error [57] [118] (it is only 'almost' unbiased because the sample size is n-1 instead of n). However, it is surprising that they found maximising the margin to be misleading as E_{error}^{n-1} and the margin ρ are intimately related by the following inequality [118]:

$$E_{error}^{n-1} \leq E\left(\frac{SD}{n\rho^2}\right) \tag{4.10}$$

where D is the smallest sphere containing the training data and S is a quantity called the 'span' of the support vectors [118]. If we take $\{S, D, n\}$ as constant, then minimising E_{error}^{n-1} is equivalent to maximising ρ. In fact, Qi et al. even found the margin to decrease as the evidence increased, which then led to the BPM overfitting. The optimal number of EP iterations was found not to be correlated with maximising the margin or evidence. They took this as proof that maximising the margin and evidence were not suitable model selection criteria. However, this can also be understood as reflecting an error in the underlying model assumptions. In chapter 2, we stated that when the evidence is not correlated with the generalisation error, we should return to our modelling assumptions and correct this failure to find a better model [58].

The BPM trained using EP seems to not automatically control complexity, unlike the SVM which does so by maximising the margin. After all, we are assuming a fixed hypothesis, which amounts to the prior $p(\mathbf{w})$ and the noise variance ϵ, and canonical EP does not perform Bayesian model comparison. Given a sufficiently small value for ϵ, we would expect the EP iterations to eventually achieve zero training error (if possible with the set of classifiers considered) even if that would worsen generalisation performance. And yet, EP and related methods [74] achieve state-of-the-art generalisation performance for

deterministic Bayesian binary classification without overfitting the training data.

However, when ϵ is set too high, EP will disastrously underfit. To remedy this we introduce the γ-EP modification for Bayes point machines. Once again, we start with a 'reuse' update for the approximate factor mean using the 'new' posterior mean:

$$m_i^{reuse} = \mathbf{x}_i^T \mathbf{m}_w^{\setminus i} + (\mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i)\alpha_i \tag{4.11}$$

$$= \mathbf{x}_i^T \mathbf{m}_w \tag{4.12}$$

Following similar lines as chapter 3 but with a full covariance matrix, we plug in the previous EP updates until we have the 'old' posterior mean in terms of the 'reuse' update:

$$\begin{aligned}
(\mathbf{m}_w^{\setminus i})^{new} &= \mathbf{m}_w^{new} + (\mathbf{V}_w^{\setminus i}\mathbf{x}_i)v_i^{-1}(\mathbf{x}_i^T\mathbf{m}_w^{new} - m_i) \\
&= \mathbf{m}_w^{new} + (\mathbf{V}_w^{\setminus i}\mathbf{x}_i)v_i^{-1}(\mathbf{x}_i^T\mathbf{m}_w^{new} - (\mathbf{x}_i^T\mathbf{m}_w^{\setminus i} + (v_i + \mathbf{x}_i^T\mathbf{V}_w^{\setminus i}\mathbf{x}_i)\alpha_i)) \\
&= \mathbf{m}_w^{new} + (\mathbf{V}_w^{\setminus i}\mathbf{x}_i)v_i^{-1}(\mathbf{x}_i^T\mathbf{m}_w^{new} - \mathbf{x}_i^T\mathbf{m}_w^{old} - v_i\alpha_i) \\
&= \mathbf{m}_w^{new} + (\mathbf{V}_w^{\setminus i}\mathbf{x}_i)v_i^{-1}(\mathbf{x}_i^T\mathbf{m}_w^{new} - m_i^{reuse} - v_i\alpha_i) \qquad (4.13)
\end{aligned}$$

Therefore, EP with (4.12) and (4.13) is equivalent to canonical EP. This new expression (4.13) incorporates a bias term $v_i\alpha_i$ into the 'old' posterior mean. Whenever we include the bias term, m_i^{reuse} will be shortened to m_i.

In order to derive (4.13) we have to remove $\tilde{t}_i$ from the posterior to get an 'old' posterior, where $\tilde{t}_i$ is given by

$$\tilde{t}_i = s_i \exp\left(-\frac{1}{2v_i}\left(\mathbf{w}^T\mathbf{x}_i - m_i + \gamma v_i\alpha_i\right)^2\right) \tag{4.14}$$

and γ is the coefficient of the bias term $v_i\alpha_i$, so γ-EP with $\gamma = -1$ is equivalent to canonical EP. By using Bayes' Theorem with the approximate factor as the likelihood and the 'old' posterior as the prior, we get an equation for the 'new' posterior. Rearranging this equation, we get expressions for $(\mathbf{m}_w^{\setminus i}, \mathbf{V}_w^{\setminus i})$:

$$(\mathbf{V}_w^{\setminus i})^{-1} = \mathbf{V}_w^{-1} - v_i^{-1}\mathbf{x}_i\mathbf{x}_i^T \tag{4.15}$$

$$\mathbf{m}_w^{\setminus i} = \mathbf{V}_w^{\setminus i}\left(\mathbf{x}_i v_i^{-1}(\gamma v_i\alpha_i - m_i) + \mathbf{V}_w^{-1}\mathbf{m}_w\right) \tag{4.16}$$

$$= \mathbf{m}_w + \mathbf{V}_w\mathbf{x}_i\left(v_i^{-1}(\mathbf{x}_i^T\mathbf{m}_w - m_i + \gamma v_i\alpha_i)\right) \tag{4.17}$$

The $\mathbf{V}_w^{\setminus i}$ update is the same as canonical EP. Rearranging (4.16), we can derive equations for the Gaussian natural parameter $\mathbf{V}_w^{-1}\mathbf{m}_w$ for canonical EP (4.18) and γ-EP (4.19):

$$\mathbf{V}_w^{-1}\mathbf{m}_w = (\mathbf{V}_w^{\setminus i})^{-1}\mathbf{m}_w^{\setminus i} + \mathbf{x}_i v_i^{-1} m_i \tag{4.18}$$

$$\mathbf{V}_w^{-1}\mathbf{m}_w = (\mathbf{V}_w^{\setminus i})^{-1}\mathbf{m}_w^{\setminus i} + \mathbf{x}_i v_i^{-1} m_i - \gamma\alpha_i\mathbf{x}_i \tag{4.19}$$

Now, we can derive $\mathbf{V}_w^{-1}\mathbf{m}_w$ from the moment matching updates using the Woodbury Identity [81] for canonical EP [41] and γ-EP:

$$\mathbf{V}_w^{-1} = (\mathbf{V}_w^{\setminus i})^{-1} + \mathbf{x}_i \left(\Gamma^{-1} - \mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i\right)^{-1} \mathbf{x}_i^T \tag{4.20}$$

$$\mathbf{V}_w^{-1}\mathbf{m}_w = (\mathbf{V}_w^{\setminus i})^{-1}\mathbf{m}_w^{\setminus i} + \mathbf{x}_i \left[\left(\Gamma^{-1} - \mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i\right)^{-1} \left(\alpha_i v_i + \mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i \alpha_i + \mathbf{x}_i^T \mathbf{m}_w^{\setminus i}\right)\right] \tag{4.21}$$

$$= (\mathbf{V}_w^{\setminus i})^{-1}\mathbf{m}_w^{\setminus i} + \mathbf{x}_i \left[\left(\Gamma^{-1} - \mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i\right)^{-1} \left(\mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i \alpha_i + \mathbf{x}_i^T \mathbf{m}_w^{\setminus i}\right)\right] + \alpha_i \mathbf{x}_i \tag{4.22}$$

By comparing (4.21) and (4.18), we get the fully factorised canonical EP approximate factor mean and variance updates. By comparing (4.22) and (4.19), we see canonical EP is equivalent to γ-EP with $\gamma = -1$. But with $\gamma = 0$, there is an additive error term $\nabla_m = \alpha_i \mathbf{x}_i$ and with $\gamma = 1$ the error term is $2\nabla_m$. If the α_i's are initialised to 0, the first iteration of γ-EP is the same irrespective of the value γ and is equivalent to ADF. The error term scales each data point by a particular error function given by [90]:

$$z_i = \frac{\mathbf{x}_i^T \mathbf{m}_w^{\setminus i}}{\sqrt{\mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i + 1}} \tag{4.23}$$

$$\alpha_i = \frac{1}{\sqrt{\mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i + 1}} \frac{\mathcal{N}(z_i|0,1)}{\Phi(z_i)} \tag{4.24}$$

By inspecting the EP fixed-point, we can solve for s_i to get the update for γ-EP:

$$\int \tilde{t}_i(\mathbf{w}) \mathcal{N}(\mathbf{w}|\mathbf{m}_w^{\setminus i}, \mathbf{V}_w^{\setminus i})\, d\mathbf{w} = \int t_i(\mathbf{w}) \mathcal{N}(\mathbf{w}|\mathbf{m}_w^{\setminus i}, \mathbf{V}_w^{\setminus i})\, d\mathbf{w} \tag{4.25}$$

$$s_i = \frac{\Phi(z_i)\sqrt{1 + v_i^{-1}\mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i}}{\exp\left(-\frac{1}{2}\alpha_i^2 (\mathbf{x}_i^T \mathbf{V}_{\mathbf{w}}^{\setminus \mathbf{i}} \mathbf{x}_i - \gamma v_i)(\mathbf{x}_i^T \mathbf{V}_{\mathbf{w}}^{\setminus \mathbf{i}} \mathbf{x}_i + v_i)^{-1}(\mathbf{x}_i^T \mathbf{V}_{\mathbf{w}}^{\setminus \mathbf{i}} \mathbf{x}_i - \gamma v_i))\right)} \tag{4.26}$$

Setting $\gamma = -1$ recovers the simpler form used in canonical EP:

$$s_i = \frac{\Phi(z_i)\sqrt{1 + v_i^{-1}\mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i}}{\exp\left(-\frac{1}{2}\alpha_i^2 \Gamma^{-1}\right)} \tag{4.27}$$

We can then compute the evidence as in canonical EP [5]:

$$B = \mathbf{m}_w^T \mathbf{V}_w^{-1} \mathbf{m}_w - \sum_i \frac{m_i^2}{v_i} \tag{4.28}$$

$$p(D) \approx |\mathbf{V}_w|^{\frac{1}{2}} \exp(B/2) \prod_i s_i \tag{4.29}$$

The decision boundaries of the BPM and SVM on a balanced two-class two-dimensional toy dataset of 40 data points are shown in Figure 4.1. The true decision boundary is a vertical line at -0.55. Canonical EP (or γ-EP with $\gamma = -1$) can estimate the approximate posterior exactly. The posterior estimated by γ-EP with $\gamma = 1$ comes with an additive error term $2\alpha_i \mathbf{x}_i$, which causes the classifier to overfit. With large ϵ, (left), the overfitting is helpful as canonical EP underfits the data. By lowering the ϵ parameter, (right), both classifiers line up in the same 'corridor'. We compare the soft-margin SVM to the BPM with $\gamma = 1$ (bottom). Both classifiers achieve the same training error but the SVM fails to find the right 'corridor'.

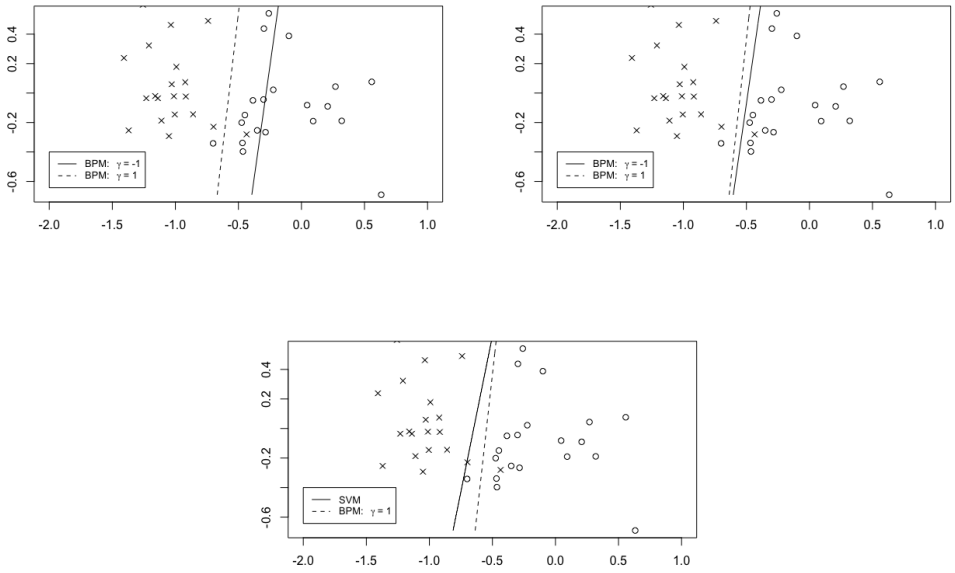


Figure 4.1: Bayes point machine with $\gamma = 1$ vs $\gamma = -1$ (top) or vs SVM (bottom) on a toy dataset both with $\epsilon = 5$ (left), $\epsilon = 2$ (right and bottom).

4.2.1 Sparse Bayes Point Machine

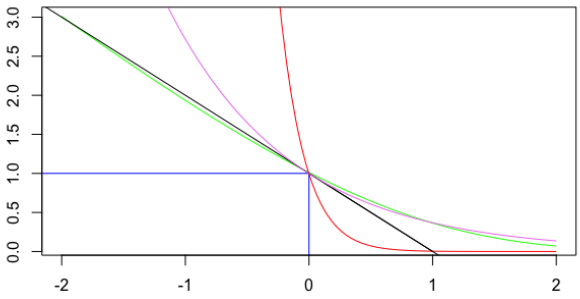


Figure 4.2: Plot of the rescaled $\frac{\mathcal{N}}{\Phi}$ (4.24) (green) and rescaled $\frac{\mathcal{N}}{\Phi}$ with $\gamma > 0$ (red) error functions, 'hinge' error (black), exponential error (violet) and '0-1' error (blue).

Figure 4.2 shows a variety of error functions including $\frac{\mathcal{N}}{\Phi}$ (green), $\frac{\mathcal{N}}{\Phi}$ with $\gamma > 0$ (red), 'hinge' (black), 'exponential' (violet) and '0-1' (blue). All errors are functions of $\tilde{z} = yf(\mathbf{x})$, where $f(\mathbf{x})$ is a linear discriminant function, and $\frac{\mathcal{N}}{\Phi}$ is rescaled to cross through the point (0,1). Sparsity can be obtained by removing examples below a threshold on the error function. Ideally, we only want to remove redundant examples, which carry no extra information given the rest of the data. The error for redundant examples will be approximately 0, depending on the function, so the threshold should be at most slightly greater than 0. Applying a threshold to the $\frac{\mathcal{N}}{\Phi}$ error only removes examples for which $\tilde{z} > 2$, if there are any at all. Whereas, with $\gamma > 0$ the same threshold includes examples closer to the separating hyperplane, which increases sparsity. This threshold defines an asymmetric 'soft-margin' (with data points allowed to be within the margin). The correctly classified examples within the margin of $\gamma > 0$ are penalised less strongly than the 'hinge' error and the penalty increases non-linearly with distance from the separating hyperplane. However, the negative values of $\tilde{z}$ are penalised far more strongly than even the exponential error which may lead to overfitting.

The linear classifier found by canonical EP can be catastrophically affected by repeated data points [63]. This is in sharp contrast to the SVM which is designed to use only non-redundant data points, the support vectors, and exclude those that are redundant. Applying a threshold to the γ-EP error function with $\gamma > 0$ defines a 'soft-margin', but the error function is defined in terms of the leave-one-out estimator $\mathbf{x}_i^T \mathbf{m}_w^{\setminus i}$ in (4.23). We

will show that sparsity is preserved for the full posterior estimator $\mathbf{x}_i^T \mathbf{m}_w$ also. To see this, consider the expression for $\mathbf{x}_i^T \mathbf{m}_w^{\setminus i}$:

$$\mathbf{x}_i^T \mathbf{m}_w^{\setminus i} = \mathbf{x}_i^T \mathbf{m}_w + \mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i v_i^{-1} \left(\mathbf{x}_i^T \mathbf{m}_w - m_i + \gamma v_i \alpha_i \right) \tag{4.30}$$

$$= \mathbf{x}_i^T \mathbf{m}_w + \mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i v_i^{-1} \left(\mathbf{x}_i^T \mathbf{m}_w - \mathbf{x}_i^T \mathbf{m}_w^{old} \right) + \mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i \gamma \alpha_i \tag{4.31}$$

$$= \mathbf{x}_i^T \mathbf{m}_w + \gamma \left(\mathbf{x}_i^T \mathbf{m}_w - \mathbf{x}_i^T (\mathbf{m}_w^{\setminus i})^{old} \right) + \delta \tag{4.32}$$

where $\delta = \mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i v_i^{-1} (\mathbf{x}_i^T \mathbf{m}_w - \mathbf{x}_i^T \mathbf{m}_w^{old})$. Plugging $\gamma = -1$ into (4.32) yields the trivial expression at convergence, $\mathbf{x}_i^T \mathbf{m}_w^{\setminus i} = \mathbf{x}_i^T (\mathbf{m}_w^{\setminus i})^{old}$, since $\delta \to 0$ when the algorithm converges. Plugging $\gamma = 1$ into (4.32) yields:

$$\mathbf{x}_i^T \mathbf{m}_w^{\setminus i} = 2\mathbf{x}_i^T \mathbf{m}_w - \mathbf{x}_i^T (\mathbf{m}_w^{\setminus i})^{old} + \delta \tag{4.33}$$

where $\mathbf{x}_i^T \mathbf{m}_w^{\setminus i} = \mathbf{x}_i^T \mathbf{m}_w$ at convergence, i.e. the classifier is robust to the removal of any data point. We can threshold the α_i's above a small value α_0 and consider those to be 'support vectors' and the rest irrelevant data points. The orthogonal distance from the ith data point to the separating hyperplane is $\epsilon y_i \frac{m_i}{||\mathbf{m}_w||^2}$. The irrelevant data points are the furthest from the separating hyperplane and will have large values of $m_i = \mathbf{x}_i^T \mathbf{m}_w$, so by (4.33) correct classifications will have $\alpha_i \approx 0$ and incorrect classifications will have $\alpha_i > \alpha_0$. The relevant data points are closer to the separating hyperplane and will have small values of $m_i = \mathbf{x}_i^T \mathbf{m}_w$, so by (4.33) $\mathbf{x}_i^T \mathbf{m}_w^{\setminus i}$ will be small and positive or negative and $\alpha_i > \alpha_0$.

Figure 4.3 demonstrates the effect of $\gamma > 0$ to produce sparsity on a balanced two-class two-dimensional toy dataset of 40 data points. The 'support vectors' for the BPM with $\gamma = 1$ and $\alpha_0 = 0.1$ are circled. (Top left) The BPM with $\gamma = 1$ has overfitted to the pattern in the centre whereas the canonical EP boundary is closer to the true decision boundary which is a vertical line at -0.55. (Top right) One data point is repeated 100 times (shown in bold). The BPM with $\gamma = 1$ is unaffected by the repeated data points whereas the canonical EP boundary is skewed onto the other side. (Bottom) One 'support vector' is repeated 100 times (shown in bold). The BPM with $\gamma = 1$ [1] is skewed slightly away from the repeated data points whereas the canonical EP boundary is skewed dramatically to place the majority of the data into the same class.

[1] Figure 4.3 (bottom right) The iterations became unstable and did not converge so we stopped γ-EP after 4 iterations. Alternatively, γ could be lowered to ensure convergence.

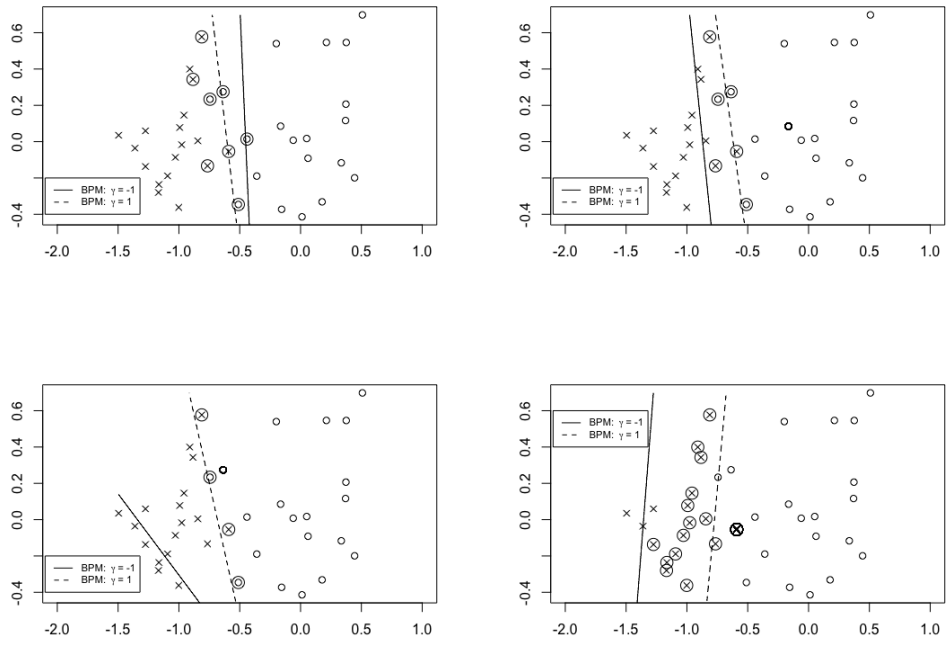


Figure 4.3: Bayes point machine with $\gamma = -1$ vs $\gamma = 1$ and $\epsilon = 2$ on a toy dataset both with and without (top left) repeated data points. The data points in bold are repeated 100 times. The support vectors for the BPM with $\gamma = 1$ and $\alpha_0 = 0.1$ are circled.

