Regression modelling using priors with Fisher information covariance kernels (I-priors)

Haziq Jamil
Department of Statistics
London School of Economics & Political Science

January 16, 2018

Abstract

Keywords: some, keywords, go, here

Declaration

I certify that the thesis I have presented for examination for the MPhil/PhD degree of the London School of Economics and Political Science is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it).

The copyright of this thesis rests with the author. Quotation from it is permitted, provided that full acknowledgement is made. This thesis may not be reproduced without my prior written consent.

I warrant that this authorisation does not, to the best of my belief, infringe the rights of any third party.

I declare that my thesis consists of XX,XXX words.

I confirm that Chapter X is jointly co-authored with Wicher Bergsma.

To-do list

Contents

List of Figures

List of Tables

List of Theorems

List of Definitions

Chapter 1

Introduction

This thesis has three main contributions. Firstly, computational methods relating to the estimation of I-prior models are studied. This includes, among others, outlining an efficient expectation-maximisation (EM) algorithm and employing a low-rank matrix approximation method to speed up estimation. Secondly, I-prior models are extended to be able to fit categorical-type responses for classification. Unlike in the continuous-response case, estimation involves an intractable integral, but this is overcome by way of a variational inference. Thirdly and finally, the use of I-priors for model selection is explored, whereby promising results are obtained especially for variable selection of linear models.

We begin by briefly introducing regularised least squares and how this coincides with the concept of placing priors on functions. We also introduce the I-prior and highlight the advantages I-prior modelling has over regularisation.

1.1 Regression and regularised least squares

Let $\mathcal{S} = \{(y_1, x_1), \dots, (y_n, x_n)\}$ denote a sample in a regression setting, where the responses y_i are real-valued, and the explanatory variables x_i belongs to some set $\mathcal{X}$. Each of the x_i represents characteristics of the ith observation, which may be real, categorical, multidimensional or even functional. Consider then the following regression model, which stipulates the dependence of y_i on x_i:

$$y_i = f(x_i) + \epsilon_i, \tag{1.1}$$

where the regression function f lies in some vector space of functions $\mathcal{F}$. Additionally, it is assumed that the errors ϵ_i are normally distributed according to

$$(\epsilon_1, \dots, \epsilon_n)^\top \sim \mathrm{N}_n(0, \Psi^{-1}). \tag{1.2}$$

where $\Psi = (\psi_{ij})_{i,j=1}^n$ is the precision matrix.

The task of regression modelling is to choose the most appropriate regression function $f \in \mathcal{F}$. It would be helpful if we had a measure of the quality of our choice of f. Define the risk functional $R : \mathcal{F} \to \mathbb{R}$ as

$$R[f] = \mathrm{E}[L\big(y, f(x)\big)] = \int L\big(y, f(x)\big) \,\mathrm{dP}(y, x), \tag{1.3}$$

where $L : \mathbb{R} \times \mathbb{R} \to [0, \infty)$ is some loss function, and $\mathrm{P}(y, x)$ is the probability measure of the observed sample. In most cases, this probability measure is unknown, and an empirical risk measure is used instead:

$$R[f] = \frac{1}{n} \sum_{i=1}^n L\big(y_i, f(x_i)\big). \tag{1.4}$$

The squared loss function given by

$$L\big(y_i, f(x_i)\big) = \sum_{j=1}^n \psi_{ij} \big(y_i - f(x_i)\big)\big(y_j - f(x_j)\big), \tag{1.5}$$

when used, defines the least squares regression. It is worthwhile noting that for the normal regression model, a solution obtained by minimising the squared loss function is equivalent to the maximum likelihood estimator.

This problem may be ill-posed, in the sense that if the space of functions is relatively unconstrained, then there is likely to be more than one solution to the regression problem. In fact, any function which passes through all the data points is an acceptable solution. This clearly leads to overfitting and poor generalisations.

The most common method to overcome this issue is Tikhonov regularisation. A regularisation term is added to the risk function, with the aim of imposing a penalty on the complexity of f. Concrete notions of complexity penalties can be introduced if $\mathcal{F}$ is a normed space, though reproducing kernel Hilbert space (RKHS) are typically used as it gives great conveniences (see Section 2). In particular, smoothness assumptions on f encoded by a suitable RKHS can be represented by using the RKHS norm $\|\cdot\|_\mathcal{F} : \mathcal{F} \to \mathbb{R}$ as the regularisation term. Therefore, the solution to the regularised least squares problem f_{reg} is the minimiser of the function from $\mathcal{F}$ to $\mathbb{R}$ defined by the mapping

$$f \mapsto \sum_{i=1}^n \sum_{j=1}^n \psi_{ij} \big(y_i - f(x_i)\big)\big(y_j - f(x_j)\big) + n\lambda \|f\|_\mathcal{F}^2, \tag{1.6}$$

which also happens to be the penalised maximum likelihood solution. The $\lambda > 0$ parameter - known as the regularisation parameter - controls the trade-off between the data-fit term and the penalty term, and is not usually known a priori and must be estimated from the data.

Tikhonov regularisation also has a well-known Bayesian interpretation, whereby the regularisation term encodes prior information about the function f. For the regression model stated earlier in (1.1) subject to the assumption in (1.2), let $\mathcal{F}$ be an RKHS equipped with the positive definite kernel $h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Then, it can be shown that f_{reg} is the posterior mean of f given a Gaussian process prior with zero mean and covariance kernel $H_\lambda = \big(\frac{1}{n\lambda}h(x_i, x_j)\big)_{i,j=1}^n$. This

There are two drawbakcs to Tikhonov regularisation, and the first is that it can systematically undersmooth. From a Bayesian viewpoint, undersmoothing can be said to occur if the support of the prior consists of functions that are rougher than those in $\mathcal{F}$. In particular, at least for certain RKHSs, the sample paths of the Gaussian process with the reproducing kernel of the RKHS as the covariance kernel are rougher (by some margin) than the roughest functions in the RKHS. Undersmoothing can then adversely impact the estimation of f, and in real terms might even show features and artefacts that are not really there.

3. Is it fair to say that most processes that we want to estimate hardly come about from realisations of smooth functions? I.e. Brownian motion paths (rough) are more likely to occur in "nature" than very, very smooth paths (say squared exponential paths).

The second drawback is in regards to estimation of the regularisation parameters. Estimation of these parameters requires either minimisation of some cross-validation error criterion, or a direct minimisation of the penalised functional (1.6). The latter of these two methods can be seen as obtaining an empirical Bayes estimate by maximising the marginal likelihood in the Bayesian interpretation of regularisation. Either way, estimation can prove difficult when there are a lot of regularisation parameters to estimate.

The I-prior methodology does not suffer from these two drawbacks. According to Bergsma (2017), ...

4. Complete this

1.2 The I-prior and its advantages

As alluded to earlier, RKHSs has many desirable properties. For the regression model (1.1) subject to (1.2), let $\mathcal{F}$ be an RKHS. Every RKHS defines a reproducing kernel function that is both symmetric and positive definite, and the converse is also true.

There are three main types of RKHS studied in this thesis, allowing linear and smooth effects of Euclidean covariates as well as the incorporation of categorical covariates: the *canonical* RKHS, consisting of linear functions of the covariates; the fractional Brownian motion (fBm) RKHS, consisting of smooth functions of the covariates; and the *Pearson* RKHS for nominal or categorical covariates. The fBm RKHS has smoothness parameter $\gamma \in (0, 1)$, called the Hurst coefficient. The most common value for this parameter is $1/2$, which for a real covariate gives a fitted function close to the familiar cubic spline smoother, although this could be treated as an unknown parameter to be estimated. More on these kernels later.

We can build upon these kernels by adding or multiplying them, and the result is still a positive definite kernel which induces a new RKHS. This is particularly useful because we can think of our regression function as being decomposed of functions belonging to different RKHS, depending on the effect of the covariate desired. As an example, suppose that each $x \in \mathcal{X}$ is 2-dimensional, so that $x = (x_1, x_2)$. We can assume that the regression function decomposes as follows:

$$f(x) = f(x_1, x_2) = f_1(x_1) + f_2(x_2) + f_{12}(x_1, x_2)$$

This is possible because $\mathcal{F}$ is a vector space over $\mathbb{R}$. Here, we have assumed that the function f partitions into two main effects f_1 and f_2 and an *interaction effect* f_{12}. Each of the main effects are in some RKHS, depending on the effect of the corresponding covariate (linear, smooth, or nominal), and thus would have a kernel $h_j : \mathcal{X}_j \times \mathcal{X}_j \to \mathbb{R}$. As the scale of an RKHS over a set $\mathcal{X}$ may be arbitrary, each of the kernels are multiplied by a scale parameter λ_j. The space of functions for the interaction effects are then assumed to be in the so-called tensor product space of the corresponding main effect functions. In our case, $f_{12} \in \mathcal{F}_{12}$, where $\mathcal{F}_{12}$ is an RKHS with kernel equal to the product of kernels $h_{12}\big((x_1, x_2), (x'_1, x'_2)\big) = \lambda_1\lambda_2 \cdot h_1(x_1, x'_1)h_2(x_2, x'_2)$.

Suppose that we have a multilevel data set, where x_1 is real-valued, x_2 is nominal-valued indicating the level to which the observation belongs to. We can model these data by choosing the canonical kernel on x_1 and the Pearson kernel for the x_2, and the interaction effect represents the varying effect of x_1 in each level x_2. If instead we had a time covariate x_1 and a categorical covariate x_2 representing treatment effect say, then we can build a longitudinal model with either a linear or smooth effect of time. Again, the interaction effect will convey x_2 as time-varying. Of course, we can partition the function as is necessary, such as excluding the interaction effect or including additional terms such as three-way interactions. Now suppose that we have functional data, i.e. the set $\mathcal{X}$ consists of functions. If we assume that the xs lie in some Hilbert space (not necessarily an RKHS) then we have an inner-product (and also a norm) defined on $\mathcal{X}$. As we will see later, the canonical and fBm kernels make use of the inner-product on $\mathcal{X}$ so we are able to proceed as we did before. We can see that this framework provides a unifying approach to various regression models.

We discussed earlier that regularisation has a Bayesian interpretation, whereby a prior distribution is assigned to the regression function. Specifically, it is a Gaussian process prior with mean zero and covariance kernel equal to the reproducing kernel of the RKHS that f belongs to. The I-prior for f is a also a zero mean Gaussian prior but has a different covariance kernel, namely the Fisher information for f. If h is the reproducing kernel for the RKHS, then the Fisher information between $f(x)$ and $f(x')$ is given as

$$\mathcal{I}[f(x), f(x')] = \sum_{i=1}^n \sum_{j=1}^n \psi_{ij} h(x, x_i) h(x', x_j). \tag{1.7}$$

Hence, f follows an I-prior distribution if it can be written in the form

$$f(x) = \sum_{i=1}^{n} h(x, x_j) w_j, \tag{1.8}$$

where

$$(w_1, \dots, w_n)^\top \sim \text{N}_n(0, \Psi).$$

Note that, of course, a non-zero value or function for the prior mean could be taken as well. The I-prior is a class of objective priors - it is the distribution for which entropy is maximised (subject to certain constraints). In this sense, it is considered the prior which gives the least amount of information a priori - then perhaps the term I-prior is somewhat of a misnomer, since the 'I' stands for (Fisher) information.

An intuitively attractive property of the I-prior is that if much information about a linear functional of f (e.g. a regression coefficient) is available, its prior variance is large, and the data have a relatively large influence on the posterior, while if little information about a linear functional is available, the posterior will be largely determined by the prior mean, which serves as a 'best guess' of f.

1.3 Estimation

The I-prior methodology consists of estimation of the regression function by its posterior distribution under the I-prior, where we take the posterior mean as the summary measure. Write $\mathbf{f} = \big(f(x_1), \dots, f(x_n)\big)^\top$. From (1.7), the Fisher covariance kernel for f is $H_\lambda \Psi H_\lambda$, where $H_\lambda = \big(h_\lambda(x_i, x_j)\big)_{i,j=1}^n$ and h_λ is the (scaled) reproducing kernel of $\mathcal{F}$. The I-prior on f for model (1.1) subject to (1.2) is

$$\mathbf{f} \sim \text{N}_n(\mathbf{f}_0, H_\lambda \Psi H_\lambda)$$

where $\mathbf{f}_0 = \big(f_0(x_1), \dots, f_0(x_n)\big)^\top$ is some prior mean typically set to zero. We are then interested in two main things:

1. The posterior distribution for the regression function

$$p(\mathbf{f}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{\int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}) \, \text{d}\mathbf{f}}$$

2. The posterior predictive distribution for new data x_{new}

$$p(y_{\text{new}}|\mathbf{y}) = \int p(y_{\text{new}}|f_{\text{new}}, \mathbf{y}) p(f_{\text{new}}|\mathbf{y}) \, \text{d}\mathbf{y},$$

where $f_{\text{new}} = f(x_{\text{new}})$.

It can be shown that for any $x \in \mathcal{X}$, the posterior distribution of f is normal with mean and variance given by

$$\text{E}[f(x)|\mathbf{y}] = f_0(x) + \mathbf{h}_\lambda(x)\Psi H_\lambda(H_\lambda \Psi H_\lambda + \Psi^{-1})^{-1}\big(y - f_0(x)\big)$$

and

$$\text{Var}[f(x)|\mathbf{y}] = \mathbf{h}_\lambda(x)^\top(H_\lambda \Psi H_\lambda + \Psi^{-1})^{-1}\mathbf{h}_\lambda(x),$$

where $\mathbf{h}_\lambda(x) = \big(h_\lambda(x, x_1), \ldots, h_\lambda(x, x_n)\big)^\top$.

There is the matter of estimating the model (hyper-)parameters - the error precision Ψ, the RKHS scale parameters λ, and any other parameters that might be associated with the kernel (e.g. the smoothing parameter in an fBm kernel). These may be estimated in a variety of ways. The first is by maximum marginal likelihood, which is also known as the empirical Bayes approach. The marginal distribution $p(y) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})\,\text{d}\mathbf{f}$ is easily obtained in the case of the normal model, and it is

$$\mathbf{y} = (y_1, \ldots, y_n)^\top \sim \text{N}_n\left(f_0(x), H_\lambda \Psi H_\lambda + \Psi^{-1}\right).$$

The marginal likelihood can be maximised in the usual way (e.g. Newton-type methods) with respect to the model parameters, but for complex models involving a lot of parameters, this may be challenging. Instead, a better approach is the expectation-maximisation (EM) algorithm. The I-prior model emits a simple E- and M-step which makes this method favourable. For most models, a closed-form solution to the M-step is available thereby reducing the EM algorithm to an iterative updating scheme of the parameters. Finally, a fully Bayesian approach may be taken as well, whereby prior distributions are assigned to the model parameters and posterior samples obtained via Markov chain Monte Carlo (MCMC) methods.

Regardless of the estimation procedure, computational complexity is dominated by the inversion of the $n \times n$ matrix $V_y = H_\lambda \Psi H_\lambda + \Psi^{-1}$ as a function of the model parameters. In the case of Newton-type approaches to likelihood maximisation, V_y^{-1} appears in the kernel of the marginal Gaussian density for $\mathbf{y}$. In the case of the EM algorithm, every update cycle also involves a similar calculation, and this is quite similar to the calculations required from a Gibbs-sampling approach for stochastic MCMC sampling.

I-priors, while being philosophically different from Gaussian process priors, do share the same computational hurdle. As such, several methods exist in the machine learning literature to overcome this issue. Amongst others, is a method to approximate the covariance kernel by a low-rank matrix, so that the most expensive operation of inverting a $n \times n$ matrix is greatly reduced. Our approach is to apply the Nyström method of low-rank matrix approximation, and we find that this works reasonably well for the fBm RKHS.

Another computational hurdle is to ensure numerical stability. We find that due to the structure of the marginal covariance V_y, numerical instabilities can and are likely to occur - which may give rise to embarrassments such as negative covariances. We employ a stable eigendecomposition regime which allows us to efficiently calculate matrix squares and inverses by making use of the spectral theorem.

1.4 I-priors for classification

Suppose now we are interested in a regression model where the responses are categorical. Assume a categorical distribution on the responses with certain probabilities for each class and for each observation. This is of course a generalisation of the Bernoulli distribution to more than two possible outcomes. The question is how can we relate the effect of the covariates through the function, which has unrestricted range, to the responses, which may only take one of m several outcomes? In the spirit of generalised linear models, we answer this by making use of an appropriate link function, and our case, the probit link function. In the binary case, this amounts to squashing our regression function through the (inverse) probit link function in order to model probabilities which are between zero and one. This idea is then extended to the multinoulli case, giving rise to a multinomial probit I-prior model, which we call I-probit.

The main issue with estimation now is that because our responses no longer follow a Gaussian distribution, the relevant marginal distribution, on which the posterior depends, can no longer be found in closed form. The integral required to perform the calculation is intractable, and the focus now is on methods to adequately approximate the integral.