To assess the accuracy of the proposed method we used benchmark classification datasets from the UCI repository [23]. We demonstrate the performance of γ-EP for sparse linear classification against two popular algorithms, C-SVM [18] and ν-SVM [99], on the 'Sonar' ($n = 208$, $d = 60$) and 'Breast' ($n = 569$, $d = 30$) datasets. Each dataset was randomly split 50 times, we trained on 60% of the data, validated on 10% to select the SVM $C \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2\}$ and $\nu \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$ parameters and tested on the remaining 30%. The training data was normalised to have zero mean and unit variance. We ran γ-EP for 100 iterations or until convergence. The evidence was often infinite; so we used the validation set to select $\epsilon \in \{0.1, 1, 10\}$ and $\gamma \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ parameters in the first instance, and we used the evidence to give a more precise comparison of parameter values when available and the difference in validation errors was within 10^{-3}. We compared two thresholds α_0 on the error function, 10^{-3} and 10^{-4}. We call these models γ-EP1 and γ-EP2 respectively. To improve canonical EP in the presence of redundant data, we could train it using only the support vectors from an SVM. Minka stated in [64] that "this idea (of training canonical

Table 4.1: Test error rate and average number of support vectors (± one standard deviation) on the 'Sonar' (left) and 'Breast' (right) datasets.

Model	Error	# SVs	Error	# SVs
γ-EP1	0.248 ± 0.049	95.54 ± 15.09	0.035 ± 0.014	33.54 ± 9.99
γ-EP2	0.237 ± 0.048	102.16 ± 13.51	0.032 ± 0.012	39.00 ± 11.16
ν-SVM	0.241 ± 0.053	74.08 ± 1.64	0.024 ± 0.009	40.92 ± 1.29
C-SVM	0.243 ± 0.048	89.68 ± 3.11	0.024 ± 0.010	43.32 ± 3.07
EP + C-SV	0.231 ± 0.043	89.68 ± 3.11	0.039 ± 0.019	43.32 ± 3.07
EP + ν-SV	0.243 ± 0.046	74.08 ± 1.64	0.042 ± 0.015	40.92 ± 1.29
EP	0.229 ± 0.046	124.80 ± 0.00	0.027 ± 0.011	341.4 ± 0.00

EP on the support vectors only) has not been tested yet." We tried this with the C-SVM and ν-SVM and called it EP + C-SV and EP+ν-SVM respectively. The results are given in Table 4.1. As γ-EP is not a 'compression scheme', the Bayes point machines had to be retrained using only the support vectors. We used canonical EP (or γ-EP with $\gamma = -1$) to retrain the models as exact inference improved the validation error. γ-EP2 has a lower average test error than the SVM on the 'Sonar' dataset by using more support vectors on average. The SVM outperforms γ-EP on the 'Breast' dataset with a similar average number of support vectors. Canonical EP has a lower average test error than γ-EP on both datasets but uses all available training data. Canonical EP with the C-SVM support vectors performed surprisingly well, better than both γ-EP and the C-SVM on 'Sonar'. However, it required more compute to optimize the SVM C parameter as well as the EP ϵ parameter. It also performed worse on the 'Breast' dataset suggesting the centre of mass was further from the centre of the largest inscribable ball in version space [40]. Figure 4.4 shows that the number of irrelevant data points increases over the entire γ-EP trajectory on the full 'Sonar' dataset. The iterations are more unstable with high values of γ, reducing the chance of converging, but with more iterations the number of support vectors drops dramatically. The number of irrelevant data points for canonical EP remained zero until convergence.

Several extensions of ADF have been suggested to obtain sparse solutions. The *Informative Vector Machine* (IVM) [56] combines ADF with greedy forward selection using an entropy reduction heuristic up to a fixed maximum number of support vectors. Csató and Opper [21] developed a more flexible method which combines forward and backward selection for a fixed maximum number of support vectors. Expectation Propagation has been extended to use 'pseudo-inputs', which could be the centers of K-means clusters,

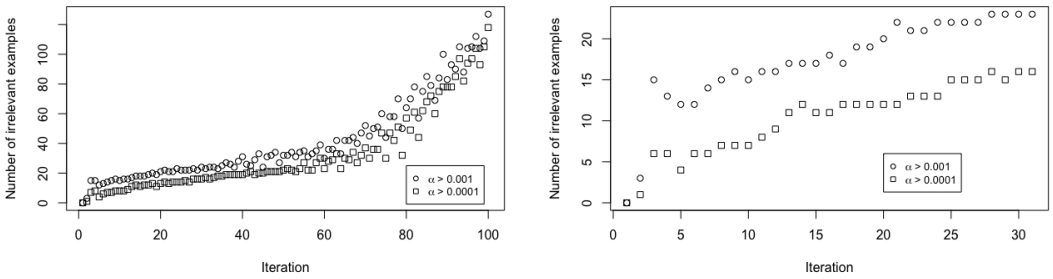


Figure 4.4: Number of irrelevant examples vs iterations for γ-EP with $\gamma = 0.8$ (left) and $\gamma = 0.75$ (right) and $\epsilon = 2$ on the 'Sonar' dataset.

by a method called *Sparse and Smooth Posterior Approximation* (SASPA) [86]. Two other state-of-the-art methods, *sparse pseudo-input Gaussian Process* (SPGP) [106] and *variable-sigma Gaussian Process* (VSGP)[2] [119], are special cases of SASPA [86]. It can be informative to know which examples are most important. Furthermore, none of these algorithms automatically learn the appropriate maximum number of support vectors for the particular dataset.

The *Relevance Vector Machine* (RVM) [112] is a Bayesian framework for automatically obtaining sparse solutions in linear regression and classification models. However, the RVM places independent Gaussian priors (also called *automatic relevance determination* (ARD)) over the feature weights so it can only produce sparsity in the features (with a radial basis function kernel, the 'features' do correspond to data points). Polson and Scott [85] developed a *Bayesian SVM* which can be trained with the usual tools of Gaussian linear models such as Expectation Maximisation and Markov Chain Monte Carlo algorithms. Both the SVM and Bayesian SVM maximise the margin which is equivalent to finding the centre of the largest inscribable ball in version space [40], which is different to the centre of mass (Bayes point). Recently, Uhrenholt, Charvet and Jensen [115] have proposed using a point process prior on inducing points to train sparse Gaussian Processes with stochastic variational inference. Interestingly, the evidence lower bound derived in [115] is trading-off complexity and capacity in terms of the number of inducing points drawn from the point process prior. This is similar in spirit to the Occam factor arguments described in chapter 2.

[2]VSGP was derived for regression only.

4.2.2 Kernel Bayes Point Machine

The Bayes point machine can also be extended to make use of the 'kernel trick' [7] [64]. Each weight vector can be written as a linear combination of basis functions $\phi(\mathbf{x}_i)$:

$$\mathbf{w} = \sum_i \alpha_i \phi(\mathbf{x}_i) \tag{4.34}$$

In this way we can rewrite the linear classifier from the d-dimensional $\mathbf{w}$ to the n-dimensional $\boldsymbol{\alpha}$, called the dual form. This linear classifier is then given by [40]:

$$\mathbf{w}^T \phi(\mathbf{x}) = \sum_i \alpha_i \phi(\mathbf{x})^T \phi(\mathbf{x}_i) = \alpha^T \mathbf{K} \tag{4.35}$$

where $\mathbf{K}$ is the n x n dimensional kernel matrix. The trick is that we do not have to explicitly state the form of the (possibly infinite dimensional) basis function ϕ, only the kernel matrix $\mathbf{K}$. A popular choice for the kernel matrix is the Gaussian kernel:

$$\mathbf{K}_{ij} = \exp\left(-\frac{1}{2\sigma^2}(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)\right) \tag{4.36}$$

For the kernel $\mathbf{K}_{ij}$, we have to separate $\mathbf{x}_i$ and y_i:

$$\mathbf{K}_{ij} = y_i y_j \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)$$

Before we outline the kernel BPM algorithm, we introduce some simplifying notation as used by Minka [64] and Opper & Winther [77] (we will use $\mathbf{x}_i$ when we mean $\phi(\mathbf{x}_i)$):

$$\lambda_i = \mathbf{x}_i^T \mathbf{V}_w^{\setminus i} \mathbf{x}_i,\ \mathbf{K}_{ij} = \mathbf{x}_i^T \mathbf{x}_j,\ \Lambda = \mathrm{diag}(v_1, ..., v_n),\ h_i^{\setminus i} = \mathbf{x}_i^T \mathbf{m}_w^{\setminus i},\ h_i = \mathbf{x}_i^T \mathbf{m}_w$$

The central relations of the kernel BPM algorithm are given by:

$$\mathbf{X}^T \mathbf{V}_w \mathbf{X} \approx \left(\mathbf{K}^{-1} + \Lambda^{-1}\right)^{-1} = \mathbf{A}_{ij} \tag{4.37}$$

$$\mathbf{m}_w = \mathbf{V}_w \sum_j \frac{\mathbf{x}_j (m_j - \gamma v_j \alpha_j)}{v_j} \tag{4.38}$$

We can derive (4.37) by approximating (4.15) and using the Kailath Variant of the Woodbury Identity [81]:

$$\mathbf{V}_w = \left((\mathbf{V}_w^{\setminus i})^{-1} + \mathbf{X}\Lambda^{-1}\mathbf{X}^T\right)^{-1} \tag{4.39}$$

$$\approx \left(\mathbf{I} + \mathbf{X}\Lambda^{-1}\mathbf{X}^T\right)^{-1} \tag{4.40}$$

$$(\mathbf{K} + \Lambda)^{-1} = \Lambda^{-1} - \Lambda^{-1}\mathbf{X}^T \left(\mathbf{I} + \mathbf{X}\Lambda^{-1}\mathbf{X}^T\right)^{-1} \mathbf{X}\Lambda^{-1} \tag{4.41}$$

$$\approx \Lambda^{-1} - \Lambda^{-1}\mathbf{X}^T \mathbf{V}_w \mathbf{X}\Lambda^{-1} \tag{4.42}$$

$$\mathbf{X}^T \mathbf{V}_w \mathbf{X} \approx \Lambda - \Lambda(\mathbf{K} + \Lambda)^{-1}\Lambda = \mathbf{A}_{ij} \tag{4.43}$$

From the expression for $\mathbf{V}_w^{\setminus i}$, we can derive an update for λ_i using the Woodbury Identity backwards and from the expression for $\mathbf{m}_w$ we can derive an update for h_i:

$$\lambda_i = \mathbf{x}_i^T \mathbf{V}_w \mathbf{x}_i \left(1 + \frac{\mathbf{x}_i^T \mathbf{V}_w \mathbf{x}_i}{v_i - \mathbf{x}_i^T \mathbf{V}_w \mathbf{x}_i}\right) \tag{4.44}$$

$$= \mathbf{x}_i^T \mathbf{V}_w \mathbf{x}_i \left(\frac{v_i}{v_i - \mathbf{x}_i \mathbf{V}_w^{\setminus i} \mathbf{x}_i}\right) \tag{4.45}$$

$$= \mathbf{x}_i^T \mathbf{V}_w \mathbf{x}_i + (\mathbf{x}_i^T \mathbf{V}_w \mathbf{x}_i)(\mathbf{I} - v_i^{-1} \mathbf{x}_i^T \mathbf{V}_w \mathbf{x}_i)^{-1} v_i^{-1} \mathbf{x}_i^T \mathbf{V}_w \mathbf{x}_i \tag{4.46}$$

$$= \left(\frac{1}{\mathbf{x}_i^T \mathbf{V}_w \mathbf{x}_i} - \frac{1}{v_i}\right)^{-1} \approx \left(\frac{1}{\mathbf{A}_{ii}} - \frac{1}{v_i}\right)^{-1} \tag{4.47}$$

$$h_i = \sum_j (\mathbf{x}_j^T \mathbf{V}_w \mathbf{x}_j) \frac{(m_j - \gamma v_j \alpha_j)}{v_j} \approx \sum_j \mathbf{A}_{ij} \frac{(m_j - \gamma v_j \alpha_j)}{v_j} \tag{4.48}$$

We can then show that canonical EP is equivalent to (4.48) with $\gamma = -1$ and $m_j^{reuse} = h_j$:

$$h_i \approx \sum_j \mathbf{A}_{ij} \frac{m_j}{v_j} \tag{4.49}$$

$$= \sum_j \mathbf{A}_{ij} \frac{h_j^{old}}{v_j} + \sum_j \mathbf{A}_{ij} \alpha_j \tag{4.50}$$

$$= \sum_j \mathbf{A}_{ij} \frac{(m_j^{reuse} + v_j \alpha_j)}{v_j} \tag{4.51}$$

In Expectation Propagation, it is also necessary to update h_i by moment matching:

$$h_i = h_i^{\setminus i} + \lambda_i \alpha_i \tag{4.52}$$

To classify a new data point, $\mathbf{x}^*$, we compute the sign of the decision function:

$$\tilde{y} = \text{sign}\left(\mathbf{m}_w^T \mathbf{x}^*\right) = \text{sign}\left(\sum_i y_i \alpha_i \mathbf{K}(\mathbf{x}^*, \mathbf{x}_i)\right) \tag{4.53}$$

The predictive distribution can be used to associate a probability measure with every classification. The resulting classifier is called a *Bayes machine* [45] or *Gaussian Process Classifier* [22]:

$$p(y^*|\mathbf{x}^*) \approx \int p(y^*|\mathbf{x}^*, \mathbf{w}) p(\mathbf{w}) d\mathbf{w} = \phi(z) \tag{4.54}$$

$$z = \frac{\mathbf{m}_w^T \mathbf{x}^*}{\sqrt{(\mathbf{x}^*)^T \mathbf{V}_w \mathbf{x}^*}} = \frac{\sum_i y_i \alpha_i \mathbf{K}(\mathbf{x}^*, \mathbf{x}_i)}{\sqrt{(\mathbf{K}(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}(\mathbf{x}^*, \mathbf{X}) \Lambda^{-1} (\Lambda - \mathbf{A}) \Lambda^{-1} \mathbf{K}(\mathbf{X}, \mathbf{x}^*)}} \tag{4.55}$$

The expression in the denominator of (4.55) can be derived using the Woodbury Identity with (4.40) and (4.42):

$$(\mathbf{x}^*)^T \mathbf{V}_w \mathbf{x}^* = (\mathbf{x}^*)^T \left(\mathbf{I} + \mathbf{X}\Lambda^{-1}\mathbf{X}^T\right)^{-1} \mathbf{x}^* \tag{4.56}$$

$$= \mathbf{K}(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}(\mathbf{x}^*, \mathbf{X})(\Lambda + \mathbf{K})^{-1}\mathbf{K}(\mathbf{X}, \mathbf{x}^*) \tag{4.57}$$

$$= \mathbf{K}(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}(\mathbf{x}^*, \mathbf{X})\Lambda^{-1}(\Lambda - \mathbf{A})\Lambda^{-1}\mathbf{K}(\mathbf{X}, \mathbf{x}^*) \tag{4.58}$$

Training the kernel BPM algorithm takes $O(n^4)$ time ($O(n^3)$ time to invert the kernel matrix for each data point) plus the time to compute $\mathbf{K}$ and testing takes $O(n^2)$ per new data point $\mathbf{x}^*$. Minka showed it is possible to reduce the training time to $O(n^3)$ by updating $\mathbf{A}$ incrementally instead of (4.37) [64]:

$$\mathbf{A}^{new} = \mathbf{A} - \frac{\mathbf{a}_i \mathbf{a}_i^T}{\delta + a_{ii}} \tag{4.59}$$

$$\delta = \left(\frac{1}{v_i^{new}} - \frac{1}{v_i^{old}}\right)^{-1} \tag{4.60}$$

Table 4.2: Test error rate (± one standard deviation) on the 'Breast', 'Heart', 'Ionosphere' and 'Pima' datasets for γ-EP, EP and SVM models.

Model	Breast	Heart	Ionosphere	Pima
γ-EP	0.038 ± 0.013	0.171 ± 0.032	0.135 ± 0.032	0.256 ± 0.026
EP	0.037 ± 0.013	0.169 ± 0.030	0.134 ± 0.032	0.239 ± 0.025
SVM	0.067 ± 0.022	0.169 ± 0.027	0.071 ± 0.021	0.236 ± 0.023

Table 4.3: P-values from a Wilcoxon paired signed rank test comparing average accuracy of γ-EP to EP and SVM. Statistically significant results at the 1% level are shown in bold.

	Breast	Heart	Ionosphere	Pima
γ-EP:EP	8.7×10^{-1}	8.7×10^{-1}	9.2×10^{-1}	$\mathbf{2.7 \times 10^{-3}}$
γ-EP:SVM	$\mathbf{7.1 \times 10^{-13}}$	8.6×10^{-1}	$< \mathbf{2.2 \times 10^{-16}}$	$\mathbf{1.2 \times 10^{-4}}$

We applied the same analysis to assess the accuracy of the kernel BPM as the linear BPM against the SVM [117] on the 'Breast' ($n = 569$, $d = 30$), 'Heart' ($n = 297$, $d = 13$), 'Ionosphere' ($n = 351$, $d = 33$) and 'Pima' ($n = 767$, $d = 8$) datasets from the UCI repository [23]. The validation set was used to select $\sigma \in \{1, 2, 3, 4, 5\}$ for the Gaussian kernel (4.36) and $\gamma \in \{-0.5, 0, 0.5\}$ for γ-EP which we compare against canonical EP ($\gamma = -1$). We also

Table 4.4: Average predictive log-likelihood (± one standard deviation) on 'Breast', 'Heart', 'Ionosphere' and 'Pima' datasets for γ-EP and EP models.

Model	Breast	Heart	Ionosphere	Pima
γ-EP	4.49 ± 0.005	3.96 ± 0.012	4.09 ± 0.012	4.92 ± 0.013
EP	4.49 ± 0.005	3.95 ± 0.011	4.08 ± 0.011	5.04 ± 0.020

Table 4.5: P-values from a Wilcoxon paired signed rank test comparing average predictive log likelihood of γ-EP to EP. Statistically significant results at the 1% level are shown in bold.

	Breast	Heart	Ionosphere	Pima
γ-EP:EP	2.6×10^{-1}	$\mathbf{7.5 \times 10^{-6}}$	$\mathbf{2.2 \times 10^{-4}}$	$< \mathbf{2.2 \times 10^{-16}}$

added a soft-margin constant $\epsilon^2 \in \{10, 100\}$ to the main diagonal of the kernel matrix. This is the square of the noise variance in (4.6) [77]:

$$\mathbf{K}_\epsilon = E[(\mathbf{w}^T\mathbf{X} + \xi)(\mathbf{w}^T\mathbf{X} + \xi)^T] - E[(\mathbf{w}^T\mathbf{X} + \xi)]E[(\mathbf{w}^T\mathbf{X} + \xi)^T] = \mathbf{K} + \epsilon^2\mathbf{I} \tag{4.61}$$

and it is added to (4.58). Unfortunately, we could not use Bayesian model comparison because the evidence is NaN. The results are given in Table 4.2. We found that γ-EP does not improve the accuracy of canonical EP in kernel classification. We tested the statistical significance of the results using a Wilcoxon [122] paired signed rank test with a 1% significance level, shown in Table 4.3. We found that the SVM significantly outperformed the BPM on 'Ionosphere' and 'Pima' and the BPM outperformed the SVM on 'Breast'. Furthermore, EP outperformed γ-EP on 'Pima'. To get a more fine-grained comparison we also computed the average predictive log likelihood for EP and γ-EP, shown in Table 4.4. The average predictive log likelihood aggregates the classifier's ability to quantify uncertainty in the form of the probability of a correct classification over the 50 test sets. The results of a Wilcoxon paired rank sign test on the average predictive log likelihoods are shown in Table 4.5. Although, γ-EP has a slightly higher test error than EP on 'Heart' and 'Ionosphere', it has a significantly better average predictive log likelihoods.

The γ-EP algorithm for the kernel Bayes Point Machine.

1. Initialise $v_i = \infty, m_i = 0, s_i = 1, h_i = 0, \lambda_i = \mathbf{K}_{ii}, \alpha_i = 0$.

2. Until (m_i, v_i) converges (change less than 10^{-4}).

For i = 1,...,n:

Compute 'old' posterior:

$$h_i^{\setminus i} = h_i + \lambda_i v_i^{-1}(h_i - m_i + \gamma v_i \alpha_i)$$

Update 'new' posterior and $\tilde{t}_i$:

$$\begin{aligned}
z_i &= \frac{h_i^{\setminus i}}{\sqrt{\lambda_i}} \\
\alpha_i &= \frac{1}{\sqrt{\lambda_i}} \frac{\mathcal{N}(z_i)}{\Phi(z_i|0,1)} \\
h_i &= h_i^{\setminus i} + \lambda_i \alpha_i \\
v_i &= \lambda_i \left(\frac{1}{\alpha_i h_i} - 1 \right) \\
m_i &= h_i^{\setminus i} + \lambda_i \alpha_i = h_i \\
s_i &= Z_i \sqrt{1 + v_i^{-1} \lambda_i} \exp\left(\frac{1}{2} \alpha_i^2 (\lambda_i - \gamma v_i)(v_i + \lambda_i)^{-1} (\lambda_i - \gamma v_i) \right)
\end{aligned}$$

Update 'new' posterior with $\mathbf{A}$:

$$\begin{aligned}
\mathbf{A} &= \left(\mathbf{K}^{-1} + \Lambda^{-1} \right)^{-1} \\
&= \mathbf{A} - \frac{\mathbf{a}_i \mathbf{a}_i^T}{\delta + a_{i,i}} \\
\delta &= \left(\frac{1}{v_i^{new}} - \frac{1}{v_i^{old}} \right)^{-1}
\end{aligned}$$

For all i:

$$\begin{aligned}
h_i &\approx \sum_j \mathbf{A}_{ij} \frac{m_j - \gamma v_j \alpha_j}{v_j} \\
\lambda_i &\approx \left(\frac{1}{\mathbf{A}_{ii}} - \frac{1}{v_i} \right)^{-1}
\end{aligned}$$

Compute the evidence:

$$\begin{aligned}
B &= \sum_{ij} \mathbf{A}_{ij} \frac{m_i m_j}{v_i v_j} - \sum_i \frac{m_i^2}{v_i} \\
p(D) &\approx \frac{|\Lambda|^{\frac{1}{2}}}{|\mathbf{K} + \Lambda|^{\frac{1}{2}}} \exp(B/2) \prod_i s_i
\end{aligned}$$

4.3 Application: Oncogenic Single Nucleotide Variants

In the final subsection of this chapter, we turn our attention to the binary classification of oncogenic (cancer causing) single nucleotide variants (SNVs). The goal for this task is to predict which SNVs of the human genome will 'drive' the growth of tumours by assigning a +1 to an oncogenic SNV and a -1 to a non-oncogenic (neutral) SNV. The classical mechanism for the development of cancer is that the genome contained in every human cell is subject to random bursts of radiation and thereby over time accumulate mutations including *driver mutations* that trigger the uncontrolled growth of tumours. This mechanism makes four immediate predictions: that cancers will be more common among the elderly, those who receive more radiation will be more likely to develop cancer, that cancers are somatic and not inherited and that certain genes suppress the growth of tumours, all of which have been confirmed and have become common knowledge. What it leaves out is precisely which mutations drive the development of tumours. In this subsection, we will set ourselves the ambitious task of building a classifier capable of predicting which of the roughly $3^2 \times 10^9$ possible SNVs of the human genome are oncogenic.

The biggest obstacle to the application of predictive models to the cancer genome are the quality of the labels and the choice of features but not the number of possible variants - all of which can be predicted by the same classifier. It is patently true that we do not yet know definitively which mutations drive the growth of tumours and which mutations are simply along for the ride and appear afterwards as the cancer genome evolves. These mutations are called *passenger mutations*. The theory of neutral evolution [51] suggests that a majority of these passenger mutations, occurring not by selective forces, are irrelevant for the growth and survival of the tumour. Any oncogenic SNV database is a snapshot of a particular cancer genome at a single point in time. Therefore, we have a very unclean dataset where the positive class of oncogenic SNVs is polluted with a majority of label noise (passengers). There are tools [71] which seek to remove data points with noisy labels to preserve a kernel of clean training data. However, even these methods are not appropriate for this problem because we do not have a complete set of features which define the cancer genome. Furthermore, we do not know without using a priori assumptions which features are most appropriate for distinguishing cancer SNVs from non-cancer SNVs [95]. Kernel methods map a finite set of features to an infinite

dimensional feature space and could overcome some of the errors in the original feature representation.

As every tumour evolves from a single normal cell, and accumulates selectively advantageous mutations due to genetic instability [72], the evolutionary history of a specific tumour exists on a continuum and that continuum is best represented by a phylogenetic tree. Therefore, not only are tumours evolving by genetic drift, they are also adapting to new selectively advantageous mutations, which may even replace the initial driver mutations and thereby sweep away the cancer's history like footsteps in the sand. This analogy provides an intuition for the finding that the mean number of drivers does not increase as the disease progresses [95].

The recurrence level, r, of an SNV is the number of independent patients with the same variant at the same site of the genome. The recurrence level is widely used as a proxy for the likelihood of a specific cancer variant being a driver [11]. However, this assumption is often fallacious. As the database contains a mixture of different types of cancers at different stages of tumour growth, increasing the recurrence level would only purify the labels if the same drivers were common to all cancers. Even if there are some drivers which are preserved at high recurrence levels, the data will be dominated by passengers due to the saturation effect of only having 4 possible nucleotide variants per site {A,C,G,T}. The probability of 2 genomes being identical is extremely low but the probability of any 2 genome sites having the same variant is extremely high. Furthermore, if high recurrence level variants were purified of passengers, we would see a non-decreasing relationship between classification accuracy and the recurrence level. Rogers et al. [96], using their predictor 'CScape' trained on neutral vs oncogenic SNVs, show that this is not the case and the accuracy decreases as the recurrence level is increased.

We aim to test whether the BPM can outperform the Gradient Boosting Machine (GBM) [28] used by CScape on the same training data. Gradient Boosting is a sophisticated ensemble learning algorithm that frequently outperforms other classification algorithms in machine learning challenges. It works by sequentially optimizing an additive model under an exponential error function [5]. In order to make kernel machines competitive with ensemble learning classifiers, we can use a combination of multiple kernels. This problem is called multiple kernel learning (MKL) and the task is to find an optimal linear

combination of kernel coefficients λ_l for a given set of p kernel matrices $\{K_l : l = 1,...,p\}$ to give a composite kernel K_λ:

$$K_\lambda = \sum_{l=1}^{p} \lambda_l K_l \tag{4.62}$$

where $\sum_{l=1}^{p} \lambda_l = 1, \lambda_l \geq 0$ [16]. We will use the MKL algorithm 'MKLdiv-dc' [124]. The MKLdiv-dc algorithm works by minimising the KL-divergence between $\mathcal{N}(0,K_\lambda)$ and $\mathcal{N}(0,K_y = yy^T)$. The target kernel, yy^T, defines the 'ideal' kernel for the training data. We call $K_y = yy^T$ 'ideal' in the sense of kernel-target alignment [20], where the difference between the sum of the between class distances and the sum of the within class distances is equivalent to the alignment between K_λ and K_y. The MKLdiv-dc algorithm defines another type of kernel alignment using the KL-divergence. There are two different MKL algorithms because the KL-divergence is not symmetric and the other algorithm is called 'MKLdiv-conv'. MKLdiv-dc uses the same form of the KL-divergence as EP which we have seen is equivalent to matching moments, $K_y = K_\lambda$, so we expect it to fit better to the training data than MKLdiv-conv, which matches modes instead of moments. The MKLdiv-dc algorithm is given by:

$$\begin{aligned}
\text{argmin}_\lambda KL(\mathcal{N}(0,K_y)||\mathcal{N}(0,K_\lambda)) &= -\text{argmin}_\lambda \int \left[\log(\mathcal{N}(0,K_\lambda)) - \log(\mathcal{N}(0,K_y))\right] \mathcal{N}(0,K_y)\, dy \\
&= \text{argmin}_\lambda \frac{1}{2} \int \left[\log\left(\frac{|K_\lambda|}{|K_y|}\right) - y^T K_y^{-1} y + x^T K_\lambda^{-1} x\right] \mathcal{N}(0,K_y)\, dy \\
&= \text{argmin}_\lambda \frac{1}{2} \log\left(\frac{|K_\lambda|}{|K_y|}\right) - \frac{1}{2}\mathbb{E}_y[y^T K_y^{-1} y] + \frac{1}{2}\mathbb{E}_y[x^T K_\lambda^{-1} x] \\
&= \text{argmin}_\lambda \frac{1}{2} \text{Tr}[K_y K_\lambda^{-1}] + \frac{1}{2}\log|K_\lambda| - \frac{1}{2}\log|K_y| - \frac{n}{2}
\end{aligned}$$

where we have used the identity for the expectation of Gaussian quadratic forms [81] to go from the third to fourth line. By adding a jitter term β to make the matrix inversions more tractable and removing the constant terms we get the MKLdiv-dc optimisation problem [124]:

$$\text{argmin}_\lambda \text{Tr}\left(K_y (K_\lambda + \beta \mathbf{I})^{-1}\right) + \log|K_\lambda + \beta \mathbf{I}| \tag{4.63}$$

for details on how to solve this problem see the original paper [124]. We used the R package 'Rsolnp' to solve the optimization problem [30] [123]. Other multiple kernel learning algorithms optimise different criteria such as maximising the margin. Lanckriet et al. [54] pioneered this approach using semi-definite programming (SDP). Further development included more efficient implementations [89] [107]. However, this approach

is unsuitable for the Bayes point machine trained with EP as the MKL algorithm is embedded within the SVM optimisation problem. Whereas, because MKLdiv-dc is performed independently of the classifier by using the target kernel, it can be combined with any kernel classifier, including the BPM trained with EP.

For the experiment we use the same dataset as used for CScape [96]. The (positive) pathogenic labels for cancer SNVs were gathered from the COSMIC [108] database and the (negative) non-cancer SNVs were gathered from the 1,000 Genomes Project [1]. Rogers et al. [96] balanced the high bias of using a high recurrence level and the high variance of using a low recurrence level and settled on $r = 5$ for coding regions. We restrict our attention to coding regions as we've previously found MKL provides an insignificant benefit in non-coding regions likely due to high level of noise in the labels. For more details on the data pre-processing pipeline see [96].

All features are based on the GRCh37/hg19 version of the human genome. The features for the coding regions are based on 4 'feature groups' which Rogers et al. called 'Evolutionary', 'Variant Effect Predictor (VEP)', 'Distance' and 'Spectrum'. Evolutionary features include: PhastCons [104] conservation probability for each site, PhyloP [84] conservation score and a range of features built from HMMER software package representing the emission probabilities of each variant at each site of the alignment [102] [103]. The VEP feature group includes 35 features which count the number of transcripts such as: UTR, missense & TF binding sites that are impacted by a particular mutation and two 20-element amino acid indicator features for the wild-type sequence and the mutation. The Spectrum features are counts indicating how many times a specific pattern is present in a window around a specific site. The possible patterns are the set of k-mers below a certain length. For CScape, a window size of 3 is used and the maximum k-mer size is 2. So there are 2 windows each contributing 20 features. We can then pass this 40-element feature vector through a Gaussian Kernel as in Rogers et al [96]. However, we can also map these count vectors to a Spectrum kernel matrix, $\mathbf{K}^*$ [16]:

$$\mathbf{K}^*_{ij} = \mathbf{x}_i^T \mathbf{x}_j \tag{4.64}$$

where $\mathbf{x}_i$ is the ith Spectrum feature vector. The Distance feature group measures the distance from each SNV to gene features annotated by ENSEMBL such as: start codon, stop codon, gene, UTR, CDS and exon [96]. There is likely to be some redundancy between the VEP and Distance feature groups and this will be learned by the MKL kernel coefficients.

To evaluate our models, we used leave-one-chromosome-out cross validation (LOCO-CV) in which one chromosome is held out as a test set and the models are trained on the remaining 21 chromosomes (we leave the X & Y chromosomes out of all of the experiments as they were not used by CScape). The data is cycled until every chromosome has been classified and an average accuracy score is computed. We use three balanced non-overlapping sets of 1,100 data points (50 data points per chromosome) to speed-up the computations and find our results are within 3 percentage points of the original CScape but it is the relative performance that we are interested in and not the minimum achievable errors which can always be improved by training on the full datasets. We used a separate validation set of 50 data points per chromosome to tune the Gaussian $\sigma \in \{3,5,7,9\}$ in (4.36), the soft-margin constant $\epsilon^2 \in \{20,100\}$, the SVM $C \in \{10^{-1},1,10\}$ parameter and γ giving $\gamma = -1$, $\sigma = 9$ for BPM, $\sigma = 5$ for SVM-MKL1 and $\sigma = 3$ for SVM, BPM-MKL1, BPM-MKL2 and SVM-MKL2, $\epsilon^2 = 20$ and C = 1. The BPM-MKL1 & SVM-MKL1 models are trained with the Spectrum feature vector in an isotropic Gaussian kernel. The Spectrum kernel, MKL2, performed better than MKL1 for BPM but not for SVM. All other features are passed through the isotropic Gaussian kernel. The results of the experiments are given in Table 4.6. Multiple Kernel Learning failed to improve the SVM average accuracy but did improve the BPM. We see that the BPM is the worst performing model but by combining it with MKL2 it outperforms the SVM and SVM-MKL1. However, the BPM-MKL2 is still almost three percentage point worse than gradient boosting and it has a lower average predictive log likelihood than the BPM. We tested the statistical significance of the results using a Wilcoxon [122] paired signed rank test with a 1% significance level, shown in Table 4.7. We didn't find that any of the LOCO-CV results were statistically significant, but the BPM showed a statistically significant improvement in predictive log likelihood over BPM-MKL2. Figure 4.5 shows the kernel coefficients λ_l for the MKL1 (left) and MKL2 (right) models. Surprisingly, we see the spectrum kernel (MKL2) has a lower weighting than the spectrum vector passed through a Gaussian kernel (MKL1).

Table 4.6: Leave-one-chromosome-out cross validation accuracy (± one standard deviation) for GBM, BPM, SVM and MKL classifiers and average leave-one-chromosome-out predictive log likelihood (± one standard deviation) for BPM classifiers.

Chrom	BPM	BPM-MKL2	SVM	SVM-MKL1	GBM
1	0.667	0.7	0.667	0.673	0.727
2	0.7	0.653	0.687	0.68	0.76
3	0.627	0.64	0.647	0.68	0.62
4	0.687	0.647	0.68	0.667	0.733
5	0.587	0.633	0.64	0.62	0.647
6	0.753	0.747	0.747	0.76	0.793
7	0.613	0.633	0.667	0.647	0.707
8	0.553	0.62	0.6	0.6	0.673
9	0.713	0.753	0.727	0.74	0.74
10	0.627	0.707	0.727	0.693	0.767
11	0.513	0.567	0.513	0.553	0.587
12	0.673	0.693	0.68	0.673	0.767
13	0.667	0.673	0.667	0.68	0.66
14	0.66	0.673	0.693	0.673	0.693
15	0.593	0.627	0.613	0.633	0.68
16	0.673	0.747	0.733	0.72	0.68
17	0.707	0.64	0.68	0.693	0.687
18	0.653	0.667	0.653	0.647	0.693
19	0.567	0.66	0.6	0.627	0.7
20	0.713	0.773	0.727	0.733	0.76
21	0.56	0.627	0.6	0.567	0.673
22	0.673	0.733	0.74	0.7	0.647
Av. acc	0.645 ± 0.062	0.673 ± 0.053	0.668 ± 0.058	0.666 ± 0.052	0.700 ± 0.052
Av. pll	3.30 ± 0.029	3.24 ± 0.008			

Table 4.7: P-values from a Wilcoxon paired signed rank test between leave-one-chromosome-out accuracy of BPM-MKL2 and GBM, BPM, SVM and SVM-MKL1 models and between leave-one-chromosome-out predictive log likelihood of BPM-MKL2 and BPM models.

	BPM-MKL:BPM	BPM-MKL:SVM	BPM-MKL:SVM-MKL	BPM-MKL:GBM
p-value acc	2.5×10^{-1}	9.6×10^{-1}	9.4×10^{-1}	7.7×10^{-2}
p-value pll	$\mathbf{3.3 \times 10^{-9}}$			

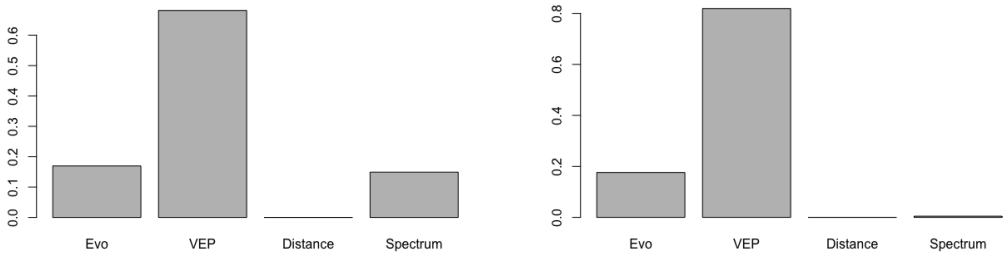


Figure 4.5: Average kernel coefficients λ_l for MKL1 (left) and MKL2 (right).

4.4 Concluding Remarks

In this chapter, we used the γ-EP algorithm to train Bayes point machines. We demonstrated a key flaw in canonical EP for training linear BPMs - if any of the data points are repeated, canonical EP cannot recognise the added redundancy which skews the decision boundary away from the Bayes point. We found that setting $\gamma > 0$ leads to a modified loss function α_i which encourages sparsity and mitigates the problem greatly, even when the support vectors are repeated. We compared the performance of γ-EP for sparse linear classification against the C-SVM and ν-SVM as well as canonical EP trained with the support vectors from the SVMs.

We also extended γ-EP to non-linear Bayes point classification using kernel matrices. However, the accuracy did not improve with $\gamma \neq -1$ and the sparsity demonstrated for linear classifiers with $\gamma > 0$ does not remain in infinite dimensional feature space. The α_i's are all equal at convergence. The linear Bayesian SVM has been extended to use kernel matrices [39] and the point process prior approach [115] can also use kernel matrices and automatically determine the number of support vectors. EP can be combined with automatic relevance determination [24] to obtain sparsity for radial basis function kernel classifiers using either the evidence or leave-one-out-error [88].

The results in this chapter do not account for label noise. Label noise can be modelled by assuming an iid flip process with label error rate κ [77] which corresponds to a modified loss function α_i in EP. The EM-EP [50] algorithm can be used to sequentially alternate between EP iterations to approximate the posterior and updating κ by maximising a lower bound to an approximation of the log marginal likelihood. D. Hernández-Lobabto and J. M. Hernández-Lobato [45] gave another approach in which κ is updated within

the EP algorithm, which they showed outperforms EM-EP.

We applied the BPM trained with EP to the problem of classifying oncogenic single nucleotide variants. This is a difficult classification problem; not because the labels are noisy, but because they're of low quality. Certainly any labelling errors are not symmetric. It is far more likely that non-oncogenic mutations are mislabelled as oncogenic than the other way around. Nevertheless, we sought to combine EP with several heterogeneous genomic data sources using multiple kernel learning (MKL). However, the BPM-MKL could not outperform gradient boosting used in the CScape [96] classifier. Nevertheless, the BPM-MKL could improve by the addition of more low quality feature groups which is unlikely to improve gradient boosting.

CHAPTER 5

PHYLOGENETIC LINEAR GAUSSIAN MODELS

"The problem seems terribly complicated at present, because in all this detail we do not know what is relevant, what is irrelevant... It might turn out that prediction of biological activity requires about only a dozen separate factors, instead of a million. If so, then one would have both the courage and insight needed to attack more complicated problems. "

— Edwin. T. Jaynes

This chapter presents two new phylogenetic comparative methods used to correct for the non-independence of related species in linear Gaussian models. The first method, Phylogenetic Relevance Vector Machine (PhyRVM), estimates the phylogenetic signal by maximising the marginal likelihood while automatically pruning irrelevant features. It achieves superior estimates of phylogenetic signal than the widely used maximum likelihood approach [78]. We apply the classical RVM [112] to predict prokaryotic optimal growth temperature (OGT). We also predict a hyperthermophilic last universal common ancestor (LUCA). The second model, Phylogenetic Probabilistic Principal Components Analysis (P3CA), is a probabilistic dimensionality reduction technique capable of estimating phylogenetic signal by maximum likelihood.

5.1 Phylogenetic Comparative Methods

A *phylogeny*, or evolutionary/bifurcating tree, is a clustering of a set of related species (called *taxa*) based on their genetic similarity. The underlying assumption of phylogenetic inference is that closely related species are more genetically similar than distantly

related species [38]. However, the true phylogeny can never be known unless it is artificially constructed. Nevertheless, approximate phylogenies have found a wide range of applications in computational and comparative biology [37] [19] and phylogenetic inference methods are continually improving [48].

The *comparative method* in evolutionary biology consists of a suite of statistical methods for the analysis of *phenotypic traits* which aim to correct for the statistical *non-independence* associated with related taxa. Furthermore, many parametric models have been developed to measure evolutionary quantities such as rates of evolution and strength of selection [36] and some biologists view the estimation of these microevolutionary parameters as a more significant problem than correcting for non-independence [59]. However, all of these more elaborate microevolutionary models must first successfully incorporate the phylogeny into a statistical model as well.

The successful incorporation of a phylogeny into a classical statistical model such as linear regression [32] which is designed for independent data is not trivial. If one is not careful, one can easily diminish the predictive power of the model by adding a phylogeny. In the comparative biology literature, models are typically compared with statistics gathered from the training data such as: training mean square error, R^2 and the log likelihood [27]. From these statistics alone it is not clear how important the phylogeny of related taxa is to the statistical model. And yet, the comparative biologists are right to persist and emphasise the importance of including a phylogeny in their statistical models. The phylogeny is a very powerful tool which contains a lot more information than any individual comparative dataset. It is clear the phylogeny is learning something underlying all comparative problems.