In the Bayesian literature, the Laplace approximation amounts to approximating the posterior distribution with a normal distribution centred around the mode of the integrand. Additionally, the covariance matrix is equal to the inverse (negative) Hessian. Having approximated the posterior by a Gaussian distribution, one could then proceed to find the marginal easily, which is then maximised. Due to the Newton step in the Laplace step, the whole procedure scales cubicly with both the sample size and the number of outcomes, which makes it undesirable to implement.

MCMC methods such as Gibbs sampling or Hamiltonian Monte Carlo can also provide a stochastic approximation to the integral. Unfortunately, the difficulties faced in the continuous case for MCMC methods also present themselves in the categorical case.

Deterministic approaches such as quadrature methods prove unfeasible. Quadrature methods scale exponentially with the variables of integration, in our case, is the sample size.

We consider a type of approximation based on minimising the Kullback-Leibler (KL) divergence from the approximating density to the true posterior density. This is done without making any distributional assumptions, only that the our approximating density factorises over its components (ie an independence assumption) - this is known as the mean-field approximation, which has its roots in the physics literature. As an aside, the term ‘variational’ stems from the fact that a minimisation of a functional, rather than a function, is involved, and this requires the calculus of variations.

By working in a fully Bayesian setting, we append the model parameters to the list of unknowns in which to estimate, and employ the variational approximation to find a suitable approximation to the required posterior density. The result is an iterative algorithm, similar to the EM.

As this variational EM works harmoniously with exponential family distributions, the probit link is much preferred over the logit. Most of the EM updating cycles which could be found in closed form are also applicable in the variational EM. Unlike Gaussian process priors, the variational method does not typically result in closed-form updates for the RKHS scale parameters. In such cases, an additional step such as importance sampling is required, which arguably reduces efficiency of the whole variational scheme.

The variational EM is implemented in the R package iprobit. This has been shown to work well for several toy examples as well as real world applications. In the binary case, the I-prior outperforms other popular classication methods including k-nearest neighbours, support vector machines, and Gaussian process classification.

1.5 I-priors for Model selection

As mentioned earlier, model selection can easily be done by comparing likelihoods (empirical Bayes factors). However, with a large number of variables, these pairwise comparisons quickly become unfeasible to perform. We suggest a fully Bayesian approach to estimating posterior model probabilities, and selecting models based on these quantities. For example, one may choose the model which gives the largest posterior model probability (maximum probability model). These are done using MCMC methods (Gibbs sampling).

We restrict ourselves to linear models only. We can easily derive an equivalent I-prior representation by working in the feature space of the betas (linear effects). As a side note, if the dimensions of the linear effects is much, much less than the sample size, then it is worth working in this representation.

We believe the I-prior performs superiorly in cases where there is multicollinearity. This is evidenced by the simulation results that we conducted on a 100-variate experiment, and also in the real-data examples comparing our method with others such as greedy selection, g-priors, and regularisation (ridge and Lasso).

Chapter 2

Regression using I-priors

Regression using I-priors. Test equation reference (1.1).

2.1 Some functional analysis

For a cleaner read, we do not use boldface type to denote vectors and matrices in this subsection. We begin by recalling that a vector space is a set endowed with two special operations: addition and scalar multiplication. A normed vector space is a vector space whose vectors have lengths, as induced by its norm.

Definition 2.1 (Norms)**.** Let $\mathcal{F}$ be a vector space over $\mathbb{R}$. A non-negative function $||\cdot||_\mathcal{F} : \mathcal{F} \times \mathcal{F} \to [0,\infty)$ is said to be a norm on $\mathcal{F}$ if all of the following are satisfied:

- **Absolute homogeneity:** $||\lambda f||_\mathcal{F} = |\lambda| ||f||_\mathcal{F}$, $\forall \lambda \in \mathbb{R}$, $\forall f \in \mathcal{F}$
- **Triangle inequality:** $||f+g||_\mathcal{F} \leq ||f||_\mathcal{F} + ||g||_\mathcal{F}$, $\forall f, g \in \mathcal{F}$
- **Separates points:** $||f||_\mathcal{F} = 0 \Leftrightarrow f = 0$

The norm $||\cdot||_\mathcal{F}$ induces a metric (a notion of distance) on $\mathcal{F}$: $d(f,g) = ||f-g||_\mathcal{F}$.

We can then define a Cauchy sequence.

Definition 2.2 (Cauchy sequence)**.** A sequence $\{f_n\}_{n=1}^\infty$ of elements of a normed vector space $(\mathcal{F}, ||\cdot||_\mathcal{F})$ is said to be a Cauchy sequence if for every $\epsilon > 0$, $\exists N = N(\epsilon) \in \mathbb{N}$, such that $\forall n, m > N$, $||f_n - f_m||_\mathcal{F} < \epsilon$.

Another important structure of vector spaces is the inner product, which allows us to study various geometrical notions such as orthogonality, among other things.

Definition 2.3 (Inner products)**.** Let $\mathcal{F}$ be a vector space over $\mathbb{R}$. A function $\langle \cdot, \cdot \rangle_\mathcal{F} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ is said to be an inner product on $\mathcal{F}$ if all of the following are satisfied:

- **Symmetry:** $\langle f, g\rangle_{\mathcal{F}} = \langle g, f\rangle_{\mathcal{F}}$, $\forall f, g \in \mathcal{F}$
- **Linearity:** $\langle af_1 + bf_2, g\rangle_{\mathcal{F}} = a\langle f_1, g\rangle_{\mathcal{F}} + b\langle f_2, g\rangle_{\mathcal{F}}$, $\forall f_1, f_2, g \in \mathcal{F}$ and $\forall a, b \in \mathbb{R}$
- **Non-degeneracy:** $\langle f, f\rangle_{\mathcal{F}} = 0 \Leftrightarrow f = 0$
- **Positive-definiteness:** $\langle f, f\rangle_{\mathcal{F}} \geq 0$, $\forall f \in \mathcal{F}$

We can always define a norm on $\mathcal{F}$ using the inner product as $||f||_{\mathcal{F}} = \sqrt{\langle f, f\rangle_{\mathcal{F}}}$. Additionally, an inner product is said to be positive definite if $\langle f, f\rangle_{\mathcal{F}} \geq 0$, $\forall f \in \mathcal{F}$, and we can always define a norm on $\mathcal{F}$ using this inner product as $||f||_{\mathcal{F}} = \sqrt{\langle f, f\rangle_{\mathcal{F}}}$. Conversely, an inner product is said to be negative definite if $\langle f, f\rangle_{\mathcal{F}} \leq 0$, $\forall f \in \mathcal{F}$. An inner product is said to be indefinite it is neither positive nor negative definite.

A vector space equipped with a positive definite inner product that is also complete (contains the limits of all Cauchy sequences) is known as a Hilbert space. We now define a reproducing kernel Hilbert space.

A generalisation of a Hilbert space, one which is equipped with an indefinite inner product, is known as a Krein space.

Definition 2.4 (Krein space)**.** A vector space $\mathcal{F}$ for which an inner product $\langle \cdot, \cdot\rangle_{\mathcal{F}}$ is defined is called a Krein space if there are two Hilbert spaces $\mathcal{F}_+$ and $\mathcal{F}_-$ spanning $\mathcal{F}$ such that

- All $f \in \mathcal{F}$ can be decomposed as $f = f_+ + f_-$ where $f_+ \in \mathcal{F}_+$ and $f_- \in \mathcal{F}_-$; and
- $\forall f, f' \in \mathcal{F}$, $\langle f, f'\rangle_{\mathcal{F}} = \langle f_+, f'_+\rangle_{\mathcal{F}_+} - \langle f_-, f'_-\rangle_{\mathcal{F}_-}$.

Any Hilbert space can be seen as a Krein space by taking $\mathcal{F}_- = \{0\}$. As we are dealing with function spaces, it might seem unusual in defining functions from $\mathcal{F}$ to $\mathbb{R}$, as the elements of $\mathcal{F}$ are themselves functions. For a space of functions $\mathcal{F}$ on $\mathcal{X}$, we define the evaluation functional that assigns a value to $f \in \mathcal{F}$ for each $x \in \mathcal{X}$.

Definition 2.5 (Evaluation functional)**.** Let $\mathcal{F}$ be a vector space of functions $f : \mathcal{X} \to \mathbb{R}$, defined on a non-empty set $\mathcal{X}$. For a fixed $x \in \mathcal{X}$, the function $\delta_x : \mathcal{F} \to \mathbb{R}$ as defined by $\delta_x(f) = f(x)$ is called the (Dirac) evaluation functional at x. Evaluation functionals are always linear.

There are two more concepts that we need to cover before defining a reproducing kernel Hilbert/Krein space.

Definition 2.6 (Linear operator)**.** A function $A : \mathcal{F} \to \mathcal{G}$, where $\mathcal{F}$ and $\mathcal{G}$ are both normed vector spaces over $\mathbb{R}$, is called a linear operator if and only if it satisfies the following properties:

- **Homogeneity**: $A(af) = aA(f)$, $\forall a \in \mathbb{R}$, $\forall f \in \mathcal{F}$
- **Additivity**: $A(f + g) = A(f) + A(f')$, $\forall f, f' \in \mathcal{F}$.

Definition 2.7 (Bounder operator)**.** The linear operator $A : \mathcal{F} \to \mathcal{G}$ between two normed spaces $(\mathcal{F}, ||\cdot||_{\mathcal{F}})$ and $(\mathcal{G}, ||\cdot||_{\mathcal{G}})$ is said to be a bounded operator if $\exists \lambda \in [0, \infty)$ such that

$$||A(f)||_{\mathcal{G}} < \lambda ||f||_{\mathcal{F}}.$$

Now we define a reproducing kernel Hilbert space.

Definition 2.8 (Reproducing kernel Hilbert space)**.** A Hilbert space of real-valued functions $f : \mathcal{X} \to \mathbb{R}$ on a non-empty set $\mathcal{X}$ is called a reproducing kernel Hilbert space if the evaluation functional $\delta_x : f \mapsto f(x)$ is bounded (equivalently, continuous[1]), i.e. $\exists \lambda_x \geq 0$ such that $\forall f \in \mathcal{F}$,

$$|f(x)| = |\delta_x(f)| \leq \lambda_x ||f||_{\mathcal{F}}.$$

The definition is similar for Krein spaces, but there is a slight technical condition regarding strong topologies (see **Ong2004**). Interestingly, the definition above has no mention of what a reproducing kernel is. Let us define it below.

Definition 2.9 (Reproducing kernel Krein space)**.** A Krein space of real-valued functions $f : \mathcal{X} \to \mathbb{R}$ on a non-empty set $\mathcal{X}$ is called a reproducing kernel Krein space (RKKS) if the evaluation functional δ_x is a bounded linear operator $\forall x \in \mathcal{X}$, endowed with its strong topology.

Definition 2.10 (Kernels)**.** Let $\mathcal{X}$ be a non-empty set. A function $h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a kernel if there exists a real Hilbert space $\mathcal{F}$ and a map $\phi : \mathcal{X} \to \mathcal{F}$ such that $\forall x, x' \in \mathcal{X}$,

$$h(x, x') = \langle \phi(x), \phi(x') \rangle.$$

Such a map $\phi : \mathcal{X} \to \mathcal{F}$ is known as the *feature map*, and the space $\mathcal{F}$ as the *feature space*. Out of interest, a given kernel may correspond to more than one feature map.

Definition 2.11 (Reproducing kernels)**.** Let $\mathcal{F}$ be a Hilbert space of functions over a non-empty set $\mathcal{X}$. A function $h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a reproducing kernel of $\mathcal{F}$ if h satisfies

- $\forall x \in \mathcal{X}$, $h(\cdot, x) \in \mathcal{F}$; and
- $\forall x \in \mathcal{X}$, $f \in \mathcal{F}$, $\langle f, h(\cdot, x) \rangle_{\mathcal{F}} = f(x)$ (the reproducing property).

In particular, for any $x, x' \in \mathcal{X}$,

$$h(x, x') = \langle h(\cdot, x), h(\cdot, x') \rangle_{\mathcal{F}}.$$

The connection between the definition of a RKHS and reproducing kernels is this: $\mathcal{F}$ is a RKHS space if and only if $\mathcal{F}$ has a reproducing kernel. It can also be proven that if

[1] For any two function $f, g \in \mathcal{F}$, $|f(x) - g(x)| = |\delta_x(f) - \delta_x(g)| = |\delta_x(f - g)| \leq \lambda_x ||f - g||_{\mathcal{F}}$ for some $\lambda_x \geq 0$, thus is said to be Lipschitz continuous, which implies uniform continuity. This property implies pointwise convergence from norm convergence in $\mathcal{F}$.

this kernel exists, it is unique. We now turn to the one of the most important properties of the kernel function: positive-definiteness.

By the definition of symmetry and positive definiteness of inner products on Hilbert spaces, it follows that kernel functions are symmetric and positive definite, and the following lemma is easily proven.

Lemma 2.1 (Positive-definiteness). *Let $\mathcal{F}$ be a Hilbert space (not necessarily a RKHS), $\mathcal{X}$ a non-empty set and $\phi : \mathcal{X} \to \mathcal{F}$. Then $h(x, x') := \langle \phi(x), \phi(x') \rangle_{\mathcal{F}}$ is a symmetric and positive definite function, where a symmetric function $h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be positive definite if*

$$\sum_{i=1}^{n} \sum_{k=1}^{n} a_i a_j h(x_i, x_k) \geq 0.$$

$\forall n \geq 1$, $\forall a_1, \ldots, a_n \in \mathbb{R}$, and $\forall x_1, \ldots, x_n \in \mathcal{X}$.

Proof.

$$\begin{aligned}
\sum_{i=1}^{n} \sum_{k=1}^{n} a_i a_j h(x_i, x_k) &= \sum_{i=1}^{n} \sum_{k=1}^{n} \langle a_i \phi(x_i), a_k \phi(x_k) \rangle_{\mathcal{F}} \\
&= \left\langle \sum_{i=1}^{n} a_i \phi(x_i), \sum_{k=1}^{n} a_k \phi(x_k) \right\rangle_{\mathcal{F}} \\
&= \left\| \sum_{i=1}^{n} a_i \phi(x_i) \right\|_{\mathcal{F}}^2 \\
&\geq 0
\end{aligned}$$

□

Corollary 2.1.1. *Reproducing kernels of a RKHS are positive definite. For an RKKS, the reproducing kernel can be shown to be the difference between two positive definite kernels, but need not be itself positive definite.*

Proof. Take $\phi : x \mapsto h(\cdot, x)$. By Lemma 2.1, one has $h(x, x') = \langle h(\cdot, x), h(\cdot, x') \rangle_{\mathcal{F}}$, which is the reproducing property of the kernel in a RKHS. The second statement follows by a similar argument and by definition of a RKKS (see Definition 2.4). □

Remarkably, the reverse direction also holds: a positive definite function is guaranteed to be the inner product in a Hilbert space between features $\phi(x)$ (Theorem 4.16 pp.118, Steinward and Christman, 2008). This proof is a bit technical so will not be shown here. What's important though, is By Definition 2.11 and Lemma 2.1 above, we can see how a reproducing kernel Hilbert space defines a reproducing kernel function that is both symmetric and positive definite. The celebrated Moore-Aronszajn theorem goes the other direction by stating that every symmetric, positive-definite function is a reproducing kernel[2] and defines a unique RKHS, thus establishing a bijection between the set of all positive definite functions on $\mathcal{X} \times \mathcal{X}$ and the set of all reproducing kernel Hilbert spaces. For Krein spaces it is slightly different: 1) The reproducing kernel of a RKKS can be shown to be the difference between two positive definite kernels, so need not be positive definite itself; and 2) Every RKKS has a unique reproducing kernel, but a given reproducing kernel may have more than one RKKS associated with it.

Thus far, we have seen that given a RKHS $\mathcal{F}$, we may define a unique reproducing kernel associated with $\mathcal{F}$ which is symmetric and positive definite. The celebrated Moore-Aronszajn theorem goes the other direction by stating that every symmetric, positive-definite function is a reproducing kernel and defines a unique RKHS, thus establishing a bijection between the set of all positive definite functions on $\mathcal{X} \times \mathcal{X}$ and the set of all reproducing kernel Hilbert spaces. In other words, the kernel completely determines the function space. It is not quite the same with Krein spaces, however. Every RKKS has a unique reproducing kernel, but a given reproducing kernel may have more than one RKKS associated with it.

So why the fascination with reproducing kernel Hilbert/Krein spaces? In our case, it is the possibility of representing a regression analysis as functions in a RKKS. This greatly helps facilitate interpretation of models.

Lemma 2.2 (Regression functions in a RKKS)**.** *$\mathcal{F}$ is an RKKS if and only if there exists a feature space $\mathcal{B}$ for which a feature map of $\mathcal{F}$ maps onto.*

Proof. We first define a feature space and a feature map of $\mathcal{F}$.