A phylogeny of N taxa can be incorporated into statistical models by transforming it into an N x N covariance matrix. The evolution of a trait is typically assumed to be due to Brownian motion. The modelling assumptions we make under a Brownian motion model are that: changes in the trait are independent of their previous state, the traits are normally distributed with zero mean and the variances are proportional to the sum of the branch lengths to the root. The Brownian motion model can represent traits evolving under genetic drift and genetic drift with mutation [59]. From these assumptions we can

derive the phylogenetic covariance matrix rule [59]:

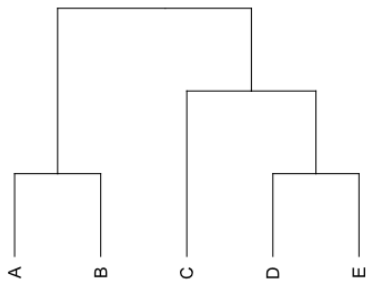
$$Cov[t_i, t_j] = Cov\left(\mathbb{E}[t_i|t_a], \mathbb{E}[t_j|t_a]\right) \tag{5.1}$$

$$= Cov[t_a, t_a] = Var[t_a] \tag{5.2}$$

$$= \sigma^2 v_a \tag{5.3}$$

where t_i and t_j are traits of two species, t_a is a common ancestor, σ^2 is the constant variance of the trait implying a constant rate of evolution and v_a is the sum of branch lengths from the root to the most recent common ancestor, that is, the amount of evolutionary time two distinct species were latent in a common ancestor for a given phylogeny. An example of a simple phylogeny with 5 taxa is shown in Figure 5.1. The phylogenetic covariance matrix $\mathbf{V}$ constructed from this phylogeny is given in (5.4), where $\{v_A, v_B, v_C, v_D, v_E\}$ are the tip lengths of branches A-E respectively and $v_{A,B}$ is the distance shared by branches A and B to the root. The diagonal elements of $\mathbf{V}$ are the root-to-tip distances for each taxon. The off-diagonal elements of $\mathbf{V}$ (the ith row and the $j$the column such that $i \neq j$) are given by the sum of the shared root to tip branch lengths between the ith and jth taxa. Each off-diagonal element of the phylogenetic covariance matrix is called a 'phylogenetic correlation' and all of the off-diagonal elements taken together is called the 'phylogenetic signal'. From (5.3) we can see the phylogenetic correlation between two species decreases linearly with the time since they diverged under a Brownian motion model [59].

Figure 5.1: A simple phylogeny with 5 taxa.



(5.4)

$$\mathbf{V} = \begin{pmatrix} \nu_{(A,B)} + \nu_A & \nu_{(A,B)} & 0 & 0 & 0 \\ \nu_{(A,B)} & \nu_{(A,B)} + \nu_B & 0 & 0 & 0 \\ 0 & 0 & \nu_{(C,(D,E))} + \nu_C & \nu_{(C,(D,E))} & \nu_{(C,(D,E))} \\ 0 & 0 & \nu_{(C,(D,E))} & \nu_{(C,(D,E))} + \nu_{(D,E)} + \nu_D & \nu_{(C,(D,E))} + \nu_{(D,E)} \\ 0 & 0 & \nu_{(C,(D,E))} & \nu_{(C,(D,E))} + \nu_{(D,E)} & \nu_{(C,(D,E))} + \nu_{(D,E)} + \nu_E \end{pmatrix}$$

(5.5)

$$\mathbf{V}_\lambda = \begin{pmatrix} \nu_{(A,B)} + \nu_A & \lambda \times \nu_{(A,B)} & 0 & 0 & 0 \\ \lambda \times \nu_{(A,B)} & \nu_{(A,B)} + \nu_B & 0 & 0 & 0 \\ 0 & 0 & \nu_{(C,(D,E))} + \nu_C & \lambda \times \nu_{(C,(D,E))} & \lambda \times \nu_{(C,(D,E))} \\ 0 & 0 & \lambda \times \nu_{(C,(D,E))} & \nu_{(C,(D,E))} + \nu_{(D,E)} + \nu_D & \lambda \times (\nu_{(C,(D,E))} + \nu_{(D,E)}) \\ 0 & 0 & \lambda \times \nu_{(C,(D,E))} & \lambda \times (\nu_{(C,(D,E))} + \nu_{(D,E)}) & \nu_{(C,(D,E))} + \nu_{(D,E)} + \nu_E \end{pmatrix}$$

Pagel [78] developed a parametric modification to the phylogenetic covariance matrix which allows the amount of phylogenetic signal in the residuals of a linear regression model to be measured [92]. The modification is often called Pagel's λ and it is a positive scalar which is multiplied to all the off-diagonal elements of the phylogenetic covariance matrix $\mathbf{V}_\lambda$ represented by (5.5). Therefore, $\lambda = 1$ represents the original Brownian motion tree and $\lambda = 0$ represents the star tree in which all taxa radiate from the root at the same time (though the branch lengths of the star tree can still differ). The star tree implies the residuals contain no phylogenetic signal and are independent [92] and $\lambda \in [L, U]$, where U is slightly greater than 1 and L is slightly less than 0. Therefore, Pagel's λ typically has the effect of shortening the internal branches of the phylogeny and we will often refer to λ as the phylogenetic signal. Negative phylogenetic signal does not have an accepted biological interpretation because the branch lengths of the phylogeny measure the expected number of amino acid substitutions per site along the branch [3].

Now, we will introduce the *phylogenetic least squares* (PGLS) [14]. For any vector of N phenotypic traits $\mathbf{t}$, we can express it as a sum of a linear combination of some N x (M+1) matrix of input data $\mathbf{X}$ with an $M+1$-dimensional weight vector $\mathbf{w}$ and an additive Brownian motion 'error' term $\boldsymbol{\epsilon}$ defined by a zero mean multivariate Gaussian with precision (inverse variance) $\beta \mathbf{V}_\lambda^{-1}$:

(5.6) $$\mathbf{t} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon}$$

The likelihood function of $\mathbf{t}$ is given by:

$$p(\mathbf{t}|\mathbf{w},\beta,\lambda) = \mathcal{N}(\mathbf{t}|\mathbf{X}\mathbf{w},\beta^{-1}\mathbf{V}_\lambda) \tag{5.7}$$

Here, we have used a multivariate Gaussian instead of assuming the data points are independent. We have also appended a column of ones to $\mathbf{X}$ and a bias to $\mathbf{w}$. The value of λ is estimated using a numerical optimization procedure such as Brent's method [9] to maximise the log likelihood:

$$\mathrm{L} = \frac{N}{2}\log(\beta) - \frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{V}_\lambda| - \beta E_\lambda(\mathbf{w}) \tag{5.8}$$

$$E_\lambda(\mathbf{w}) = \frac{1}{2}(\mathbf{t}-\mathbf{X}\mathbf{w})^T\mathbf{V}_\lambda^{-1}(\mathbf{t}-\mathbf{X}\mathbf{w}) \tag{5.9}$$

Differentiating L with respect to $\mathbf{w}$ and β and setting the derivative to zero and rearranging yields the simple expressions:

$$\mathbf{w}_{ML} = (\mathbf{X}^T\mathbf{V}_\lambda^{-1}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{V}_\lambda^{-1}\mathbf{t} \tag{5.10}$$

$$\beta_{ML} = \frac{N}{2\,E_\lambda(\mathbf{w})} \tag{5.11}$$

However, according to Freckleton, Harvey & Pagel "There is not a corresponding simple expression for λ" [27] or Revell "We do not have an analytic solution for this equation, so it must be optimized numerically" [92]. This can be proved by using a novel representation for $\mathbf{V}_\lambda$:

$$\mathbf{V}_\lambda = \lambda\mathbf{V}_0 + \mathbf{V}_{ii} \tag{5.12}$$

$$\mathbf{V}_D = \mathbf{V}_0 + \frac{1}{\lambda}\mathbf{V}_{ii} \tag{5.13}$$

$$\mathbf{V}_\lambda = \lambda\mathbf{V}_D \tag{5.14}$$

$$\mathbf{V}_\lambda^{-1} = \frac{1}{\lambda}\mathbf{V}_D^{-1} \tag{5.15}$$

where $\mathbf{V}_0$ is a matrix with off-diagonal elements equal to $\mathbf{V}_\lambda$ and diagonal elements equal to zero and $\mathbf{V}_{ii}$ is the diagonal matrix of tip lengths. The interpretation of $\mathbf{V}_D$ is opposite to $\mathbf{V}_\lambda$: for $\lambda \in [0,1]$ the tip lengths extend but the interior branch lengths remain the same. This is most striking as $\lambda \to 0$ and $\mathbf{V}_D$ approaches a star tree with infinitely long

tips. Now, we can rewrite the log likelihood in terms of $\mathbf{V}_D$:

$$\mathrm{L} = \frac{N}{2}\log(\beta) - \frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\lambda\mathbf{V}_D| - \frac{\beta}{\lambda}E_D(\mathbf{w}) \tag{5.16}$$

$$E_D(\mathbf{w}) = \frac{1}{2}(\mathbf{t}-\mathbf{X}\mathbf{w})^T\mathbf{V}_D^{-1}(\mathbf{t}-\mathbf{X}\mathbf{w}) \tag{5.17}$$

and take the derivative with respect to λ and set it to zero:

$$\frac{\partial \mathrm{L}}{\partial \lambda} = \frac{\beta E_D(\mathbf{w})}{\lambda^2} - \frac{N}{2\lambda} = 0 \tag{5.18}$$

$$\lambda^{new} = \frac{2\beta E_D(\mathbf{w})}{N} \tag{5.19}$$

At first, it seems as though we have derived a new expression for λ_{ML}. However, by rewriting (5.11) in terms of $\mathbf{V}_D$, we see nothing is gained:

$$\beta = \frac{N\lambda^{old}}{2E_D(\mathbf{w})} \tag{5.20}$$

$$\lambda^{new} = \frac{2\beta E_D(\mathbf{w})}{N} = \lambda^{old} \tag{5.21}$$

Therefore, whichever update, (5.20) or (5.21), comes second will be left unchanged. As (5.10) and (5.11) are independent closed-form solutions [101], the maximum likelihood method is non-iterative and we cannot refine our estimates of β and λ or remove their dependence on the initial values. The method of maximum likelihood is inappropriate for an *analytical* treatment of Pagel's λ in comparative least squares and with this in mind we turn our attention to Bayesian methods, specifically those which maximise the marginal likelihood (evidence).

In section 5.1.1, we derive a new phylogenetic regression model by maximising the evidence and give the first analytical solution for Pagel's λ. We show on simulated data that maximising the evidence gives more accurate estimates of Pagel's λ than maximising the likelihood and a lower root mean square error in cross validation on a real dataset of prokaryotic optimal growth temperatures. In section 5.1.2, we extend Probabilistic Principal Components Analysis to use phylogenetic covariance matrices. In section 5.2, we evaluate the 'relevant' features of the Relevance Vector Machine (RVM) and find that relevance does not imply correlation with the trait. In section 5.3, we train two

RVM models to predict archaeal and bacterial optimal growth temperatures (OGT) using genome derived features including amino acid proportions and find that the archaeal model outperforms the state-of-the-art in the literature. In section 5.4, we reconstruct the amino acid sequences of the ancestral prokaryotes to extant archaea and bacteria and use an RVM to predict the OGT of the last universal common ancestor.

5.1.1 Phylogenetic Relevance Vector Machine (PhyRVM)

For a Bayesian treatment of phylogenetic regression we need a prior over the weights $\mathbf{w}$. We choose a Gaussian prior as it is conjugate to the Gaussian likelihood. Here, we use an *automatic relevance determination* (ARD) prior [58] by placing a separate parameter over each feature (including the bias):

$$p(\mathbf{w}|\alpha) = \prod_{i=0}^{M} \mathcal{N}(0, \alpha_i^{-1}) \tag{5.22}$$

Each α_i represents an inverse-length scale of the covariance and learning these parameters involves stretching the covariance in the dimensions providing the most uncertainty and thereby contracting the covariance in the dimensions providing most information. Therefore, α_i^{-1} can be used to remove 'irrelevant' features and rank the rest by their 'relevance'. The features with the largest value of α_i^{-1} are the most relevant. This is why the procedure is called automatic relevance determination and it is very widely used in a variety of Bayesian machine learning models such as the Tipping's *Relevance Vector Machine* (RVM) [112]. By incorporating a phylogenetic covariance matrix into the RVM, we derive a new algorithm which we call the *Phylogenetic Relevance Vector Machine* (PhyRVM).

We could have instead explicitly separated the majority of irrelevant features from the few relevant ones using a Bernoulli latent variable Z_i. This prior forms a mixture of a *slab* distribution (e.g. Gaussian when $Z_i = 0$) and a *spike* distribution (e.g point mass at $w_i = 0$ when $Z_i = 1$). The advantage of the spike and slab prior [67] over ARD is that the few relevant features or slab are separated from the rest or spike producing truly sparse solutions without using a threshold. However, computing the evidence is intractable with this prior as it requires evaluating all 2^n states of the Bernoulli distribution [12], precluding analytic solutions. We can also specify a *hyperprior* over α. In the case of ARD, a suitable choice is the Gamma distribution [112]. If we instead had chosen the half-Cauchy distribution ($\mathrm{C}^+(0,1)$), we would have the *horseshoe* prior [17]. The name

comes from the horseshoe shape density of the Beta distribution (Be(0.5,0.5)) for the shrinkage weights κ_i:

$$\kappa_i = \frac{1}{1+\alpha_i^{-2}} \tag{5.23}$$

which means that total shrinkage ($\kappa_i = 1$) and no shrinkage ($\kappa_i = 0$) are both contained in the same model. This creates a discontinuity that could be more appropriate to distinguish between relevant and irrelevant features than ARD. However, similar to ARD, a threshold is required to determine which features are relevant. An alternative to the Bayesian sparsity inducing priors is the frequentist *Lasso* [111] given by adding an L_1 penalty term to the log likelihood:

$$L^{Lasso} = L + \eta \sum_{i=0}^{M} |w_i| \tag{5.24}$$

If η is sufficiently large, some w_i's will equal zero without the use of a threshold [5].

To make predictions in a fully Bayesian framework we define a predictive distribution over new traits t by marginalizing over $\mathbf{w}$ and hyperparameters $\boldsymbol{\alpha}$, β and λ.

$$p(t|\mathbf{t}) = \int\int\int\int p(t|\mathbf{w},\beta,\lambda)p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha},\beta,\lambda)p(\boldsymbol{\alpha},\beta,\lambda|\mathbf{t})\, d\mathbf{w}\, d\boldsymbol{\alpha}\, d\beta\, d\lambda \tag{5.25}$$

However, full marginalization over all these variables is analytically intractable. We can either use sampling methods to build up an estimate of the predictive distribution or make an analytical approximation. Here, we choose the latter alternative by making use of the *evidence approximation* [58] and for the derivation of the PhyRVM we have made use of the derivation of the evidence approximation for Bayesian linear regression given by Bishop [5] and the derivation of the RVM given by Fletcher [25].

If the posterior over $(\boldsymbol{\alpha},\beta,\lambda)$ is sharply peaked around fixed values $(\hat{\boldsymbol{\alpha}},\hat{\beta},\hat{\lambda})$, which will be the case for sufficiently large data sets when the posterior is Gaussian and the fixed values approach the true values, then we can plug those fixed values into the predictive distribution (5.25) so we are left with a marginalization over only $\mathbf{w}$ which is analytically tractable. In order to compute these fixed values, we maximise the posterior over $(\boldsymbol{\alpha},\beta,\lambda)$:

$$p(\boldsymbol{\alpha},\beta,\lambda|\mathbf{t}) \approx p(\mathbf{t}|\boldsymbol{\alpha},\beta,\lambda)p(\boldsymbol{\alpha},\beta,\lambda) \tag{5.26}$$

By assuming the prior is flat, we can maximise the posterior by maximising the marginal likelihood $p(\mathbf{t}|\boldsymbol{\alpha},\beta,\lambda)$ and thus determine the optimal fixed values $(\hat{\boldsymbol{\alpha}},\hat{\beta},\hat{\lambda})$. There is no

closed form solution for $(\boldsymbol{\alpha}, \beta, \lambda)$, so we resort to iterative re-estimation via a set of update equations. The λ^{new} which maximises the evidence will not necessarily be the same as the λ_{ML} which maximises the likelihood. The likelihood cannot be used for model selection without any regularization to prevent overfitting as it directly measures the fit (in terms of sum of squared errors) to the data. On the other hand, as we showed in chapter 2, one of the main advantages of using a Bayesian framework is to use the evidence for model selection.

In the evidence framework, the marginal likelihood $p(\mathbf{t}|\boldsymbol{\alpha}, \beta, \lambda)$ can be evaluated by marginalizing over $\mathbf{w}$:

$$p(\mathbf{t}|\boldsymbol{\alpha}, \beta, \lambda) = \int p(\mathbf{t}|\mathbf{w}, \beta, \lambda) p(\mathbf{w}|\boldsymbol{\alpha})\, d\mathbf{w} \tag{5.27}$$

$$= \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}} \left(\frac{1}{2\pi}\right)^{\frac{M+1}{2}} |\mathbf{V}_\lambda|^{-\frac{1}{2}} \prod_{i=0}^{M} \alpha_i^{\frac{1}{2}} \int \exp(-E(\mathbf{w})) \tag{5.28}$$

$$E(\mathbf{w}) = \beta E_\lambda(\mathbf{w}) + \frac{1}{2}\mathbf{w}^T \mathbf{A} \mathbf{w} \tag{5.29}$$

where $\mathbf{A}$ is a diagonal matrix of α_i's and $E(\mathbf{w})$ is a phylogenetic error function. We can derive a distribution over $\mathbf{w}$ by completing the square:

$$E(\mathbf{w}) = \frac{1}{2}\left(\beta \mathbf{t}^T \mathbf{V}_\lambda^{-1} \mathbf{t} - 2\beta \mathbf{t}^T \mathbf{V}_\lambda^{-1} \mathbf{X}\mathbf{w} + \mathbf{w}^T (\mathbf{A} + \beta \mathbf{X}^T \mathbf{V}_\lambda^{-1} \mathbf{X})\mathbf{w}\right) \tag{5.30}$$

$$= \frac{1}{2}\left(\beta \mathbf{t}^T \mathbf{V}_\lambda^{-1} \mathbf{t} + (\mathbf{w} - \mathbf{m})^T \boldsymbol{\Sigma} (\mathbf{w} - \mathbf{m}) - \mathbf{m}^T \boldsymbol{\Sigma} \mathbf{m}\right) \tag{5.31}$$

$$= E(\mathbf{m}) + \frac{1}{2}(\mathbf{w} - \mathbf{m})^T \boldsymbol{\Sigma} (\mathbf{w} - \mathbf{m}) \tag{5.32}$$

where $\boldsymbol{\Sigma} = \mathbf{A} + \beta \mathbf{X}^T \mathbf{V}_\lambda^{-1} \mathbf{X}$ is the posterior precision matrix and $\mathbf{m} = \beta \boldsymbol{\Sigma}^{-1} \mathbf{X}^T \mathbf{V}_\lambda^{-1} \mathbf{t}$ is the posterior mean. This expression can be used to solve the integral in (5.28):

$$\int \exp(-E(\mathbf{w}))\, d\mathbf{w} = \exp(-E(\mathbf{m}))(2\pi)^{\frac{M+1}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \tag{5.33}$$

Before evaluating the marginal likelihood, we shall rewrite $E(\mathbf{m})$ by completing the square:

$$E(\mathbf{m}) = \frac{1}{2}\left(\beta \mathbf{t}^T \mathbf{V}_\lambda^{-1} \mathbf{t} - \mathbf{m}^T \mathbf{\Sigma} \mathbf{m}\right) \tag{5.34}$$

$$= \beta E_\lambda(\mathbf{m}) + \frac{1}{2}\mathbf{m}^T \mathbf{A} \mathbf{m} \tag{5.35}$$

$$E_\lambda(\mathbf{m}) = \frac{1}{2}(\mathbf{t} - \mathbf{X}\mathbf{m})^T \mathbf{V}_\lambda^{-1}(\mathbf{t} - \mathbf{X}\mathbf{m})$$

Now, we are ready to write the log marginal likelihood:

$$\log p(\mathbf{t}|\boldsymbol{\alpha}, \beta, \lambda) = \frac{N}{2}\log\beta - \frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{V}_\lambda| - \frac{1}{2}\log|\mathbf{\Sigma}| + \frac{1}{2}\sum_i \log\alpha_i - E(\mathbf{m}) \tag{5.36}$$

We can then iteratively maximise (5.36) with respect to $(\boldsymbol{\alpha}, \beta, \lambda)$ until all the hyperparameters converge to their fixed values. By differentiating the log marginal likelihood for each α_i, setting the derivatives equal to zero and rearranging for α_i yields [112] [25]:

$$\frac{\partial}{\partial \alpha_i}\log p(\mathbf{t}|\boldsymbol{\alpha}, \beta, \lambda) = \frac{1}{2\alpha_i} - \frac{1}{2}m_i^2 - \frac{1}{2}\Sigma_{ii}^{-1} = 0 \tag{5.37}$$

$$\alpha_i^{new} = \frac{1 - \alpha_i \Sigma_{ii}^{-1}}{m_i^2} \tag{5.38}$$

The derivative with respect to β is found to be:

$$\frac{\partial}{\partial \beta}\log p(\mathbf{t}|\boldsymbol{\alpha}, \beta, \lambda) = \frac{1}{2}\left(\frac{N}{\beta} - (\mathbf{t} - \mathbf{X}\mathbf{m})^T \mathbf{V}_\lambda^{-1}(\mathbf{t} - \mathbf{X}\mathbf{m}) - \mathrm{Tr}[\mathbf{\Sigma}^{-1}\mathbf{X}^T \mathbf{V}_\lambda^{-1}\mathbf{X}]\right) = 0 \tag{5.39}$$

Simplifying the expression in the trace [25]:

$$\mathbf{\Sigma}^{-1}\mathbf{X}^T \mathbf{V}_\lambda^{-1}\mathbf{X} = \mathbf{\Sigma}^{-1}\mathbf{X}^T \mathbf{V}_\lambda^{-1}\mathbf{X} + \beta^{-1}\mathbf{\Sigma}^{-1}\mathbf{A} - \beta^{-1}\mathbf{\Sigma}^{-1}\mathbf{A} \tag{5.40}$$

$$= \mathbf{\Sigma}^{-1}\left(\beta \mathbf{X}^T \mathbf{V}_\lambda^{-1}\mathbf{X} + \mathbf{A}\right)\beta^{-1} - \beta^{-1}\mathbf{\Sigma}^{-1}\mathbf{A} \tag{5.41}$$

$$= \left(\mathbf{I} - \mathbf{\Sigma}^{-1}\mathbf{A}\right)\beta^{-1} \tag{5.42}$$

Plugging (5.42) into (5.39) and rearranging for β yields:

$$\beta^{new} = \frac{N - \mathrm{Tr}[\mathbf{I} - \mathbf{\Sigma}^{-1}\mathbf{A}]}{2E_\lambda(\mathbf{m})} \tag{5.43}$$

The α_i^{new} update is equivalent to the corresponding RVM update and the β update incorporates the phylogenetic covariance matrix but otherwise is also the same. The update we have derived for λ^{new} is novel and has not appeared in the literature. First, we shall rewrite the log marginal likelihood in terms of V_D:

$$\log p(\mathbf{t}|\boldsymbol{\alpha},\beta,\lambda) = \frac{N}{2}\log\beta - \frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\lambda\mathbf{V}_D| - \frac{1}{2}\log|\boldsymbol{\Sigma}_D| + \frac{1}{2}\sum_i \log\alpha_i - \frac{\beta}{\lambda}E_D(\mathbf{m}) - \frac{\beta^2}{2\lambda^2}\mathbf{m}_D^T\mathbf{A}\mathbf{m}_D$$

$$E_D(\mathbf{m}) = \frac{1}{2}(\mathbf{t}-\mathbf{X}\mathbf{m})^T\mathbf{V}_D^{-1}(\mathbf{t}-\mathbf{X}\mathbf{m})$$

where $\boldsymbol{\Sigma}_D = \mathbf{A} + \frac{\beta}{\lambda}\mathbf{X}^T\mathbf{V}_D^{-1}\mathbf{X}$ is the skewed posterior precision matrix and $\mathbf{m}_D = \boldsymbol{\Sigma}^{-1}\mathbf{X}^T\mathbf{V}_D^{-1}\mathbf{t}$ is the skewed posterior mean. The derivative of the log marginal likelihood with respect to λ is given by:

$$\frac{\partial}{\partial\lambda}\log p(\mathbf{t}|\boldsymbol{\alpha},\beta,\lambda) = \frac{1}{2}\left(\frac{2\beta}{\lambda^2}E_D(\mathbf{m}) + \frac{\beta}{\lambda^2}\mathrm{Tr}[\boldsymbol{\Sigma}_D^{-1}\mathbf{X}^T\mathbf{V}_D^{-1}\mathbf{X}] + \frac{2\beta^2}{\lambda^3}\mathbf{m}_D^T\mathbf{A}\mathbf{m}_D - \frac{N}{\lambda}\right) = 0$$

Simplifying the expression in the trace:

$$\boldsymbol{\Sigma}_D^{-1}\mathbf{X}^T\mathbf{V}_D^{-1}\mathbf{X} = \boldsymbol{\Sigma}_D^{-1}\mathbf{X}^T\mathbf{V}_D^{-1}\mathbf{X} + \frac{\lambda}{\beta}\boldsymbol{\Sigma}_D^{-1}\mathbf{A} - \frac{\lambda}{\beta}\boldsymbol{\Sigma}_D^{-1}\mathbf{A} \tag{5.44}$$

$$= \boldsymbol{\Sigma}_D^{-1}\left(\mathbf{A} + \frac{\beta}{\lambda}\mathbf{X}^T\mathbf{V}_D^{-1}\mathbf{X}\right)\frac{\lambda}{\beta} - \frac{\lambda}{\beta}\boldsymbol{\Sigma}_D^{-1}\mathbf{A} \tag{5.45}$$

$$= \left(\mathbf{I} - \boldsymbol{\Sigma}_D^{-1}\mathbf{A}\right)\frac{\lambda}{\beta} \tag{5.46}$$

Plugging (5.46) back into the trace and rearranging for λ gives a quadratic equation in λ:

$$\lambda^2\left(\mathrm{Tr}[\mathbf{I} - \boldsymbol{\Sigma}_D^{-1}\mathbf{A}] - N\right) + \lambda\left(2\beta E_D(\mathbf{m})\right) + 2\beta^2\mathbf{m}_D^T\mathbf{A}\mathbf{m}_D = 0 \tag{5.47}$$

By employing the quadratic formula we can derive the update equation for Pagel's λ in the PhyRVM:

$$\lambda^{new} = \frac{-2\beta E_D(\mathbf{m}) \pm \sqrt{4\beta^2 E_D^2(\mathbf{m}) - 8\beta^2\left(\mathrm{Tr}[\mathbf{I} - \boldsymbol{\Sigma}_D^{-1}\mathbf{A}] - N\right)\mathbf{m}_D^T\mathbf{A}\mathbf{m}_D}}{2\left(\mathrm{Tr}[\mathbf{I} - \boldsymbol{\Sigma}_D^{-1}\mathbf{A}] - N\right)} \tag{5.48}$$

If the value of Pagel's λ is not independent of β^{new} then we get a simplified formula:

$$\lambda^{new} = \frac{\left(\mathrm{Tr}[\mathbf{I} - \boldsymbol{\Sigma}^{-1}\mathbf{A}] - N\right)}{\left(\mathrm{Tr}[\mathbf{I} - \boldsymbol{\Sigma}_D^{-1}\mathbf{A}] - N\right)}\lambda^{old} \pm \frac{\sqrt{E_D^2(\mathbf{m}) - 2\left(\mathrm{Tr}[\mathbf{I} - \boldsymbol{\Sigma}_D^{-1}\mathbf{A}] - N\right)\mathbf{m}_D^T\mathbf{A}\mathbf{m}_D}}{\left(\mathrm{Tr}[\mathbf{I} - \boldsymbol{\Sigma}_D^{-1}\mathbf{A}] - N\right)}\beta^{new} \tag{5.49}$$

To gain a better understanding of the hyperparameter updates, we can write $\gamma_i = 1 - \alpha_i \Sigma_{ii}^{-1}$. Notice, γ_i appears in all PhyRVM hyperparameter updates (5.38), (5.43), (5.49). The value of $\gamma_i \in [0,1]$ measures how well the corresponding $\mathbf{w}_i$ is determined by the data [5] and $\sum_i \gamma_i$ measures how well the full model is determined by the data. Therefore, when α_i is very large, γ_i will be very small, implying $\mathbf{w}_i$ is not well determined by the data and so the $i$the feature is irrelevant. By using a threshold on α_i we can prune irrelevant features with $\alpha_i \to \infty$. Similarly, we can plot α_i^{-1}, called 'relevance vectors', in a relevance vector plot. The update for Pagel's λ (5.49) contains two terms and the term on the left measures how well the weights $\mathbf{w}$ are determined by the data under the $\mathbf{V}_\lambda$ model in the numerator and the $\mathbf{V}_D$ model in the denominator and the term on the right is the phylogenetic correction.

Learning the PhyRVM requires iterating the hyperparameter updates (5.38), (5.43) and (5.48) or (5.49) while updating the posterior statistics $(\mathbf{m}, \mathbf{\Sigma})$ and the skewed posterior statistics $(\mathbf{m}_D, \Sigma_D)$ until the evidence (5.36) converges. The λ^{new} update is dependent on the initial value of λ^{old} and so unlike the other hyperparameters, we must optimize for the initial value of λ^{old} with a numerical optimisation method such as Brent's method [9]. At first, it seems we've replaced one optimization for another. However, as the PhyRVM is an iterative algorithm, learning each hyperparameter depends on learning all of the others so the algorithm will perform better if all the updates are allowed to adapt their learning automatically as necessary rather than keeping λ fixed throughout. The update (5.48) can be performed independently of $\boldsymbol{\alpha}^{new}$ and β^{new}, however update (5.49) must use β^{new}.

To predict the trait t for a new data point, $\mathbf{x}^*$, we use the mean of the predictive distribution with the learned hyperparameters:

$$p(t|\mathbf{t}, \hat{\boldsymbol{\alpha}}, \hat{\beta}, \hat{\lambda}) = \int p(t|\mathbf{w}, \hat{\beta}, \hat{\lambda}) p(\mathbf{w}|\mathbf{t}, \hat{\boldsymbol{\alpha}}, \hat{\beta}, \hat{\lambda}) d\mathbf{w} \tag{5.50}$$

$$= \mathcal{N}(t|\mathbf{Xm}, \beta^{-1}\mathbf{V}_\lambda + \mathbf{X\Sigma}^{-1}\mathbf{X}^T) \tag{5.51}$$

To test how well the PhyRVM can estimate phylogenetic signal, we simulated 10 phylogenies with 200 taxa using a uniform birth-death process and scaled the branch lengths with known $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ using the R package 'geiger' [80]. We then simulated 11 Brownian motion trait variables using the R package 'phytools' [93] to fit the phylogenetic comparative methods to one trait using the other ten. We used

Table 5.1: Average phylogenetic signal (± one standard deviation) estimated by PhyRVM and PGLS with low true values of $\lambda \in \{0, 0.1, 0.2, 0.3, 0.4\}$. The average marginal likelihood (± one standard deviation) is below and root mean square error in parenthesis. The best estimate of λ on average is underlined. The largest evidence is in bold if it also has the better average λ.

λ	0	0.1	0.2	0.3	0.4
PhyRVM-	$\lambda = 0.598 \pm 0.089$ -428 ± 18.9 (2.14 ± 0.192)	0.362 ± 0.276 -429 ± 19.3 (2.13 ± 0.248)	0.425 ± 0.259 -424 ± 20.2 (2.12 ± 0.282)	0.679 ± 0.442 -417 ± 20.2 (2.11 ± 0.318)	$\underline{0.537 \pm 0.208}$ $\mathbf{-411 \pm 21.5}$ (2.09 ± 0.340)
PhyRVM+	$\lambda = \underline{0.057 \pm 0.009}$ -428 ± 18.9 (2.14 ± 0.192)	$\underline{0.033 \pm 0.029}$ -430 ± 21.0 (2.13 ± 0.247)	$\underline{0.048 \pm 0.015}$ -428 ± 24.3 (2.11 ± 0.273)	$\underline{0.047 \pm 0.018}$ -424 ± 24.9 (2.09 ± 0.294)	0.051 ± 0.015 -418 ± 28.4 (2.06 ± 0.310)
PGLS	$\lambda = 0.500 \pm 0.00$ (2.13 ± 0.192)	0.501 ± 0.506 (2.12 ± 0.252)	0.706 ± 0.367 (2.12 ± 0.286)	0.762 ± 0.339 (2.11 ± 0.317)	0.821 ± 0.241 (2.10 ± 0.348)

Table 5.2: Average phylogenetic signal (± one standard deviation) estimated by PhyRVM and PGLS with high true values of $\lambda \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. The average marginal likelihood (± one standard deviation) is below and root mean square error in parenthesis. The best estimate of λ on average is underlined. The largest evidence is in bold if it also has the better average λ.

λ	0.5	0.6	0.7	0.8	0.9
PhyRVM-	$\lambda = \underline{0.663 \pm 0.137}$ $\mathbf{-404 \pm 16.6}$ (2.08 ± 0.371)	$\underline{0.560 \pm 0.246}$ $\mathbf{-396 \pm 15.5}$ (2.04 ± 0.404)	$\underline{0.715 \pm 0.143}$ $\mathbf{-380 \pm 14.1}$ (2.03 ± 0.411)	$\underline{0.665 \pm 0.164}$ $\mathbf{-359 \pm 16.5}$ (1.98 ± 0.421)	$\underline{0.859 \pm 0.190}$ $\mathbf{-343 \pm 14.0}$ (1.95 ± 0.416)
PhyRVM+	$\lambda = 0.051 \pm 0.011$ -414 ± 25.8 (2.03 ± 0.319)	0.051 ± 0.011 -408 ± 26.5 (1.97 ± 0.311)	0.046 ± 0.015 -396 ± 26.7 (1.92 ± 0.318)	0.052 ± 0.007 -388 ± 23.9 (1.85 ± 0.301)	0.049 ± 0.008 -372 ± 23.8 (1.76 ± 0.263)
PGLS	$\lambda = 0.867 \pm 0.176$ (2.09 ± 0.378)	0.902 ± 0.135 (2.08 ± 0.407)	0.930 ± 0.102 (2.07 ± 0.436)	0.955 ± 0.068 (2.05 ± 0.464)	0.977 ± 0.032 (2.04 ± 0.488)

PGLS to estimate the maximum likelihood value of λ and the PhyRVM with (5.49) to estimate the maximum evidence value of λ. There are two solutions for the maximum evidence λ due to the square root in the quadratic formula. We consider both solutions where 'PhyRVM-' uses the negative phylogenetic correction and 'PhyRVM+' uses the positive phylogenetic correction. The result are displayed in Tables 5.1 and 5.2.

PGLS greatly overestimated the phylogenetic signal in every test. The average PGLS λ's get more accurate as the true value increases. In fact, when true value is less than 0.7, the average PGLS λ can be improved by subtracting 0.5. This is most striking

when the true λ is 0 because the PGLS λ is 0.5. The accuracy of the two PhyRVM λ's is distinguished by the amount of true phylogenetic signal. When the true $\lambda < 0.4$, PhyRVM+ gave the better λ on average and when the true $\lambda \geq 0.4$, PhyRVM- gave the better λ on average with a higher average evidence than PhyRVM+. Except for $\lambda = 0$, PhyRVM- has a higher average evidence than PhyRVM+. The lower likelihood solution is missing for PGLS because there is no analytical solution for the maximum likelihood λ. Therefore, maximum likelihood and Bayesian model comparison for λ will be more useful on average when the true phylogenetic signal is high. However, none of the differences in maximum marginal log likelihood are statistically significant using a Wilcoxon [122] paired signed rank test at the 1% level. Use of PhyRVM- exclusively will lead to potentially high Type I errors (although not as high as PGLS) but low Type II errors. That is, high numbers of false positives but low numbers of false negatives. This problem can be partially mitigated by applying a suitable threshold on the estimates of λ to be regarded as significant. From this study, it would seem that a significant amount of phylogenetic signal should be $\lambda \geq 0.6$ for PhyRVM- and $\lambda \geq 0.9$ for PGLS, although this will increase the number of Type II errors and further studies on real and simulated data will be necessary to nail down an appropriate threshold. It is worth noting that the low phylogenetic signal model PhyRVM+ has a lower RMSE than PhyRVM- in all true λ settings and the improvement gets larger as the true λ increases. This is less surprising than it seems at first. The calculation of the RMSE is not weighted by a phylogeny so maximising the independent data likelihood will still minimise the RMSE of traits even if simulated on a phylogeny. This suggests that phylogenetic regression models will not predict as well as classical linear regression models. To test this we performed a cross validation study on a real dataset.

We compared PhyRVM-, PhyRVM+ and PGLS [14] against their classical counterparts RVM and Ordinary Least Squares (OLS) respectively by predicting the *optimal growth temperature* (OGT) of 209 species of archaea using input data given by the proportions of each of the 20 amino acids in the proteomes of each archaeal species. By reducing the entire archaeal genomes to the 20 amino acid proportions, we can afford a massive computational and conceptual simplification without a massive loss of information (because the information lost is only non-protein-coding DNA). However, the 20 amino acid proportions are limited by treating each amino acid individually, when in fact the order they appear is as important as their quantity. In section 5.1.4, we will also add in the dipeptide proportions which constitute the 400 possible pairs of amino acids. We used

Table 5.3: Prediction of archaeal OGT using 20 amino acid proportions. 10-Fold cross validation error (Test RMSE), training error (RMSE), marginal likelihood (p(D)), (average) predictive log likelihood (Pred log-lik) and phylogenetic signal (λ) for the PhyRVM-, PhyRVM+, RVM, PGLS and OLS models.

	PhyRVM-	PhyRVM+	RVM	PGLS	OLS
λ	0.056	-0.004	0	0.653	0
p(D)	-671.4	-685.3	-691.6	NA	NA
RMSE	5.41	5.38	5.25	6.73	5.21
Test RMSE	5.97 ± 1.03	5.81 ± 1.11	5.77 ± 1.19	6.84 ± 1.41	5.68 ± 1.15
Pred log-lik	-67.8 ± 4.18	-67.5 ± 4.36	0.037 ± 0.085	NA	NA

Brent's method [68] to optimize the initial value of λ^{old} in the PhyRVM and L-BFGS-B [15] to optimize λ in PGLS. For PGLS, we had to add a very small jitter constant in the $\mathbf{w}_{ML}$ update to prevent singularity. The results are shown in Table 5.3. The genomic features, traits and phylogeny used for this analysis were prepared by Edmund Moody[1]. The phylogeny was inferred using IQTREE [70], the sequences were downloaded from NCBI refseq [109] and the OGTs came from Sauer and Wang [98]. PGLS has overestimated the phylogenetic signal and reduced its capacity to fit the training data. Hence, it has the highest training and cross validation error on average. PhyRVM finds the phylogenetic signal is very low in this dataset. PhyRVM- finds that λ is positive and PhyRVM+ finds that λ is negative. A negative value of λ is difficult to interpret but could suggest a convergent instead of divergent evolution because the branch lengths represent the average number of amino acid substitutions per site along the branch [3]. PhyRVM+ has a lower RMSE than PhyRVM- but this is unlikely due to the sign of λ and more likely due to its smaller magnitude. The PhyRVM has a higher evidence than the RVM but also a higher RMSE. The predictive log likelihood for the PhyRVM is significantly lower than the RVM. This is because the phylogenetic covariance matrix is included in the predictive distribution for the PhyRVM (5.51). Ordinary least squares has the lowest training and cross validation RMSE which is not surprising as it is minimising the sum of square errors. However, we tested the statistical significance of the results using a Wilcoxon [122] paired rank sign test at the 1% level and found none of the cross validation results to be significant.

[1] A PhD student in Tom Williams' lab at the University of Bristol.

5.1.2 Phylogenetic Probabilistic Principal Components Analysis (P3CA)

Principal components analysis (PCA) is a widely used dimensionality reduction technique in which a D-dimensional dataset is linearly projected onto a subspace of lower dimension, M. PCA is commonly used in comparative biology to reduce the dimension of a multivariate dataset to a handful of independent components for use in phylogenetic regression analyses. However, if the components of PCA are not phylogenetically independent, systematic errors can occur in the phylogenetic regressions [116]. Phylogenetic PCA (pPCA) [91] has been developed as an improvement to PCA for comparative data by incorporating an inverse phylogenetic covariance matrix into the data covariance matrix producing phylogenetically independent components [91]. However, pPCA does not use a likelihood function, so it cannot be used to find the optimal value of Pagel's λ. Here, we present a novel algorithm called Phylogenetic Probabilistic Components Analysis (P3CA) which incorporates a phylogenetic covariance matrix into a matrix Gaussian latent variable model and can be solved by maximum likelihood. Advantages of P3CA are: multivariate data can be naturally handled by the matrix Gaussian distribution, Pagel's λ can be optimised by maximising the likelihood, P3CA can be extended to use the EM-algorithm for diagonal covariances (factor analysis [5]) and ARD via the evidence approximation. P3CA is an extension of probabilistic PCA [113] for non-independent data points and it is not limited to phylogenetic data. Any positive semi-definite kernel matrix can be used, but for this chapter we will focus on phylogenetic covariance matrices.