Definition 2.12 (Features)**.** Consider a Krein space $\mathcal{F}$ of real functions over $\mathcal{X}$ with reproducing kernel h. Let $\mathcal{B}$ be a real Krein space over $\mathcal{X}$, and ϕ a map from $\mathcal{X}$ to $\mathcal{B}$, such that for every $f \in \mathcal{F}$, $\exists \beta \in \mathcal{B}$ such that

$$f(x) = \langle \phi(x), \beta \rangle_{\mathcal{B}}, \forall x \in \mathcal{X} \tag{2.1}$$

and

$$\langle f, f' \rangle_{\mathcal{F}} = \langle \beta, \beta' \rangle_{\mathcal{B}}. \tag{2.2}$$

Then $\mathcal{B}$ is called a *feature space* and ϕ a *feature map* of $\mathcal{F}$.

Now suppose $\mathcal{F}$ is a Krein space of real functions over $\mathcal{X}$ with a feature space $\mathcal{B}$ and a feature map ϕ. Then by defining the kernel function as $h(x, x') = \langle \phi(x), \beta \rangle_{\mathcal{B}}$, we show the reproducing property

$$\langle f, h(\cdot, x) \rangle_{\mathcal{F}} = \langle \phi(x), \beta \rangle_{\mathcal{B}} = f(x),$$

[2] Basically every positive definite function is a reproducing kernel, and every reproducing kernel is a kernel, and every kernel is positive definite, so all three notions are exactly the same.

where the first equality is by (2.2) and the second by (2.1). Hence h is a reproducing kernel of $\mathcal{F}$ and $\mathcal{F}$ is a RKKS. The other direction is proven by Definition 8. □

A consequence of the proof of the Moore-Aronszajn theorem (**Hein2004**) is that we can show that any function f in a RKHS $\mathcal{F}$ with kernel h can be written in the form $f(x) = \sum_{i=1}^n h(x, x_i)w_i$ for some $n \in \mathbb{N}$ (i.e. $\mathcal{F}$ is spanned by the functions $h(\cdot, x)$). More precisely, $\mathcal{F}$ is the completion of the space $\mathcal{G} = \operatorname{span}\{h(\cdot, x) \,|\, x \in \mathcal{X}\}$ endowed with the inner product

$$\left\langle \sum_{i=1}^n w_i h(\cdot, x_i), \sum_{j=1}^n w_j h(\cdot, x_j) \right\rangle_{\mathcal{G}} = \sum_{i=1}^n \sum_{j=1}^n w_i w_j h(x_i, x_j).$$

2.2 The Fisher information

Let Y be a random variable with density in the parameteric family $\{p(\cdot\ ; f)|f \in \mathcal{F}\}$ with f belonging to a Hilbert space $\mathcal{F}$. If $p(Y; f) > 0$, the log-likelihood function of f is denoted $l(f|Y) = \log p(Y; f)$. Assuming existence, the score is defined as the gradient[3] $\nabla l(f|Y)$. The Fisher information $I[f] \in \mathcal{F} \otimes \mathcal{F}$ for $f \in \mathcal{F}$ is

$$I[f] = -\operatorname{E}[\nabla^2 l(f|Y)|f].$$

Specifically for our regression function as defined in (**??**) subject to f belonging to a RKHS, we can derive the Fisher information for f to be

$$I[f] = \sum_{i=1}^n \sum_{j=1}^n \psi_{ij} h(\cdot, x_i) \otimes h(\cdot, x_j),$$

where ψ_{ij} are the (i, j)-th entries of the precision matrix Ψ.

Proof. For $x \in \mathcal{X}$, let $k_x : \mathcal{F} \to \mathbb{R}$ be defined by $k_x(f) = \langle h(\cdot, x), f\rangle_{\mathcal{F}}$. By the reproducing

[3] Let $k : \mathcal{F} \to \mathbb{R}$. Denote the directional derivate of k in the direction g by $\nabla_g k$, that is,

$$\nabla_g k(f) = \lim_{\delta \to 0} \frac{k(f + \delta g) - k(f)}{\delta}.$$

The gradient of k, denoted by ∇k, is the unique vector field satisfying

$$\langle \nabla k(f), g\rangle_{\mathcal{F}} = \nabla_g k(f), \quad \forall f, g \in \mathcal{F}.$$

property, $k_x(f) = f(x)$. The directional derivative of $k_x(f)$ in the direction g is

$$\begin{aligned}\nabla_g k_x(f) &= \lim_{\delta \to 0} \frac{k(f+\delta g) - k(f)}{\delta} \\ &= \lim_{\delta \to 0} \frac{\langle h(\cdot, x), f + \delta g\rangle_{\mathcal{F}} - \langle h(\cdot, x), f\rangle_{\mathcal{F}}}{\delta} \\ &= \lim_{\delta \to 0} \frac{\delta \langle h(\cdot, x), g\rangle_{\mathcal{F}}}{\delta} = \langle h(\cdot, x), g\rangle_{\mathcal{F}}.\end{aligned}$$

Thus, the gradient is $\nabla k_x(f) = h(\cdot, x)$ by definition. The log-likelihood of f is given by

$$l(f|y, \alpha, \Psi) = C - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \psi_{ij} \big(y_i - \alpha - k_{x_i}(f)\big)\big(y_j - \alpha - k_{x_j}(f)\big)$$

for some constant C, and the score by

$$\nabla l(f|y, \alpha, \Psi) = \sum_{i=1}^n \sum_{j=1}^n \psi_{ij} \big(y_i - \alpha - k_{x_i}(f)\big) \nabla k_{x_j}(f).$$

We can then calculate the Fisher information as

$$\begin{aligned}I[f] = -\operatorname{E}[\nabla^2 l(f|Y)|f] &= \operatorname{E}\left[\sum_{i=1}^n \sum_{j=1}^n \psi_{ij} \nabla k_{x_i}(f) \otimes \nabla k_{x_j}(f) \;\middle|\; f \right] \\ &= \sum_{i=1}^n \sum_{j=1}^n \psi_{ij} h(\cdot, x_i) \otimes h(\cdot, x_j).\end{aligned}$$

by substituting $\nabla k_x(f) = h(\cdot, x)$, the expectation is free of f

□

We can also compute the Fisher information for a linear functional of f, or between two linear functionals of f. We quote the following lemma (**Bergsma2014**):

Lemma 2.3 (Fisher information for linear functionals of elements in a Hilbert space)**.** *Let $\mathcal{F}$ be a Hilbert space. Denote the Fisher information for $f \in \mathcal{F}$ as $I[f]$. The Fisher information for $\langle f, g\rangle$ is given as*

$$I[\langle f, g\rangle_{\mathcal{F}}] = \langle I[f], g \otimes g\rangle_{\mathcal{F}\otimes\mathcal{F}}$$

and more generally, the Fisher information between $\langle f, g\rangle_{\mathcal{F}}$ and $\langle f, g'\rangle_{\mathcal{F}}$ is given as

$$I[\langle f, g\rangle_{\mathcal{F}}, \langle f, g'\rangle_{\mathcal{F}}] = \langle I[f], g \otimes g'\rangle_{\mathcal{F}\otimes\mathcal{F}}$$

The proof of Lemma 2.3 will not be shown here, but in involves the use of Parseval's identity in an inner product space. Using Lemma 2.3, we can derive the Fisher information for our regression function as defined in (**??**) subject to f belonging to a RKHS.