We can define a matrix Gaussian [62] prior distribution over the M x N dimensional latent variable $\mathbf{Z}$:

$$p(\mathbf{Z}) = \mathcal{N}(\mathbf{Z}|\mathbf{0},\mathbf{I},\mathbf{V}_\lambda) \tag{5.52}$$

$$= \frac{|\mathbf{V}_\lambda|^{M/2}}{(2\pi)^{N/2}} \exp\left(-\frac{1}{2}\mathrm{Tr}\left[\mathbf{Z}^T\mathbf{Z}\mathbf{V}_\lambda\right]\right) \tag{5.53}$$

Similarly, we can also define the matrix Gaussian conditional distribution of the D x N dimensional data $\mathbf{X}$ given $\mathbf{Z}$:

$$p(\mathbf{X}|\mathbf{Z}) = \mathcal{N}(\mathbf{X}|\mathbf{W}\mathbf{Z}+\mathbf{\Lambda},\sigma^2\mathbf{I},\mathbf{V}_\lambda) \tag{5.54}$$

$$= \frac{|\mathbf{V}_\lambda|^{D/2}}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{1}{2\sigma^2}\mathrm{Tr}\left[(\mathbf{X}-(\mathbf{W}\mathbf{Z}+\mathbf{\Lambda}))^T(\mathbf{X}-(\mathbf{W}\mathbf{Z}+\mathbf{\Lambda}))\mathbf{V}_\lambda\right]\right) \tag{5.55}$$

where $\mathbf{W}$ is a D x M dimensional transformation matrix and Λ is a D x N dimensional mean matrix. The log joint distribution $\mathbf{Y} = (\mathbf{Z}\ \ \mathbf{X})$ can be written as:

$$\log p(\mathbf{Y}) = \log p(\mathbf{Z}) + \log p(\mathbf{X}|\mathbf{Z}) \tag{5.56}$$

$$= -\frac{1}{2}\mathrm{Tr}\left[\mathbf{Z}^T\mathbf{Z}\mathbf{V}_\lambda\right] - \frac{1}{2}\mathrm{Tr}\left[(\mathbf{X}-\mathbf{W}\mathbf{Z}-\Lambda)^T\beta\mathbf{I}(\mathbf{X}-\mathbf{W}\mathbf{Z}-\Lambda)\mathbf{V}_\lambda\right] + \text{const} \tag{5.57}$$

where $\beta = \sigma^{-2}$ and const represents terms that don't include $\mathbf{Z}$ or $\mathbf{X}$. Consider the exponent in a general matrix Gaussian distribution $\mathcal{N}(\mathbf{Y}|\boldsymbol{\mu}, \Sigma_1, \Sigma_2)$:

$$-\frac{1}{2}(\mathbf{Y}-\boldsymbol{\mu})^T\Sigma_1^{-1}(\mathbf{Y}-\boldsymbol{\mu})\Sigma_2 = -\frac{1}{2}\mathbf{Y}^T\Sigma_1^{-1}\mathbf{Y}\Sigma_2 + \mathbf{Y}\Sigma_1^{-1}\boldsymbol{\mu}\Sigma_2 + \text{const} \tag{5.58}$$

Notice, the coefficient of the second-order terms is Σ_1^{-1} and the coefficient of the linear terms is $\Sigma_1^{-1}\boldsymbol{\mu}$ (ignoring Σ_2 which can be factored separately). We can factor the second-order terms in (5.57) to get the precision matrix $\mathrm{cov}[\mathbf{Y}]^{-1}$ and then use the matrix inversion formula [5] to find the covariance of $\mathbf{Y}$:

$$\mathrm{cov}[\mathbf{Y}]^{-1} = \begin{pmatrix} \mathbf{I} + \mathbf{W}^T\beta\mathbf{I}\mathbf{W} & -\mathbf{W}^T\beta\mathbf{I} \\ -\beta\mathbf{I}\mathbf{W} & \beta\mathbf{I} \end{pmatrix} \tag{5.59}$$

$$\mathrm{cov}[\mathbf{Y}] = \begin{pmatrix} \mathbf{I} & \mathbf{W}^T \\ \mathbf{W} & \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I} \end{pmatrix} \tag{5.60}$$

Similarly, we can factor the first-order terms in (5.57) and multiply by (5.60) to get the expectation of $\mathbf{Y}$:

$$\mathbb{E}[\mathbf{Y}] = \begin{pmatrix} \mathbf{0} \\ \Lambda \end{pmatrix} \tag{5.61}$$

By marginalizing over the latent variables and using (5.60) and (5.61) we can write the marginal distribution $p(\mathbf{X})$:

$$p(\mathbf{X}) = \int p(\mathbf{X}|\mathbf{Z})p(\mathbf{Z})\ d\mathbf{Z} \tag{5.62}$$

$$= \mathcal{N}\left(\mathbf{X}|\mathbb{E}[\mathbf{X}], \mathrm{cov}_1[\mathbf{X}], \mathrm{cov}_2[\mathbf{X}]\right) \tag{5.63}$$

$$\mathbb{E}[\mathbf{X}] = \Lambda \tag{5.64}$$

$$\mathrm{cov}_1[\mathbf{X}] = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I} \tag{5.65}$$

$$\mathrm{cov}_2[\mathbf{X}] = \mathbf{V}_\lambda \tag{5.66}$$

When $\lambda = 0$ and the root to tip distance is 1, then the distribution $p(\mathbf{X})$ in P3CA is equivalent to PPCA. We will require $\text{cov}_1[\mathbf{X}]$ for the posterior over latent variables. The precision matrix (inverse covariance) can be found using the Woodbury identity [81] to give:

$$\text{cov}_1[\mathbf{X}]^{-1} = \sigma^{-2}\mathbf{I} - \sigma^{-2}\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^T \tag{5.67}$$

$$\mathbf{M} = \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I} \tag{5.68}$$

This has the added benefit of reducing an $O(D^3)$ inversion of $\text{cov}_1[\mathbf{X}]$ to an $O(M^3)$ inversion of $\mathbf{M}$. By using (5.60) and (5.61) and the standard formula for conditional Gaussians [5], the posterior $p(\mathbf{Z}|\mathbf{X})$ can be written as:

$$p(\mathbf{Z}|\mathbf{X}) = \mathcal{N}(\mathbf{Z}|\mathbb{E}(\mathbf{Z}|\mathbf{X}), \text{cov}_1(\mathbf{Z}|\mathbf{X}), \text{cov}_2(\mathbf{Z}|\mathbf{X})) \tag{5.69}$$

$$\mathbb{E}(\mathbf{Z}|\mathbf{X}) = \mathbf{M}^{-1}\mathbf{W}^T(\mathbf{X} - \Lambda) \tag{5.70}$$

$$\text{cov}_1(\mathbf{Z}|\mathbf{X}) = \sigma^2\mathbf{M}^{-1} \tag{5.71}$$

$$\text{cov}_2(\mathbf{Z}|\mathbf{X}) = \mathbf{V}_\lambda \tag{5.72}$$

Once again, with $\lambda = 0$ and root to tip distance equal to 1, the posterior of P3CA is equivalent to PPCA. The posterior mean (5.70) corresponds to the mapping from the original data space to the lower dimensional latent space. Now, the foundation for P3CA has been laid, we can learn the parameters $(\mathbf{W}_{ML}, \sigma^2_{ML}, \lambda_{ML})$ by maximising the (log) likelihood:

$$\begin{aligned}\log p(\mathbf{X}|\mathbf{W}, \Lambda, \sigma^2, \lambda) = \frac{D}{2}\log|\mathbf{V}_\lambda| - \frac{ND}{2}\log(2\pi) - \frac{N}{2}\log|\mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}| \\ - \frac{1}{2}\text{Tr}\left[\mathbf{X}^T(\mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I})^{-1}\mathbf{X}\mathbf{V}_\lambda\right]\end{aligned} \tag{5.73}$$

Note, the mean Λ is not definable for a single data matrix $\mathbf{X}$. There are two options: either the data matrix should be split into smaller, equally sized matrices or vectors representing individual samples. Here, we choose the latter by using the sample mean which is equivalent to normalising the data and setting $\Lambda = \mathbf{0}$. Applying the 'trace trick' to the log-likelihood (5.73) yields a simplified expression in terms of a phylogenetically weighted sample covariance $\mathbf{S}_\lambda$:

$$\log p(\mathbf{X}|\mathbf{W}, \sigma^2, \lambda) = \frac{D}{2}\log|\mathbf{V}_\lambda| - \frac{N}{2}\left(D\log(2\pi) + \log|\mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}| + \text{Tr}\left[\mathbf{S}_\lambda(\mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I})^{-1}\right]\right)$$

$$\mathbf{S}_\lambda = \frac{1}{N}\mathbf{X}\mathbf{V}_\lambda\mathbf{X}^T$$

$$\frac{\partial}{\partial\mathbf{W}}\log p(\mathbf{X}|\mathbf{W}, \sigma^2, \lambda) = N(\mathbf{C}^{-1}\mathbf{S}_\lambda\mathbf{W} - \mathbf{C}^{-1}\mathbf{W}) \tag{5.74}$$

where $\mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}$. At the stationary points:

$$\mathbf{S}_\lambda \mathbf{C}^{-1}\mathbf{W} = \mathbf{W} \tag{5.75}$$

A trivial solution is $\mathbf{W} = 0$, which is a minimum of the likelihood. A non-trivial solution exists but it is more difficult to find:

$$\mathbf{S}_\lambda = \mathbf{C}_{ML} \tag{5.76}$$

where $\mathbf{C}_{ML}$ is the maximum likelihood solution of $\mathbf{C}$. Fortunately, the solution is in the same form as PPCA [5], so we can make use of the closed-form solutions derived in [113] by substituting the sample covariance for the phylogenetically weighted sample covariance. The maximum likelihood solution for $\mathbf{W}$ is given by:

$$\mathbf{W}_{ML} = \mathbf{U}_M(\mathbf{L}_M - \sigma^2\mathbf{I})^{1/2} \tag{5.77}$$

where L_M is an M x M dimensional diagonal matrix of the M largest eigenvalues, γ_i, of $\mathbf{S}_\lambda$ and U_M is an D x M matrix with columns given by the corresponding eigenvectors. This result is proved for PPCA in [113]. Assuming the eigenvectors are arranged in order of decreasing eigenvalues, where the M largest eigenvalues are $\gamma_1, ..., \gamma_M$, the maximum likelihood solution for σ^2 is given by:

$$\sigma^2_{ML} = \frac{1}{D-M}\sum_{i=M+1}^{D}\gamma_i \tag{5.78}$$

We can also derive a closed-form solution for the maximum likelihood estimate of Pagel's λ using the representation (5.14) to rewrite the log likelihood in terms of V_D:

$$\frac{\partial}{\partial\lambda}\log p(\mathbf{X}|\mathbf{W},\sigma^2,\lambda) = \frac{ND}{2\lambda} - \frac{N}{2}\mathrm{Tr}\left[\mathbf{S}_D(\mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I})^{-1}\right] \tag{5.79}$$

$$\lambda_{new} = \frac{D}{\mathrm{Tr}\left[\mathbf{S}_D(\mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I})^{-1}\right]} = \lambda^{old} \tag{5.80}$$

$$\mathbf{S}_D = \frac{1}{N}\mathbf{X}\mathbf{V}_D\mathbf{X}^T \tag{5.81}$$

Once again, we see that maximum likelihood is unsuitable for an analytical treatment of Pagel's λ. To assess the effect of phylogenetic signal we projected 20 archaeal and bacterial amino acid proportions onto the first two components using P3CA shown in Figure 5.2 with $\lambda = 0$ (left) and $\lambda = 1$ (right). (Left) The first principal component is

not phylogenetically independent as both archaea and bacteria are widely spread out horizontally. However, the second principal component does show some phylogenetic signal which can be related to traits. As bacteria and archaea are the simplest extant living organisms there are not many traits to measure. One possibility is their optimal growth temperature (OGT) which is the temperature at which their growth rate is fastest under laboratory conditions. It is known that bacteria and archaea OGTs do not follow the same distribution. Bacteria are predominantly psychrophiles (OGT < 10 degrees) and mesophiles (10 degrees ≤ OGT ≤ 50 degrees) and archaea are predominately thermophiles (OGT > 50 degrees). This could explain why Groussin & Galtier [33] found a strong correlation between the 2nd component and prokaryotic OGT. But this weak signal is more an artifact of the fact that the majority of the variance is in the first component. (Right) The components clearly separate archaea from bacteria vertically and horizontally much better and by only using two components we can almost achieve full separation. The places where they overlap could point to mistakes in the phylogenetic inference, i.e. archaea misclassified as bacteria.

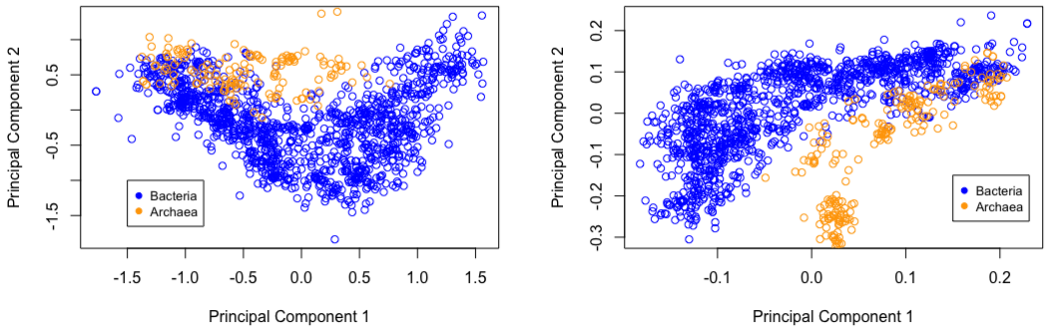


Figure 5.2: The first two principal components of the 20 amino acids proportions of archaea and bacteria using P3CA with $\lambda = 0$ (left) and $\lambda = 1$ (right).

5.2 Are 'Relevant' Genomic Features Correlated with OGT?

The prediction of the optimal growth temperature (OGT) of prokaryotes (archaea and bacteria) from genomic features began with the simple linear regression used by Zeldovich et al. [125] and is now an active field of research with each paper increasing the number

of species and the number of features. Yet, the extremely simple model of Zeldovich et al. is still very appealing. They found that a particular subset of summed amino acid proportions corresponding to Ile, Val, Tyr, Trp, Arg, Glu, Leu (IVYWREL) is most highly correlated with OGT. The significance of summing the amino acid proportions is that it creates an artificial binary classification between IVYWREL and the other 13 amino acids. It is most fortuitous then that this set has real biological significance. The set IVYWREL, called the 'universal set', contains only amino acids which are loaded to tRNA by class I aminoacyl-tRNA synthetases [125]. This is a significant finding and it is interesting to see whether automatic relevance determination (ARD) can be used to identify individual correlates of OGT. For our analysis, we use the classical RVM as the phylogenetic signal in this dataset was found to be insignificant.

Figures 5.3 shows the most relevant whole genome features for an RVM trained on A) 213 archaea (left) and B) 1237 bacteria (right). The most relevant features for archaea and bacteria are AC and CA respectively. These two dinucleotides, in opposite orientations, represent the same information biochemically but neither is strongly correlated with OGT. Figures 5.4 shows the relationship between archaeal dinucleotide AC vs OGT (left) and AG vs OGT (right). The dinucleotide AG is strongly correlated with OGT in archaea and yet it is far less relevant. The scenario this implies is one in which the features which are most correlated with OGT are themselves correlated with each other - as many genomic features incontrovertibly are. This suggests that in the presence of correlated features, the most 'relevant' features are not the ones which are most correlated with the trait. The horizontal line in these plots at 37 degrees represents a known bias in experimental recordings of prokaryotic OGT [31]. The bacteria data is far noisier and there are no strong correlations between any individual feature and OGT.

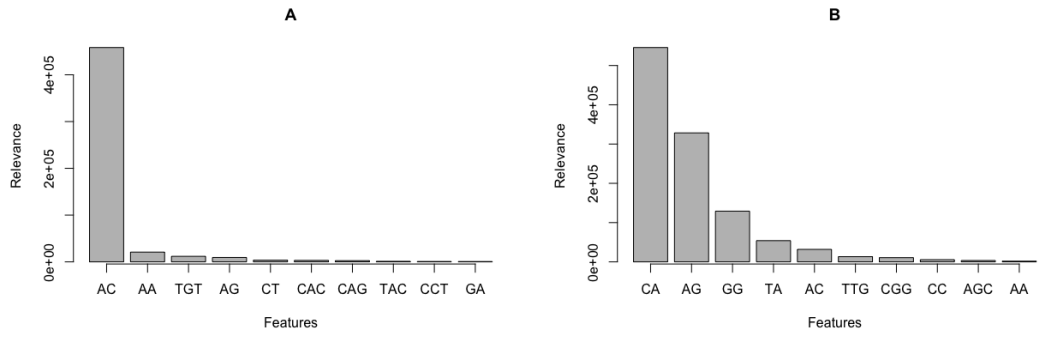


Figure 5.3: Most 'relevant' whole genomic features for archaea (left) and bacteria (right).

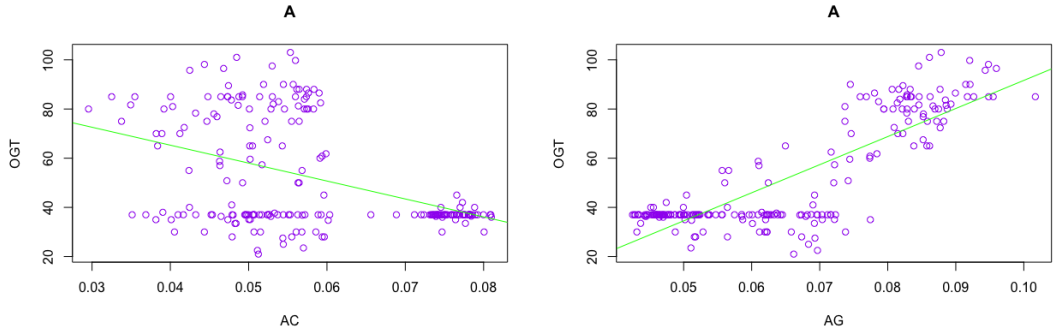


Figure 5.4: Archaeal AC (left) and AG (right) dinucleotide proportion vs OGT.

5.3 Model Comparison for OGT Regression

We want to test if we can train an RVM to be competitive with the state-of-the-art OGT prediction models in the literature. We loaded N Gaussian basis functions over 20 amino acids (where N is the number of species) and 400 dipeptide features for archaea and an additional 104 genome derived features for bacteria into the RVM and let ARD prune out the irrelevant features. The benefit of this approach over a kernel method is that by constraining the model to be linear we are forcing it to find a linear representation in a lower dimensional feature space (instead of finding a linear representation in a higher dimensional feature space). In addition, we also spotted a simple sigmoid relationship between three amino acids (D, Q, T) and one dinucleotide (AG) and archaeal OGT shown in Figure 5.5. We used a logistic regression trained on these four features to classify thermophiles (OGT > 50) from non-thermophiles. The logistic regression achieved 97.7% training accuracy. The square of the prediction probability shows a strong linear relationship with OGT, with adjusted R^2 = 0.9253, so we included it as an additional feature in the archaea model. The positive correlation of dinucleotide AG proportion and OGT has been reported previously in the literature [31] and offers some interpretation. AG dinucleotides contribute to nucleic acid thermostability via base-stacking interactions [31]. The bacterial dataset contains far more noise and we were unable to find any similar simple nonlinear relationships between individual features and bacterial OGT.

We performed an extensive comparison of models from the literature for the prediction of prokaryotic optimal growth temperature using genomic features against our RVM model. The models include the simple linear regression [125], multiple linear regression [98] and Support Vector Regression (SVR) with a Gaussian kernel on 20 amino acids

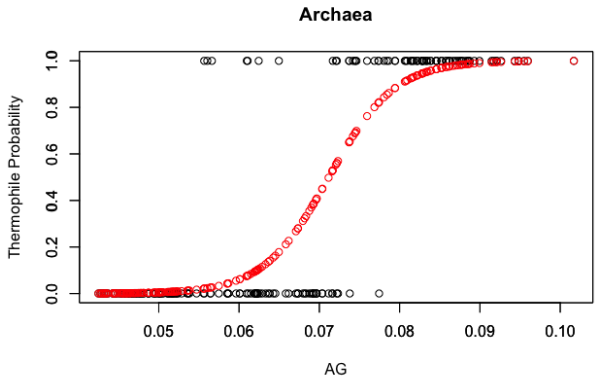


Figure 5.5: Sigmoidal relationship between AG proportion and thermophilic probability.

and 400 dipeptides [29]. We performed 10-fold cross-validation. Nested cross validation was used both to select optimal hyperparameters for SVR and RVM and to evaluate performance. To avoid bias in cross validation, the logistic regression is not trained on the validation data and thermophilic classification is based on only the training data for each fold. The Sauer and Wang bacteria model was computationally singular and could not be trained without a substantial amount of jitter. However, this problem did not occur during cross validation, as the size of the training set was smaller. The results are given in Table 5.4. The best performing models achieved less than 5 degrees cross validation root mean square error. Our RVM model performed the best on the archaea data and only used protein features (amino acids and dipeptides) and one dinucleotide (AG) and the SVR with 400 dipeptides performed the best on the bacteria data (likely because the bacteria data is much noisier which favours the SVR's greater power to fit to the peculiarities of the training data). We tested the statistical significance of the results using a Wilcoxon [122] paired signed rank test in Table 5.5. The RVM only showed a statistically significant improvement over the Zeldovich et al. simple linear regression and the Sauer and Wang multiple linear regression models.

Figures 5.6 shows the predicted OGTs computed during cross validation plotted against the experimental OGTs for archaea (left) and bacteria (right). The archaea RVM approximates the experimental values very well across the full range. For bacteria we see very good performance in the thermophilic range 50-80 degrees. The model is significantly weaker at predicting psychrophiles (OGT < 25). The concept of a psychrophile and its relationship to OGT is not as clear as a mesophile or thermophile. It is hard to say whether these species are environmental psychrophiles or psychrophiles in terms of OGT.

By contrast, the inaccurate prediction on the hyperthermophile at the top of the plot is straightforward as it is the only species of bacteria in our dataset with OGT above 90 degrees.

Table 5.4: Average 10-fold cross validation error (± one standard deviation) and training error in parenthesis for archaea and bacteria OGT prediction using simple and multiple linear regression, SVR and RVM models.

Model	No. of features	Predictive Performance	
		Archaea	Bacteria
Simple Linear Regression	1	7.60 ± 1.33 (7.66)	8.60 ± 0.489 (8.60)
Multiple Linear Regression	22 (A), 23 (B)	5.85 ± 0.651 (5.18)	5.97 ± 0.367 (9.95)
Support Vector Regression	20 amino acids	5.20 ± 0.948 (3.44)	5.33 ± 0.372 (4.00)
Support Vector Regression	400 dipeptides	5.36 ± 1.15 (3.06)	4.97 ± 0.283 (3.27)
Relevance Vector Machine	261 (A), 662 (B)	4.62 ± 0.656 (3.28)	5.17 ± 0.255 (4.21)

Table 5.5: P-values from a Wilcoxon paired signed rank test comparing cross validation RMSE of RVM to Simple Linear Regression (SLR), Multiple Linear Regression (MLR) and Support Vector Regression (SVR). Statistically significant results at the 1% level are shown in bold.

Model	Archaea	Bacteria
RVM:SLR	$\mathbf{4.3 \times 10^{-5}}$	$\mathbf{1.1 \times 10^{-5}}$
RVM:MLR	$\mathbf{3.2 \times 10^{-4}}$	$\mathbf{4.3 \times 10^{-5}}$
RVM:SVR(AA)	3.2×10^{-1}	3.2×10^{-1}
RVM:SVR(Dipeptides)	3.2×10^{-1}	3.2×10^{-1}

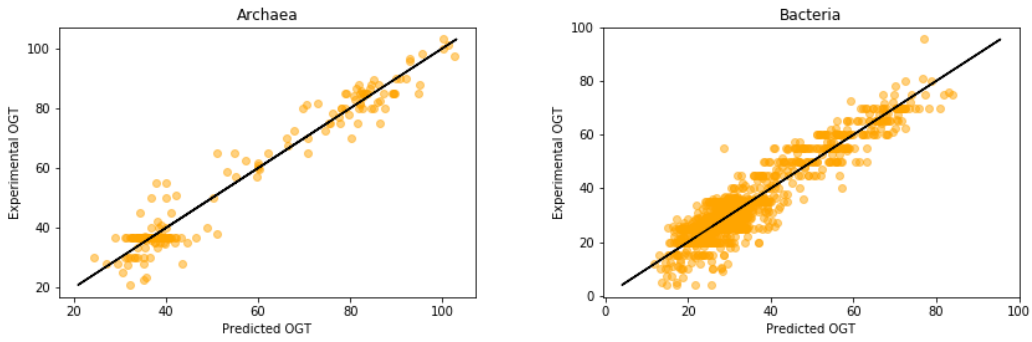


Figure 5.6: Predicted OGT vs Experimental OGT on archaea (left) and bacteria (right) data.

5.4 Ancestral Sequence & OGT Reconstruction

One of the most exciting open questions in microbial ecology is the determination of the conditions under which the hypothetical last universal common ancestor (LUCA) thrived and gave impetus to every extant living organism today. The LUCA is taken to be the root of the prokaryotic phylogeny. We can also define a last archaeal common ancestor (LACA) and last bacterial common ancestor (LBCA) as the two descendants of LUCA. Using phylogenetic inference packages such as RevBayes [47], we can reconstruct the amino acid sequences of these ancestral organisms. We can then derive genomic features for the ancestors and predict OGTs using our RVM model trained on extant taxa. To reconstruct the ancestral sequences we implemented a branch-heterogeneous amino acid substitution model in RevBayes [47] with a relaxed molecular clock model using Markov Chain Monte Caro (MCMC). The branch-heterogeneous model was based on the node-discrete compositional heterogeneity (NDCH) model [26], where each interior node of the phylogeny is fit to one of a finite set of GTR (Generalised Time Reversible) [110] composition vectors with replacement. The overall model is a mixture of the different GTR models. Ideally, we would use a separate GTR model for every branch but this would be computationally impractical for even a modest number of species. If we can reduce the number of mixtures without sacrificing the quality of reconstructions then we can potentially scale the method to use larger numbers of species. To determine the effect of the number of mixtures (GTR models) we performed ancestral sequence reconstruction (ASR) with 1, 2, 4, 8 and 16 mixtures. Figure 5.7 shows the root branch lengths (right) and their variances (left) for each number of mixtures used. The LACA root branch is in red and the LBCA root branch is in black. Both LACA and LBCA are equidistant to the root with 1 mixture. The LBCA root branch is shorter than the LACA root branch suggesting LBCA was the first of the two to diverge from LUCA.

Before we apply a regression model to the ancestral species, it is important to know whether the ancestral sequences follow a similar composition to the extant species in the training data. Figure 5.8 shows the two P3CA plots for first 2 principal components of the 20 amino acid proportions with the inclusion of LUCA, LBCA, LACA (in red and circled) reconstructed using 8 mixtures of GTR models. We see that all three ancestors overlap and lie precisely in the middle, where the vertical of the first component meets the horizontal of the second (right). This is where we would expect it to be intuitively as an average of extant archaea and bacteria sequences.

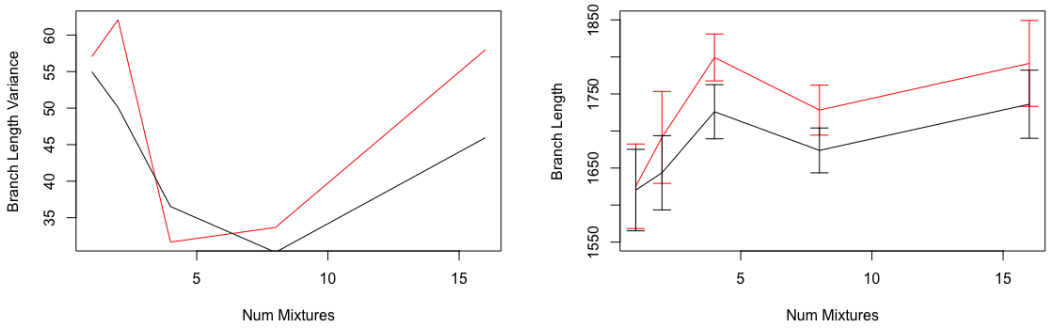


Figure 5.7: Number of GTR mixtures vs branch length variance (left) and branch length (right). The LACA root branch is in red and the LBCA root branch is in black.

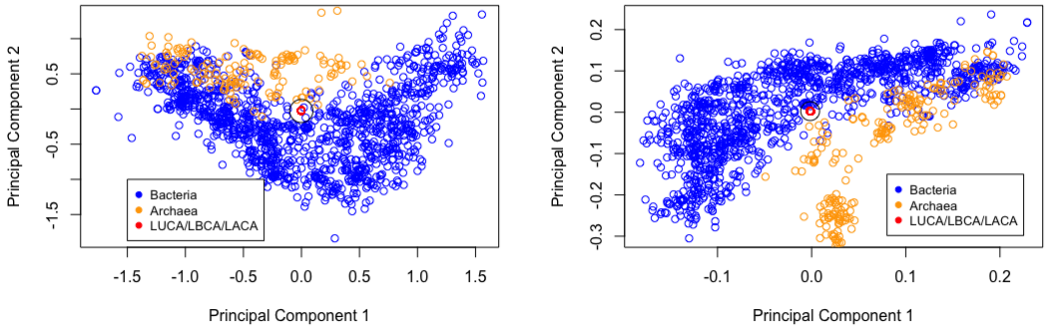


Figure 5.8: P3CA with $\lambda = 0$ (left) and $\lambda = 1$ (right). Archaea is in orange and bacteria is in blue. LUCA, LBCA and LACA, in red and circled, all line up at the same spot in the centre of the plot.

We used an RVM with a Gaussian kernel over the 20 amino acids to predict the OGT of LUCA, LACA and LBCA reconstructed with 1, 2, 4, 8 and 16 mixtures. The OGT of LBCA did not vary significantly over the range of mixtures with a predicted OGT of 51 degrees. Figures 5.9 shows the predicted OGTs of LACA (left) and LUCA (right) for the range of mixtures considered. The predicted OGT of LACA is stable with 4 or more mixtures at 55 degrees. The predicted OGT of LUCA changes significantly with increasing number of mixtures, ranging from 59 degrees with 1 mixture to 90 degrees with 8 mixtures. Therefore, with 8 mixtures our RVM model predicts a thermophilic origin for archaea and bacteria and a hyperthermophilic LUCA. These results are consistent with the phylogenetic analysis [121] suggesting that LUCA contained reverse gyrase - an enzyme only present in hyperthermophiles.

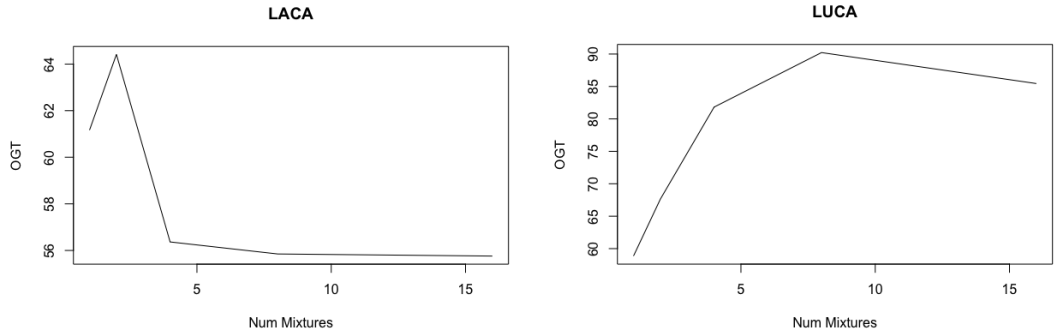


Figure 5.9: Number of GTR mixtures vs OGT for LACA (left) and LUCA (right).

5.5 Concluding Remarks

In this chapter, we introduced an empirical Bayesian approach to estimate phylogenetic signal of a continuous valued trait. While there is no analytic maximum likelihood solution for Pagel's λ, we found that there is a solution for the maximum evidence. We developed the phylogenetic relevance vector machine (PhyRVM) with this analytical update to get more accurate estimates of Pagel's λ by using Bayesian model comparison. We developed Phylogenetic Probabilistic Principal Components Analysis (P3CA) which is a matrix Gaussian latent variable model capable of estimating phylogenetic signal by maximum likelihood.

There is still the problem of optimizing the initial value λ^{old} which will significantly slow down the PhyRVM. Ho and Ané [46] developed an algorithm to speed-up the costly computation of the determinant and inverse of the phylogenetic covariance matrix. This same speed-up can be applied to the PhyRVM.

We built a state-of-the-art archaeal OGT prediction model using the RVM. It appears linear modelling is sufficient to predict archaeal OGT. And yet, by only using protein primary structure information. By including protein secondary structure information, such as the proportions of amino acids in alpha helices, beta sheets and loops, we could potentially improve the model and find out which protein region is most informative, or most irrelevant, using ARD. We also used the RVM to predict the last universal common ancestor (LUCA) was a hyperthermophile which is corroborated by independent studies in the literature [121].

CHAPTER 6

DISCUSSION

"All models are wrong, but some are useful."

— George. E. P. Box

This thesis developed tools for approximating Bayesian inference for independent and phylogenetically dependent data. Classical techniques, like Expectation Propagation (EP) and Evidence Approximation (EA), perform *exact* inference on an approximate posterior. We relaxed this requirement within the EP and EA frameworks by performing *approximate* inference on an approximate posterior, for which exact inference is a special case. The approximation to EP was to incorporate an additive bias term into the posterior natural parameter update. The approximation to EA was to incorporate a phylogenetic covariance matrix into the likelihood function. A parameter was then chosen by maximising the evidence to determine how far the new methods, called γ-EP, PhyRVM and P3CA, should stray from their classical counterparts, called EP, RVM and PPCA respectively.

Chapter 3 derived the spherical Gaussian γ-EP algorithm for the clutter problem, extended it to use Lagrange multipliers, and achieved a statistically significant improvement over ADF and EP in low clutter levels. By maximising the evidence, γ-EP is able to determine the value of γ to find different local maxima to canonical EP which make the data more probable.

Chapter 4 developed the sparse linear Bayes point machine using γ-EP with $\gamma > 0$. Experiments on real data from the UCI database showed that the support vectors found

by γ-EP are comparable to the support vectors found by the SVM in quantity and accuracy of the classifier. The γ-EP modification to the kernel Bayes point machine was presented and combined with multiple kernel learning (MKL) to classify oncogenic single nucleotide variants using several heterogeneous genomic data sources.

Chapter 5 developed the Phylogenetic Relevance Vector Machine (PhyRVM) and Phylogenetic Probabilistic Principal Components Analysis (P3CA) to incorporate phylogenetic non-independence of extant taxa into two classical probabilistic machine learning models. Experiments on simulated data showed that PhyRVM achieves superior estimation accuracy of Pagel's λ to PGLS. A state-of-the-art predictor of archaeal OGT was developed using an RVM. The ancestral OGT of the last universal common ancestor (LUCA) was reconstructed to be a hyperthermophile.

We found the maximum evidence estimate of Pagel's λ to be more accurate than the maximum likelihood estimate in several simulations. It would be interesting to use the PhyRVM to recompute published maximum likelihood estimates of λ and see if any significant hypotheses are affected. More accurate estimates of λ should also be sought either by using variational inference or Markov Chain Monte Carlo methods. An ideal next step would be to derive a similar analytical λ update for the Variational RVM [114]. The PhyRVM presented in this thesis can only predict a single trait. It is also possible to predict multiple traits using a matrix Gaussian distribution [27].

The γ-EP modification sits within the canonical EP framework. Therefore, much of the body of EP research developed since Minka's original publication should be applicable to γ-EP as well. For example, instead of the KL-divergence, we could use the more general α-divergence:

$$D_\alpha(p||q) = \frac{4}{1-\alpha^2}\left(1 - \int_x p(x)^{(1+\alpha)/2} q(x)^{(1-\alpha)/2} dx\right) \tag{6.1}$$

When $\alpha = 1$, $D_\alpha(p||q)$ is equivalent to $KL(p||q)$ as used in canonical EP, and when $\alpha = -1$, $D_\alpha(p||q)$ is equivalent to $KL(q||p)$, as used in variational inference. Minka showed that an extension of EP called *Power EP* [65], in which each approximate factor is raised to a power $1/n_i$, is equivalent to using the α-divergence with $\alpha_i = (2/n_i) - 1$. Power EP has been used to unify [13] several previous sparse Gaussian Process classification methods and furthermore, non standard values of α, such as 0.5, have been found to outperform canonical EP and variational inference. It would be very interesting to test γ-EP using

the Power EP formalism. For example, do the quantity and quality of support vectors change with different alpha divergences?

In this thesis we have presented new methods for approximating posteriors by selecting a parameter to maximise the evidence. For EP, the parameter could push the iterations towards better local maxima. Although, the iterations could become unstable and not converge or even find local minima instead. For EA, the parameter controls how much phylogenetic signal is in the residuals. Although, the inferred phylogeny may not be a good representation of the true evolution of the species. Since the true model will have the maximum evidence on average (see appendix A.1), it is interesting to see that the approximate methods were often favoured.

APPENDIX A

APPENDIX

"Beyond the well-traversed path, mathematics loses its bearings in a jungle of unnamed special functions and impenetrable combinatorial particularities. Thus, the mathematical technique can only reach far if it starts from a point close to the simple essentials of a problem which has simple essentials."

— Jacob. T. Schwarz

A.1 Bayes factors cannot systematically reject the truth

Suppose, we know the true hypothesis $\mathscr{H}_T$ for a set of observed data and we want to check that Bayesian theory gives preference to $\mathscr{H}_T$ over an alternative $\mathscr{H}_A$. Furthermore, suppose (rather absurdly) that we do not use our prior knowledge of the true hypothesis at all by assuming a flat prior. Then, we can compute the ratio of evidences $p(D|\mathscr{H}_T) / p(D|\mathscr{H}_A)$ (called the *Bayes factor* [5]) to determine which hypothesis is most probable. It is possible to invent an example for which the evidence of the incorrect hypothesis is greater even than the true hypothesis [105]. However, by averaging the Bayes factor over the distribution of datasets with respect to the true distribution of the data, we get a quantity called the *Kullback-Leibler divergence*:

$$\int p(D|\mathscr{H}_T) \log \frac{p(D|\mathscr{H}_T)}{p(D|\mathscr{H}_A)} \, dD \tag{A.1}$$

The KL-divergence is always positive and zero when the two distributions are equal. Therefore, as the natural logarithm of the Bayes factor is positive, the Bayes factor must

be greater than 1, so on average the Bayes factor cannot systematically reject the true hypothesis [58].

A.2 Minimising the Kullback-Leibler divergence in the exponential family

Consider the Kullback-Leibler divergence between any distribution $p(\mathbf{x})$ and an exponential family distribution $q(\mathbf{x})$.

$$f(\boldsymbol{\eta}) = \mathrm{KL}(p||q) = \int p(\mathbf{x}) \log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) d\mathbf{x} \tag{A.2}$$

$$= E_p[\log(p(\mathbf{x}))] + E_p[\log(Z(\boldsymbol{\eta}))] - E_p[\boldsymbol{\eta} T(\mathbf{x})] \tag{A.3}$$

$$= E_p[\log(p(\mathbf{x}))] + \log(Z(\boldsymbol{\eta})) - \boldsymbol{\eta} E_p[T(\mathbf{x})] \tag{A.4}$$

Plugging (2.16) into (A.4) at a minimum:

$$\nabla_{\boldsymbol{\eta}} f(\boldsymbol{\eta}^*) = E_q[T(\mathbf{x})] - E_p[T(\mathbf{x})] = \mathbf{0} \tag{A.5}$$

To show that this solution is indeed a minimum, we take the second derivative of f:

$$\nabla\nabla_{\boldsymbol{\eta}} f(\boldsymbol{\eta}) = \frac{\partial^2 \log(Z(\boldsymbol{\eta}))}{\partial \eta_i \eta_j} = \frac{\partial}{\partial \eta_j} \frac{\int T_i(\mathbf{x}) \exp(\boldsymbol{\eta} T(\mathbf{x})) d\mathbf{x}}{Z(\boldsymbol{\eta})} \tag{A.6}$$

$$= E_q(T_i(\mathbf{x}) T_j(\mathbf{x})) - E_q[T_i(\mathbf{x})] E_q[T_j(\mathbf{x})] \tag{A.7}$$

This is the covariance matrix of $T(\mathbf{x})$ which is positive semi-definite by definition [42].

A.3 Deriving the moment matching updates

Consider approximating any distribution $p(\mathbf{x})$ with a multivariate Gaussian $q(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ such that $p(\mathbf{x}) \propto t(\mathbf{x}) q(\mathbf{x})$ and $Z = \int t(\mathbf{x}) q(\mathbf{x}) d\mathbf{x}$. We will need to take derivatives of $q(\mathbf{x})$ with respect to the mean and variance.

$$\nabla_{\boldsymbol{\mu}} q(\mathbf{x}) = \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}) q(\mathbf{x}) \tag{A.8}$$

$$\mathbf{x} q(\mathbf{x}) = \boldsymbol{\mu} q(\mathbf{x}) + \boldsymbol{\Sigma} \nabla_{\boldsymbol{\mu}} q(\mathbf{x}) \tag{A.9}$$

$$\nabla_{\boldsymbol{\Sigma}} q(\mathbf{x}) = \frac{1}{2}\left(-\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \boldsymbol{\Sigma}^{-1}\right) q(\mathbf{x}) \tag{A.10}$$

$$\mathbf{x}\mathbf{x}^T q(\mathbf{x}) = 2\boldsymbol{\Sigma}(\nabla_{\boldsymbol{\Sigma}} q(\mathbf{x}))\boldsymbol{\Sigma} + \left(\boldsymbol{\Sigma} + \mathbf{x}\boldsymbol{\mu}^T + \boldsymbol{\mu}\mathbf{x}^T - \boldsymbol{\mu}\boldsymbol{\mu}^T\right) q(\mathbf{x}) \tag{A.11}$$

Multiplying both sides of (A.9) and (A.11) by $Z^{-1}t(\mathbf{x})$ and integrating over $\mathbf{x}$ gives:

$$E_p[\mathbf{x}] = \boldsymbol{\mu} + Z^{-1}\boldsymbol{\Sigma}\left[\nabla_{\boldsymbol{\mu}}\int t(\mathbf{x})q(\mathbf{x})d\mathbf{x}\right] \tag{A.12}$$

$$= \boldsymbol{\mu} + Z^{-1}\boldsymbol{\Sigma}\nabla_{\boldsymbol{\mu}}Z \tag{A.13}$$

$$= \boldsymbol{\mu} + \boldsymbol{\Sigma}\nabla_{\boldsymbol{\mu}}\log(Z) \tag{A.14}$$

$$E_p[\mathbf{x}\mathbf{x}^T] = \boldsymbol{\Sigma} + 2\boldsymbol{\Sigma}\left(Z^{-1}\nabla_{\boldsymbol{\Sigma}}Z\right)\boldsymbol{\Sigma} + E_p[\mathbf{x}]\boldsymbol{\mu}^T + \boldsymbol{\mu}E_p[\mathbf{x}]^T - \boldsymbol{\mu}\boldsymbol{\mu}^T \tag{A.15}$$

$$= \boldsymbol{\Sigma} + 2\boldsymbol{\Sigma}\left(\nabla_{\boldsymbol{\Sigma}}\log(Z)\right)\boldsymbol{\Sigma} + E_p[\mathbf{x}]\boldsymbol{\mu}^T + \boldsymbol{\mu}E_p[\mathbf{x}]^T - \boldsymbol{\mu}\boldsymbol{\mu}^T \tag{A.16}$$

$$= \boldsymbol{\Sigma} + 2\boldsymbol{\Sigma}\left(\nabla_{\boldsymbol{\Sigma}}\log(Z)\right)\boldsymbol{\Sigma} + E_p[\mathbf{x}]E_p[\mathbf{x}]^T - \boldsymbol{\Sigma}\left(\nabla_{\boldsymbol{\mu}}\log(Z)\nabla_{\boldsymbol{\mu}}\log(Z)^T\right)\boldsymbol{\Sigma} \tag{A.17}$$

Rearranging (A.17) and using the moment matching update for the covariance $\boldsymbol{\Sigma}^* = E_p[\mathbf{x}\mathbf{x}^T] - E_p[\mathbf{x}]E_p[\mathbf{x}]^T$ gives the required result [42].