Corollary 2.3.1 (Fisher information for regression function)**.** *For our regression model as defined in* (**??**) *subject to f belonging to a RKHS $\mathcal{F}$, the Fisher information $I[f(x), f(x')]$ is given by*

$$I[f(x), f(x')] = \sum_{i=1}^{n} \sum_{j=1}^{n} \psi_{ij} h(x, x_i) h(x', x_j).$$

Proof. Note that in a RKHS $\mathcal{F}$, the reproducing property gives $f(x) = \langle f, h(\cdot, x)\rangle_{\mathcal{F}}$ and in particular, $\langle h(\cdot, x), h(\cdot, x')\rangle_{\mathcal{F}} = h(x, x')$. By Lemma 2.3, we have

$$\begin{aligned} I[f(x), f(x')] &= I[\langle f, h(\cdot, x)\rangle_{\mathcal{F}}, \langle f, h(\cdot, x')\rangle_{\mathcal{F}}] \\ &= \big\langle I[f], h(\cdot, x) \otimes h(\cdot, x')\big\rangle_{\mathcal{F}\otimes\mathcal{F}} \\ &= \left\langle \sum_{i=1}^{n} \sum_{j=1}^{n} \psi_{ij} h(\cdot, x_i) \otimes h(\cdot, x_j) \ , \ h(\cdot, x) \otimes h(\cdot, x') \right\rangle_{\mathcal{F}\otimes\mathcal{F}} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \psi_{ij} \big\langle h(\cdot, x_i), h(\cdot, x)\big\rangle_{\mathcal{F}} \big\langle h(\cdot, x_j), h(\cdot, x')\big\rangle_{\mathcal{F}} \\ &\quad \text{(by using the fact that inner products are linear, and that } \forall a_1, a_2 \in \mathcal{A} \\ &\quad \text{and } \forall b_1, b_2 \in \mathcal{B}, \ \langle a_1 \otimes b_1, a_2 \otimes b_2\rangle_{\mathcal{A}\otimes\mathcal{B}} = \langle a_1, a_2\rangle_{\mathcal{A}} \langle b_1, b_2\rangle_{\mathcal{B}}) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \psi_{ij} h(x, x_i) h(x', x_j). \quad \text{(by the reproducing property)} \end{aligned}$$

□

Note that any regression function $f \in \mathcal{F}$ can be decomposed into $f = f_n + r$, with $f \in \mathcal{F}_n$ and $r \in \mathcal{R}$ where $\mathcal{F} = \mathcal{F}_n + \mathcal{R}$ and $\mathcal{F}_n \perp \mathcal{R}$. Fisher information exists only on the n-dimensional subspace $\mathcal{F}_n$, while there is no information for $\mathcal{R}$. Thus, we will only ever consider the RKHS $\mathcal{F}_n \subset \mathcal{F}$ where there is Fisher information. Let h be a real symmetric and positive definite function over $\mathcal{X}$ defined by $h(x, x') = I[f(x), f(x')]$. As we saw earlier, h defines a RKHS, and it can be shown that the RKHS induced is in fact $\mathcal{F}_n$ spanned by the reproducing kernel on the dataset with the squared norm $||f||^2_{\mathcal{F}_n} = w^\top \Psi^{-1} w$.

2.3 The I-prior

For our linear model in (**??**) with f belonging to a RKHS $\mathcal{F}$ with kernel h over the set $\mathcal{X}$, define the subspace

$$\mathcal{F}_n = \left\{ f : \mathcal{X} \to \mathbb{R} \;\middle|\; f(x) = \sum_{i=1}^n h(x, x_i), \text{ for some } w_1, \dots, w_n \in \mathbb{R}, \text{ and } x \in \mathcal{X} \right\}$$

for which the Fisher information exists. Effectively, our functions $f \in \mathcal{F}_n$ are parameterized by $w = (w_1, \dots, w_n)^\top \in \mathbb{R}^n$, so we need only consider priors over $\mathbb{R}^n$. The entropy of a prior π relative to a Lebesgue measure over $\mathbb{R}^n$ is defined as

$$\mathrm{H}(\pi) = -\int_{\mathbb{R}^n} \pi(w) \log \pi(w) \, \mathrm{d}w.$$

Maximising this entropy subject to a suitable constraint gives us the I-prior definition.

Definition 2.13 (I-prior)**.** [**Bergsma2014**]. Let $\mathcal{F}$ be a Krein space and let $Y \in \mathbb{R}^n$ be a random variable whose distribution depends on $f \in \mathcal{F}$. Denote the Fisher information for f by $I[f]$, and suppose it exists. For a given $f_0 \in \mathcal{F}$, let π be a probability distribution independent of Y such that $\mathrm{Cov}_\pi(f) = I_{f_0}[f]$. Then π is called an I-prior for f with hyperparameter f_0.

Definition 2.14 (I-prior)**.** A prior π for f for the linear model in (**??**) with f belonging to a RKHS $\mathcal{F}$ is called an I-prior if $\pi(\mathcal{F}_n) = 1$, and conditionally on $f \in \mathcal{F}_n$,

$$\pi = \arg\max \mathrm{H}(\pi) \;\; \text{subject to} \;\; \mathrm{E}_\pi \, ||f||^2_{\mathcal{F}_n} = 1.$$

The following theorem associates I-priors with the Fisher information.

Theorem 2.4 (I-prior for linear models is Gaussian with mean f_0 and covariance matrix the Fisher information)**.** [**Bergsma2014**]. *Consider the linear model in* (**??**) *with f belonging to a RKHS $\mathcal{F}$ with kernel $h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Then an I-prior π for f is Gaussian with a hyperparameter f_0 (the prior mean) and covariance matrix as defined by*

$$\mathrm{Cov}_\pi(f(x), f(x')) = I[f(x), f(x')]$$

where

$$I[f(x), f(x')] = \sum_{i=1}^n \sum_{j=1}^n \psi_{ij} h(x, x_i) h(x', x_j)$$

is the Fisher information for f, and ψ_{ij} is the (i,j)-th entry of the precision matrix Ψ

of the errors. An I-prior for f will then have the random effect representation

$$f(x) = f_0(x) + \sum_{i=1}^{n} h(x, x_i) w_i$$
$$(w_1, \ldots, w_n) \sim \mathrm{N}(0, \Psi).$$

For convenience, we can write the I-prior for f in the more compact matrix notation

$$\mathbf{f} = \mathbf{f}_0 + \mathbf{H}\mathbf{w}$$
$$\mathbf{w} \sim \mathrm{N}(\mathbf{0}, \mathbf{\Psi})$$

where $\mathbf{H}$ is the $n \times n$ symmetric kernel matrix whose (i,j)-th entries contain $h(x_i, x_j)$, for $i, j = 1, \ldots, n$.

For the model defined in (**??**), an I-prior on f is a Gaussian distribution with prior mean f_0 and covariance matrix equal to the Fisher information for f. For this model, the Fisher information does not depend on f_0 and can be simply written as $I[f]$. We can also write the I-prior for f in a random effect representation, given by the following theorem:

Theorem 2.5 (I-prior for linear models)**.** [**Bergsma2014**]. *For the linear regression model stated in* (**??**)*, let $\mathcal{F}$ be the RKKS over $\mathcal{X}$ with kernel $h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The Fisher information $I[f] \in \mathcal{F} \otimes \mathcal{F}$ for f is given by*

$$I[f](\mathbf{x}_i, \mathbf{x}'_i) = \sum_{k=1}^{n} \sum_{l=1}^{n} \psi_{kl} h(\mathbf{x}_i, \mathbf{x}_k) h(\mathbf{x}'_i, \mathbf{x}_l)$$

where ψ_{kl} is the (k,l)-th entry of the precision matrix $\mathbf{\Psi}$ of the errors. Denote by π be the Gaussian distribution mean f_0 and covariance kernel $I[f]$. Then by definition, π is an I-prior for f. Thus, a random vector $f \sim \pi$ will have the covariance matrix as defined by $\mathrm{Cov}_\pi(f(\mathbf{x}_i), f(\mathbf{x}'_i)) = I[f](\mathbf{x}_i, \mathbf{x}'_i)$, and that the I-prior for f will have the random effect representation

$$f(\mathbf{x}_i) = f_0(\mathbf{x}_i) + \sum_{k=1}^{n} h(\mathbf{x}_i, \mathbf{x}_k) w_k$$
$$(w_1, \ldots, w_n) \sim N(\mathbf{0}, \mathbf{\Psi}).$$

For convenience, we can write the I-prior for f in the more compact matrix notation

$$\mathbf{f} = \mathbf{f}_0 + \mathbf{H}\mathbf{w}$$
$$\mathbf{w} \sim N(\mathbf{0}, \mathbf{\Psi})$$

where the boldface $\mathbf{f}$ represents the vector of functional evaluations $(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$, and $\mathbf{H}$ is the symmetric kernel matrix whose (i,j)-th entries contain $h(\mathbf{x}_i, \mathbf{x}_j)$.

The proof for this theorem can be found in **Bergsma2014** The prior mean f_0 is a hyperparameter of the I-prior model, and can be given a fixed value such as 0. **Bergsma2014** derives the closed form expression for the posterior distribution of the I-prior regression function f, for which the posterior mean is used as an estimate. An EM algorithm can be employed to find the maximum likelihood estimators of the hyperparameters of the I-prior model, or alternatively the random effects can be integrated out and the marginal likelihood maximised directly. These consist of the intercept α, the error precision Ψ, and any other parameters that the kernel may depend on (more on this in Section 2.4.1). While the intercept can be viewed as being part of the regression function f (technically, it would be a function in the RKHS of constant functions), practically it is much easier to treat it as a separate fixed parameter to be estimated. Hence the reason for segregating the intercept from the regression function in our models thus far.