BIBLIOGRAPHY

[1] *The 1000 genomes project consortium. an integrated map of genetic variation from 1,092 human genomes.*, Nature, 491 (2012), pp. 56–65.

[2] H. AKAIKE, *Statistical predictor identification*, Ann. Inst. Statist. Math., 22 (1970), pp. 203–217.

[3] M. BINET, O. GASCUEL, C. SCORNAVACCA, E. J. P. DOUZERY, AND F. PARDI, *Fast and accurate branch lengths estimation for phylogenomic trees*, BMC Bioinformatics, 17 (2016).

[4] C. BISHOP, *Neural networks for pattern recognition*, Clarendon Press, 1995.

[5] ——, *Pattern Recognition and Machine Learning*, Springer, 2006.

[6] D. M. BLEI, A. KUCUKELBIR, AND J. D. MCAULIFFE, *Variational inference: A review for statisticians*, (2017).

[7] B. E. BOSER, I. M. GUYON, AND V. N. VAPNIK, *A training algorithm for optimal margin classifiers*, Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, (1992), pp. 144–152.

[8] X. BOYEN AND D. KOLLER, *Tractable inference for complex stochastic processes*, Uncertainty in Artificial Intelligence, (1998).

[9] R. P. BRENT, *Algorithms for minimization without derivatives.*, Prentice Hall, Englewood Cliffs, N.J., (1973).

[10] R. BRIBIESCA, L. HERRERA-ALSINA, E. RUIZ-SANCHEZ, A. SANCHEZ-GONZALEZ, AND J. E. SCHONDUBE, *Body mass as a supertrait linked to abundance and behavioural dominance in hummingbirds: a phylogenetic approach*, Ecology And Evolution, 9 (2019), pp. 1623–1637.

[11] A. L. BROWN, M. LI, A. GONCEARENCO, AND A. R. PANCHENKO, *Finding driver mutations in cancer: Elucidating the role of background mutational processes*, Plos Computational Biology, 15 (2019).

[12] W. BRUINSMA, *Spike and slab priors*. `https://wesselb.github.io/assets/write-ups/Bruinsma,%20Spike%20and%20Slab%20Priors.pdf`.

[13] T. D. BUI, J. YAN, AND R. E. TURNER, *A unifying framework for gaussian process pseudo-point approximations using power expectation propagation*, Journal of Machine Learning Research, 18 (2017), pp. 1–72.

[14] M. A. BUTLER AND A. A. KING, *Phylogenetic comparative analysis: a modeling approach for adaptive evolution.*, Am. Nat, 164 (2004), pp. 683–695.

[15] R. H. BYRD, P. LU, J. NOCEDAL, AND C. ZHU, *A limited memory algorithm for bound constrained optimization*, SIAM Journal on Scientific Computing, 16 (1995), pp. 1190–1208.

[16] C. CAMPBELL AND Y. YING, *Learning with Support Vector Machines*, Morgan & Claypool Publishers, 2011.

[17] C. CARVALHO, N. POLSON, AND J. G. SCOTT, *Handling sparsity via the horseshoe*, 12th International Conference on Artificial Intelligence and Statistics, 5 (2008), pp. 73–80.

[18] C. CORTES AND V. VAPNIK, *Support vector networks*, Machine Learning, 20 (1995), pp. 273–297.

[19] N. CRISTIANINI, *Introduction to Computational Genomics: A Case Studies Approach*, Oxford Press, 2007.

[20] N. CRISTIANINI, J. SHAWE-TAYLOR, A. ELISSEEF, AND J. KANDOLA, *On kernel-target alignment*, Advances in Neural Information Processing Systems 14, (2001).

[21] L. CSATÓ AND M. OPPER, *Sparse representation for gaussian process models*, Advances in Neural Information Processing Systems 13, (2000).

[22] ——, *Sparse on-line gaussian processes*, Neural Computation, 14 (2002), pp. 641–668.

[23] D. DUA AND C. GRAFF, *Uci machine learning repository*. http://archive.ics.uci.edu/ml, 2017.

[24] A. C. FAUL AND M. E. TIPPING, *Analysis of sparse bayesian learning*, Advances in Neural Information Processing Systems 14, (2001).

[25] T. FLETCHER, *Relevance vector machines explained*, UCL Technical Report, (2010).

[26] P. G. FOSTER, *Modeling compositional heterogeneity*, Syst. Biol., 53 (2004), pp. 485–495.

[27] R. P. FRECKLETON, P. H. HARVEY, AND M. PAGEL, *Phylogenetic analysis and comparative data: A test and review of evidence*, The American Naturalist, 160 (2002).

[28] J. H. FRIEDMAN, *Greedy function approximation: A gradient boosting machine*, IMS 1999 Reitz Lecture, (1999).

[29] L. GANG, K. S. RABE, J. NIELSEN, AND M. K. M. ENGQVIST, *Machine learning applied to predicting microorganism growth temperatures and enzyme catalytic optima*, PLOS Computational Biology, 8 (2019).

[30] A. GHALANOS AND S. THEUSSL, *Rsolnp: General Non-linear Optimization Using Augmented Lagrange Multiplier Method*, 2015.
R package version 1.16.

[31] A. GONCEARENCO, B.-G. MA, AND I. N. BEREZOVSKY, *Molecular mechanisms of adaptation emerging from the physics and evolution of nucleic acids and proteins*, Nucleic Acids Research, 42 (2013), pp. 2879–2892.

[32] A. GRAFEN, *The phylogenetic regression*, Philos.Trans. R. Soc. Lond. B, 16 (1989), pp. 119–157.

[33] M. GROUSSIN AND M. GOUY, *Adaptation to environmental temperature is a major determinant of molecular evolutionary rates in archaea*, Molecular Biology and Evolution, 28 (2011), pp. 2661–2674.

[34] S. F. GULL, *Bayesian inductive inference and maximum entropy*, Maximum Entropy and Bayesian Methods in Science and Engineering, Vol. 1: Foundations, (1988), pp. 53–74.

[35] M. HANSEN AND B. YU, *Minimum description length and model selection criteria for generalized linear models*, Institute of Mathematical Statistics Lecture Notes - Monograph Series, 40 (2003), pp. 145–163.

[36] L. J. HARMON, *Phylogenetic comparative methods: Learning from trees*. `https://lukejharmon.github.io/pcm/`, 2019.

[37] P. H. HARVEY AND A. J. L. BROWN, *New Uses for New Phylogenies*, Oxford, 1996.

[38] P. H. HARVEY AND M. D. PAGEL, *The Comparative Method in Evolutionary Biology*, Oxford Press, 1991.

[39] R. HENAO, X. YUAN, AND L. CARIN, *Bayesian nonlinear support vector machines and descriminative factor modelling*, Advances in Neural Information Processing Systems 27, (2014).

[40] R. HERBRICH, *Learning Kernel Classifiers: Theory and Algorithms*, The MIT Press, 2002.

[41] ——, *On gaussian expectation propagation*, Microsoft Research Technical Report, (2005).

[42] ——, *On minimising the kullback-leibler divergence*, Microsoft Research Technical Report, (2005).

[43] R. HERBRICH AND T. GRAEPEL, *A pac-bayesian margin bound for linear classifiers: Why svms work*, Advances in Neural Information Processing Systems 13, (2000), pp. 224–230.

[44] R. HERBRICH, T. GRAEPEL, AND C. CAMPBELL, *Bayesian learning in reproducing kernel hilbert spaces - the usefulness of the bayes point*, T U Berlin, (1999).

[45] D. HERNÁNDEZ-LOBATO AND J. M. HERNÁNDEZ-LOBATO, *Bayes machines for binary classification*, Pattern Recognition Letters, 29 (2008), pp. 1466–1473.

[46] L. S. T. HO AND C. ANÉ, *A linear-time algorithm for gaussian and non-gaussian trait evolution models*, Syst. Biol., 63 (2014), pp. 379–408.

[47] HÖHNA AND LANDIS, *Revbayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language*, Syst. Biol., 65 (2016), pp. 726–736.

[48] P. KAPLI, Z. YANG, AND M. J. TELFORD, *Phylogenetic tree building in the genomic age*, Nature Reviews Genetics, 21 (2020), pp. 428–444.

[49] W. KARUSH, *Minima of functions of several variables with inequalities as side constraints*, Master's thesis, Department of Mathematics, University of Chicago, 1939.

[50] H.-C. KIM AND Z. GHAHRAMANI, *Bayesian gaussian classification with the em-ep algorithm*, IEEE Trans. Pattern Anal. Machine Intell., 28 (2006), pp. 1948––1959.

[51] J. L. KING AND T. H. JUKES, *Non-darwinian evolution*, Science, 164 (1969), pp. 788–798.

[52] H. W. KUHN AND A. W. TUCKER, *Nonlinear programming*, Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics, (1951), pp. 481–492.

[53] S. KULLBACK AND R. A. LEIBLER, *On information and sufficiency*, Annals of Mathematical Statistics, 22 (1951), pp. 79–86.

[54] G. R. G. LANCKRIET, N. CRISTIANNINI, L. E. GHAOUI, AND M. I. JORDAN, *Learning the kernel matrix with semidefinite programming*, Journal of Machine Learning Research, 5 (2004), pp. 27–72.

[55] S. L. LAURITZEN, *Propagation of probabilities, means and variances in mixed graphical association models*, Journal of the American Statistical Association, 87 (1992), pp. 1098–1108.

[56] N. LAWRENCE, M. SEEGER, AND R. HERBRICH, *Fast sparse gaussian process methods: The informative vector machine*, Advances in Neural Information Processing Systems 15, (2002).

[57] A. LUNTZ AND V. BRAILOVSKY, *On estimation of characters obtained in statistical procedure of recognition*, Technicheskaya Kibernetica, (1969).

[58] D. J. C. MACKAY, *Bayesian interpolation*, Neural Computation, 4 (1992), pp. 415–447.

[59] E. P. MARTINS AND T. F. HANSEN, *New Uses for New Phylogenies. A microevolutionary link between phylogenies and comparative data*, Oxford, 1996.

[60] P. S. MAYBECK, *Stochastic models, estimation and control, chapter 12.7*, Academic Press, 1982.

[61] D. A. MCALLESTER, *Some pac-bayesian theorems*, Machine Learning, 37 (1999), pp. 355–363.

[62] T. P. MINKA, *Bayesian linear regression*, Technical Report, (1998).

[63] ——, *Expectation propagation for approximate bayesian inference*, Uncertainty in Artificial Intelligence, 15 (2001), pp. 362–369.

[64] ——, *A Family of Algorithms for Approximate Bayesian Inference*, PhD thesis, MIT Media Lab, Massachusetts Institute of Technology, 2001.

[65] ——, *Power ep*, Microsoft Research Technical Report MSR-TR-2004-149, (2004).

[66] ——, *Divergence measures and message passing*, Microsoft Research Technical Report MSR-TR-2005-173, (2005).

[67] T. J. MITCHELL AND J. BEAUCHAMP, *Bayesian variable selection in linear-regression.*, Journal of the American Statistical Association, 83 (1988), pp. 1023–1032.

[68] J. NELDER AND R. MEAD, *A simplex method for function minimization*, Computer Journal., 7 (1965), pp. 308–313.

[69] A. Y. NG, *Preventing 'overfitting' of cross-validation data*, Proceedings of the Fourteenth International Conference on Machine Learning, (1997).

[70] L. NGUYEN, H. A. SCHMIDT, A. VON HAESELER, AND B. Q. MINH, *Iq-tree: A fast and effective stochastic algorithm for estimating maximum likelihood phylogenies*, Molecular Biology and Evolution, 32 (2015), pp. 268–274.

[71] C. G. NORTHCUTT, T. WU, AND I. L. CHUANG, *Learning from confident examples: Rank pruning for robust classification with noisy labels*, Uncertainty in Artificial Intelligence, (2017).

[72] P. C. NOWELL, *The clonal evolution of tumor cell populations*, Science, 194 (1976).

[73] M. OPPER, *Model Neural Networks For Computation And Learning*, 2001.

[74] M. OPPER, U. PAQUET, AND O. WINTHER, *Peturbative corrections for approximate inference in gaussian latent variable models*, The Journal of Machine Learning Research, 14 (2013), pp. 2857–2898.

[75] M. OPPER AND O. WINTHER, *Mean field methods for classification with gaussian processes*, Advances in Neural Information Processing Systems 11, (1998).

[76] ——, *A bayesian approach to on-line learning*, Cambridge University Press, (1999).

[77] ——, *Advances in Large Margin Classifiers. Gaussian Processes and SVM: Mean Field and Leave-One-Out*, The MIT Press, 2000.

[78] M. PAGEL, *Inferring the historical patterns of biological evolution*, Nature, 401 (1999), pp. 877–884.

[79] G. PARISI, *Statistical Field Theory*, Addison-Wesley, 1988.

[80] M. W. PENNELL, J. M. EASTMAN, G. J. SLATER, J. W. BROWN, J. C. UYEDA, R. G. FITZJOHN, M. E. ALFARO, AND L. J. HARMON, *geiger v2.0: an expanded suite of methods for fitting macroevolutionary models to phylogenetic trees.*, 2014.

[81] K. B. PETERSEN AND M. S. PEDERSEN, *THE MATRIX COOKBOOK*, 2012.

[82] C. PETERSON AND J. R. ANDERSON, *A mean field theory learning algorithm for neural networks*, Complex Systems, 1 (1987), pp. 995–1019.

[83] J. C. PLATT AND A. H. BARR, *Constrained differential optimization*, Neural Information Processing Systems, (1987).

[84] K. S. POLLARD, M. J. HUBISZ, K. R. ROSENBLOOM, AND A. SIEPEL, *Detection of non-neutral substitution rates on mammalian phylogenies.*, Genome Research, 20 (2010), pp. 110–121.

[85] N. G. POLSON AND S. L. SCOTT, *Data augmentation support vector machines*, Bayesian Analysis, 6 (2011), pp. 1—23.

[86] Y. QI, A. ABDEL-GAWAD, AND T. P. MINKA, *Sparse-posterior gaussian processes for general likelihoods.*, Uncertainty in Artificial Intelligence, (2010).

[87] Y. QI, T. P. MINKA, AND R. W. PICARD, *Automatic determination of relevant features for bayes point machine*, Technical Report, (2001).

[88] Y. QI, T. P. MINKA, R. W. PICARD, AND Z. GHAHRAMANI, *Predictive automatic relevance determination by expectation propagation*, Proceedings of the 21st International Conference on Machine Learning, (2004).

[89] A. RAKOTOMAMONJY, F. BACH, S. CANU, AND Y. GRANDVALET, *Simplemkl*, Journal of Machine Learning Research, 9 (2008), pp. 2491–2521.

[90] C. E. RASMUSSEN AND C. K. I. WILLIAMS, *Gaussian Processes for Machine Learning*, The MIT Press, 2006.

[91] L. J. REVELL, *Size-correction and principal components for interspecific comparative studies*, Evolution, 63 (2009), pp. 3258–3268.

[92] ——, *Phylogenetic signal and linear regression on species data*, Methods in Ecology and Evolution, 1 (2010), pp. 319–329.

[93] ——, *phytools: An R package for phylogenetic comparative biology (and other things).*, 2014.

[94] J. RISSANEN, *Modelling by shortest data description*, Automatica, 14 (1978), pp. 465–471.

[95] M. F. ROGERS, T. R. GAUNT, AND C. CAMPBELL, *Prediction of driver variants in the cancer genome via machine learning methodologies*, Briefings in Bioinformatics, 22 (2021).

[96] M. F. ROGERS, H. A. SHIHAB, T. R. GAUNT, AND C. CAMPBELL, *Cscape: a tool for predicting oncogenic single-point mutations in the cancer genome*, Scientific Reports, (2017).

[97] D. ROJAS, C. A. MANCINA, J. J. FLORES-MARTINEZ, AND L. NAVARRO, *Phylogenetic signal, feeding behaviour, and brain volume in neotropical bats*, Journal of Evolutionary Biology, 26 (2013), pp. 1925–1933.

[98] D. B. SAUER AND D.-N. WANG, *Prediction of optimal growth temperature using only genome derived features*, Bioinformatics, 35 (2019), pp. 3224–3231.

[99] B. SCHOLKÖPF, A. J. SMOLA, AND R. C. WILLIAMSON, *New support vector algorithms*, Neural Computation, 12 (2000), pp. 1207—1245.

[100] G. E. SCHWARZ, *Estimating the dimension of a model*, Annals of Statistics, 6 (1978), pp. 461–464.

[101] G. A. F. SEBER AND A. J. LEE, *Linear Regression Analysis, 2nd edition*, Wiley, 2003.

[102] H. A. SHIHAB, J. GOUGH, D. N. COOPER, P. D. STENSON, G. L. A. BARKER, K. J. EDWARDS, I. N. M. DAY, AND T. R. GAUNT, *Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden markov models*, Human Mutation, 34 (2013), pp. 57–65.

[103] H. A. SHIHAB, M. F. ROGERS, J. GOUGH, M. MORT, D. N. COOPER, I. N. M. DAY, T. R. GAUNT, AND C. CAMPBELL, *An integrative approach to predicting the functional effects of non-coding and coding sequence variation*, Bioinformatics, 31 (2015), pp. 1536–1543.

[104] A. SIEPEL AND D. HAUSSLER, *Phylogenetic hidden markov models*, Statistical Methods in Molecular Evolution, Statistics for Biology and Health. Springer, New York, NY., (2005), pp. 325–351.

[105] J. SKILLING, *On parameter estimation and quantified maxent*, Maximum Entropy and Bayesian Methods, (1991), pp. 267–273.

[106] E. SNELSON AND Z. GHAHRAMANI, *Sparse gaussian processes using pseudo-inputs*, Advances in Neural Information Processing Systems 18, (2005).

[107] S. SONNENBURG, G. RÄTSCH, AND K. RIECK, *Large scale multiple kernel learning*, Journal of Machine Learning Research, 7 (2006), pp. 1531–1565.

[108] J. TATE, S. BAMFORD, AND H. JUBB, *Cosmic: the catalogue of somatic mutations in cancer*.
`http://cancer.sanger.ac.uk/cosmic/help/gene/analysis`, 2019.

[109] T. TATUSOVA, M. DICUCCIO, A. BADRETDIN, V. CHETVERNIN, E. P. NAWROCKI, L. ZASLAVSKY, A. LOMSADZE, K. D. PRUITT, M. BORODOVSKY, AND J. OSTELL, *Ncbi prokaryotic genome annotation pipeline*, Nucleic Acids Res., 44 (2016), pp. 6614–6624.

[110] S. TAVARÉ, *Some probabilistic and statistical problems in the analysis of dna sequences*, American Mathematical Society, (1986).

[111] R. TIBSHIRANI, *Regression shrinkage and selection via the lasso*, J. R. Stat. Soc. Ser. B (Stat. Method.), 58 (1996), pp. 267–288.

[112] M. E. TIPPING, *Sparse bayesian learning and the relevance vector machine*, Journal of Machine Learning Research, (2001), pp. 211–244.

[113] M. E. TIPPING AND C. M. BISHOP, *Probabilistic principal components analysis*, Journal of the Royal Statistical Society, Series B, 21 (1999), pp. 611–622.

[114] ——, *Variational relevance vector machines*, Uncertainty in Artificial Intelligence, (2000).

[115] A. K. UHRENHOLT, V. CHARVET, AND B. S. JENSEN, *Probabilistic selection of inducing points in sparse gaussian processes*, Uncertainty in Artificial Intelligence, (2021).

[116] J. C. UYEDA, D. S. CAETANO, AND M. W. PENNELL, *Comparative analyses of principal components can be misleading*, Systematic Biology, 64 (2015), pp. 677–689.

[117] V. VAPNIK, *The Nature Of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[118] V. VAPNIK AND O. CHAPELLE, *Advances in Large Margin Classifiers. Bounds on Error Expectation for SVM*, The MIT Press, 2000.

[119] C. WALDER, K. I. KIM, AND B. SCHÖLKOPF, *Sparse multiscale gaussian processes regression*, Proceedings of the 25th International Conference on Machine Learning, (2008).

[120] T. WATKIN, *Optimal learning with a neural network*, Europhysics Letters, 22 (1993), pp. 871–876.

[121] M. C. WEISS, F. L. SOUSA, N. MRNJAVAC, S. NEUKIRCHEN, M. ROETTGER, S. NELSON-SATHI, AND W. F. MARTIN, *The physiology and habitat of the last universal common ancestor*, Nature Microbiology, 1 (2016).

[122] F. WILCOXON, *Individual comparisons by ranking methods*, Biometrics Bull., 1 (1968), pp. 80–83.

[123] Y. YE, *Interior Algorithms for Linear, Quadratic, and Linearly Constrained Non-Linear Programming*, PhD thesis, Department of ESS, Stanford University, 1987.

[124] Y. YING, K. HUANG, AND C. CAMPBELL, *Enhanced protein fold recognition through a novel data integration approach.*, BMC Bioinformatics, 10 (2009).

[125] K. B. ZELDOVICH, I. N. BEREZOVSKY, AND E. I. SHAKHNOVICH, *Protein and dna sequence determinants of thermophilic adaptation*, PLOS Computational Biology, (2007).