2.3.1 Example of I-prior modelling: Multiple regression

Now let us take a look at an example of regression modelling with I-priors on the familiar standard linear model as described in (8.1). For this model, we can compute the Fisher information for the regression coefficients $\boldsymbol{\beta}$, by twice differentiating the log-likelihood function and taking negative expectations. This is found to be

$$I[\boldsymbol{\beta}] = \psi \mathbf{X}^\top \mathbf{X}.$$

Thus, an I-prior for $\boldsymbol{\beta}$ with prior mean $\boldsymbol{\beta}_0$ is

$$\boldsymbol{\beta} \sim \mathrm{N}(\boldsymbol{\beta}_0, \psi \mathbf{X}^\top \mathbf{X}).$$

An equivalent way of writing this I-prior would be

$$\begin{gathered} \boldsymbol{\beta} = \boldsymbol{\beta}_0 + \mathbf{X}^\top \mathbf{w} \\ \mathbf{w} \sim \mathrm{N}(\mathbf{0}, \psi \mathbf{I}_n) \end{gathered}$$

where $\mathbf{w} = (w_1, \dots, w_n)$ are the so called I-prior random effects as described in the second part of Theorem 2.5 above. Substituting the above back into model (8.1) we arrive at the I-prior random effects representation

$$\begin{gathered} \mathbf{y} = \boldsymbol{\alpha} + \overbrace{\mathbf{X}\boldsymbol{\beta}_0}^{\mathbf{f}_0} + \overbrace{\mathbf{X}\mathbf{X}^\top \mathbf{w}}^{\mathbf{Hw}} + \boldsymbol{\epsilon} \\ \boldsymbol{\epsilon} \sim \mathrm{N}(\mathbf{0}, \psi^{-1} \mathbf{I}_n) \\ \mathbf{w} \sim \mathrm{N}(\mathbf{0}, \psi \mathbf{I}_n). \end{gathered} \tag{2.3}$$

Remark 1. The multiple regression model relates to the I-prior methodology by considering the regression function $f(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\beta}$, for some $\boldsymbol{\beta} \in \mathbb{R}^p$. Lemma **??** tells us the form of the Fisher information for f, while Theorem 2.5 sets the I-prior for f as Gaussian with prior mean f_0 and covariance matrix the Fisher information. Deriving the I-prior this way gives similar results to the above.

2.4 Kernel functions

2.4.1 Other commonly used models: A toolbox of kernels

In the above multiple regression example, the regression function are straight line functions over the set of reals. This is a reproducing kernel space with the Euclidean space inner product/dot product as its kernel, i.e. $h(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle = \mathbf{x}_i \cdot \mathbf{x}_j$, which is what makes up the entries of $\mathbf{H}$ in (2.3). This is known as the Canonical kernel (**Bergsma2014**). As it turns out, exchanging this canonical kernel with a different kernel, hence a different RKHS of functions, we can perform various types of modelling. Some commonly used models can be achieved using the following kernels:

Type	Description of $\mathcal{X}$=$\{x_i\}$	Name of space	Kernel $h(x_i, x_j)$
Nominal	1) Categorical covariates; 2) In a multilevel setting, x_i = group no. of unit i.	Pearson	$\frac{\mathbb{1}[x_i=x_j]}{p_i} - 1$ where $p_i = \P[X = x_i]$
Real	In a classical regression setting, x_i = covariate associated with unit i.	Canonical / Centred Canonical	$x_i x_j$ / $x_i x_j - \bar{x}$
Real	In 1-dim smoothing, x_i = data point associated with observation y_i.	Fractional Brownian Motion (FBM)	$\lvert x_i \rvert^{2\gamma} + \lvert x_j \rvert^{2\gamma} - \lvert x_i - x_j \rvert^{2\gamma}$ with Hurst index $\gamma \in (0, 1)$

Table 2.1: A toolbox of kernels - Names and descriptions of some useful RKHS of functions.

Remark 2. The origin of a Hilbert space over a set $\mathcal{X}$ may be arbitrary, in which case a centering may be appropriate. Hence, the centred Canonical kernel.

New reproducing kernel spaces can be constructed from existing ones. An example is the so-called ANOVA kernel constructed from a Canonical and Pearson kernel applied to the two-dimensional vector $\mathbf{x}_i = (x_{i1}, x_{i2})$, where the first component is real-valued, and the second component consists of nominal values. The ANOVA kernel is constructed as

$$h(\mathbf{x}_i, \mathbf{x}_j) = h_1(x_{i1}, x_{j1}) + h_2(x_{i2}, x_{j2}) + h_1(x_{i1}, x_{j1}) h_2(x_{i2}, x_{j2}).$$

This kernel is particularly useful to model interaction effects. Take for example a random slope model. The effect of a covariate is assumed to be different for each group. This can be thought of as having an interaction present between the real-valued covariate x_{i1} and the grouping x_{i2}, which is captured by the product of the two kernels $h_1 h_2$.

Remark 3. We are able to circumvent the positive definite restriction of inner products (and kernels which define them in the reproducing kernel space) by working in a Krein space, and hence a reproducing kernel Krein space (RKKS). Krein spaces generalise Hilbert spaces by dropping the positive-definiteness requirement of inner products. Inner products may turn out to be not positive definite when scale parameters for the space are considered (which may be negative) and new kernels are constructed by way of adding and multiplying kernels together, as in the ANOVA kernel above. For a review of RKKSs, see **alpay1991** and **Ong2004** RKKSs are actively being researched, and is out of the scope of this paper for now.

2.4.2 The RKHS scale parameter

The scale of an RKHS $\mathcal{F}$ over a set $\mathcal{X}$ with kernel $h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ may be arbitrary. To resolve this, a scale parameter $\lambda \in \mathbb{R}$ is introduced, resulting in the RKHS denoted by $\mathcal{F}_\lambda$ with kernel $h_\lambda = \lambda h$. This results in at most p scale parameters $\lambda_1, \dots, \lambda_p$ - one for each of the function space over the set of p covariates. If there are several covariates which are known to be measured on the same scale, e.g. repeated measures of weight, then these may share the same scale parameter (technically, the same RKHS $\mathcal{F}_\lambda$).

Remark 4. For the ANOVA kernel described above, there are two possible ways of introducing scale parameters. Since the ANOVA kernel is constructed from two existing kernels, the Canonical kernel h_1 and Pearson kernel h_2, each with their own scale parameter λ_1 and λ_2 respectively, then the interaction effect or the product between the two kernels has the scale parameter equal to the product of the two scale parameters $\lambda_1\lambda_2$. This is the more parsimonious method. Another valid way is to introduce a separate scale parameter for the interactions, λ_{12} say. This is the less parsimonious method, and in this case, there will be at most $p(p-1)/2$ scale parameters when a model with p covariates and all its two-way interactions are considered.

In the example of multiple regression in Section 2.3.1, the canonical kernel with scale parameters $\lambda_1, \dots, \lambda_p$ can be written as

$$\mathbf{H} = \mathbf{H}_{\boldsymbol{\lambda}} := \mathbf{X}\boldsymbol{\Lambda}\mathbf{X}^\top$$

where $\boldsymbol{\Lambda} = \text{diag}[\lambda_1, \dots, \lambda_p]$. This corresponds to the scaled Canonical RKHS with kernel $h_\lambda(\mathbf{x}_i, \mathbf{x}_j) = \lambda_1 x_{i1} x_{j1} + \cdots + \lambda_p x_{ip} x_{jp}$, and the covariance matrix for the I-prior on $\boldsymbol{\beta}$ is adjusted to be $\psi\boldsymbol{\Lambda}\mathbf{X}^\top\mathbf{X}\boldsymbol{\Lambda}$.

Proof. In the I-prior method, our model is $\mathbf{y} = \boldsymbol{\alpha} + \mathbf{H}\mathbf{w} + \boldsymbol{\epsilon}$ where f_0 has been assumed to be zero for simplicity. Replacing the canonical kernel matrix $\mathbf{H}$ with the scaled canonical kernel matrix, we have

$$\mathbf{y} = \boldsymbol{\alpha} + \mathbf{X}\overbrace{\boldsymbol{\Lambda}\mathbf{X}^\top\mathbf{w}}^{\beta} + \boldsymbol{\epsilon}$$

with $\mathbf{w} \sim \mathrm{N}(\mathbf{0}, \psi \mathbf{I}_n)$. Equivalently, $\boldsymbol\beta$ is normally distributed with mean and variance

$$\mathrm{E}\,\boldsymbol\beta = \mathrm{E}[\boldsymbol\Lambda \mathbf{X}^\top \mathbf{w}] = \mathbf{0}$$

and

$$\mathrm{Var}\,\boldsymbol\beta = \mathrm{Var}[\boldsymbol\Lambda \mathbf{X}^\top \mathbf{w}] = \psi \boldsymbol\Lambda \mathbf{X}^\top \mathbf{X} \boldsymbol\Lambda.$$

□

2.5 Comparison to Gaussian process priors

Key differences:

1. Typically no scale parameter is estimated for the kernels in GPR. Instead, the x and y variables are centred *and* scaled before estimating. New data points are then centred and scaled on the mean and s.d. of the training points.

2. GPR not usually interested in estimating the error precision ψ.

3. The "go-to" kernel is the squared exponential kernel or Gaussian radial basis function defined as
$$k(x, x') = \exp(-\sigma \|x - x'\|^2)$$
σ usually chosen by cross-validation or grid-search methods.

Why do we need to estimate scale parameters and error precision in I-prior models?

2.5.1 The Bayesian connection

The I-prior methodology is less of a fully Bayesian approach and more of an empirical-Bayes approach, whereby an objective using the Fisher information as the covariance matrix of the prior is used to estimate the parameters of the model through maximisation of the likelihood, set up in a RKHS paradigm. However, the I-prior methodology is still this notion of priors and posteriors, something which is arguably Bayesian. Recall the standard linear regression model with independent errors:

$$\mathbf{y} = \boldsymbol\alpha + \mathbf{X}\boldsymbol\beta + \boldsymbol\epsilon$$
$$\boldsymbol\epsilon \sim \mathrm{N}(\mathbf{0}, \psi^{-1}\mathbf{I}_n).$$

The I-prior method transformed this model into the random effect representation with kernels that we saw earlier in Section 2.3.1. However, by simply taking the fundamental idea of I-priors, which is a prior with the covariance matrix equal to the Fisher information, nothing is really stopping us from estimating this model fully Bayes. We simply need to assign further priors on the intercept and precision, such as

$$\underline{\text{Priors}}$$
$$\boldsymbol{\beta} \sim \text{N}(\mathbf{0}, \psi\boldsymbol{\Lambda}\mathbf{X}^\top\mathbf{X}\boldsymbol{\Lambda})$$
$$\alpha \sim \text{N}(0, 1000)$$
$$\psi, \lambda_1^{-2}, \dots, \lambda_p^{-2} \sim \Gamma(0.001, 0.001).$$

Here, an I-prior with mean zero is chosen. The choices of normal for α, and gamma for the scale parameters ψ and a reparameterization of the λs is chosen for conjugacy convenience. In the absence of any prior knowledge about the parameters, it is reasonable to choose such hyperparameters to make the priors quite flat and uninformative. Another choice of uninformative prior for the scale parameters would be the **Jeffreys1946**' prior, which is in fact the limit of the gamma distribution as both hyperparameters approach zero. An MCMC approach such as Gibbs or Metropolis-Hastings sampling is then able to estimate this model, and software such as WinBUGS or JAGS are then able to be used.

The main motivation behind I-priors was to guard against over-fitting in cases where model dimensionality is very large relative to sample size. A prior is devised based on an objective principle (of maximum entropy) which brings about simpler estimation while requiring minimal assumptions, as well as model parsimony. A maximum likelihood approach is used to fit I-prior models, which give promising results in terms of predictive abilities from the simulations conducted. In the next section, I-priors will be discussed with a more Bayesian connotation, applied to Bayesian variable selection.

Chapter 3

Estimation and computational methods for I-prior models

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Theorem 3.1 (Euclid)**.** *For every prime p, there is a prime $p' > p$. In particular, the list of primes,*

$$2, 3, 5, 7, \dots \tag{3.1}$$

is infinite.

Proof. My proof is complete. □

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Lemma 3.2 (Something)**.** *For every prime p, there is a prime $p' > p$. In particular, there are infinitely many primes.*

Remark 5. Actually, this is a remark.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Corollary 3.2.1 (Anything)**.** *This is my corollary.*

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

A restatement of the theorem is as follows.

Corollary 3.2.1 (Anything)**.** *This is my corollary.*

Definition 3.1 (Apple)**.** An apple is a fruit.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Theorem 3.3 (Keyed theorem)**.** *This is a key-val theorem.*

Theorem 3.3 (continuing from p. 25)**.** *And it's spread out.*

3.1 Direct maximisation

3.2 EM algorithm

3.2.1 EM Algorithm

Substituting (**??**) into (**??**), we can rewrite the I-prior model in a "random-effects" representation. Using matrix notation, we have

$$\begin{aligned} \mathbf{y} &= \boldsymbol{\alpha} + \mathbf{H}_\lambda \mathbf{w} + \boldsymbol{\epsilon} \\ \mathbf{w} &:= (w_1, \dots, w_n) \sim \mathrm{N}(\mathbf{0}, \psi \mathbf{I}_n) \\ \boldsymbol{\epsilon} &:= (\epsilon_1, \dots, \epsilon_n) \sim \mathrm{N}(\mathbf{0}, \psi^{-1} \mathbf{I}_n) \end{aligned} \tag{3.2}$$

where the intercept $\boldsymbol{\alpha} = \alpha \mathbf{1}_n$ has been separated from the regression function. Here, $\mathbf{1}_n$ is a vector of length n containing all ones, and $\mathbf{H}_\lambda$ is the matrix whose (i,j)th entries are $h_\lambda(x_i, x_j)$, where h_λ is the (scaled) reproducing kernel over the set of covariates. An EM algorithm can be applied by treating the random effects $w_1, \dots, w_n$ as 'missing' in order to estimate the parameters α, $\boldsymbol{\lambda}$ and ψ. The assumption of normality also makes the EM algorithm particularly appealing as the required joint and conditional distributions are easy to obtain. To start, write $\mathbf{y} \sim \mathrm{N}(\boldsymbol{\alpha}, \mathbf{V}_y)$, where $\mathbf{V}_y = \psi \mathbf{H}_\lambda^2 + \psi^{-1} \mathbf{I}_n$ (the marginal distribution of the responses). Given the random effects $\mathbf{w}$, the distribution of $\mathbf{y}|\mathbf{w}$ is also multivariate normal with mean $\boldsymbol{\alpha} + \mathbf{H}_\lambda$ and covariance matrix $\psi^{-1} \mathbf{I}_n$.

The covariance between $\mathbf{y}$ and $\mathbf{w}$ is

$$\begin{aligned} \mathrm{Cov}[\mathbf{y}, \mathbf{w}] &= \mathrm{Cov}[\boldsymbol{\alpha} + \mathbf{H}_\lambda \mathbf{w} + \boldsymbol{\epsilon}, \mathbf{w}] \\ &= \cancelto{0}{\mathrm{Cov}[\boldsymbol{\alpha}, \mathbf{w}]} + \mathbf{H}_\lambda \cancelto{\psi \mathbf{I}_n}{\mathrm{Cov}[\mathbf{w}, \mathbf{w}]} + \cancelto{0}{\mathrm{Cov}[\boldsymbol{\epsilon}, \mathbf{w}]} \\ &= \psi \mathbf{H}_\lambda := \mathrm{Cov}[\mathbf{w}, \mathbf{y}]. \end{aligned}$$

Thus, the joint distribution of $(\mathbf{y}, \mathbf{w})$ is

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{w} \end{pmatrix} \sim \mathrm{N} \left(\begin{pmatrix} \boldsymbol{\alpha} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{V}_y & \psi \mathbf{H}_\lambda \\ \psi \mathbf{H}_\lambda & \psi \mathbf{I}_n \end{pmatrix} \right).$$

Using standard results of multivariate normal distributions (see, for example, **krzanowski2000principle** the conditional distribution of $\mathbf{w}$ given $\mathbf{y}$ is normal with mean and variance

$$\begin{aligned}
\mathrm{E}[\mathbf{w}|\mathbf{y}] &= \mathrm{E}[\mathbf{w}] + \mathrm{Cov}[\mathbf{w},\mathbf{y}]\ \mathrm{Var}[\mathbf{y}]^{-1}\,(\mathbf{y} - \mathrm{E}[\mathbf{y}]) \\
&= \psi\mathbf{H}_\lambda\mathbf{V}_y^{-1}(\mathbf{y} - \boldsymbol{\alpha}) =: \tilde{\mathbf{w}} \\
\mathrm{Var}[\mathbf{w}|\mathbf{y}] &= \mathrm{Var}[\mathbf{w}] + \mathrm{Cov}[\mathbf{w},\mathbf{y}]\ \mathrm{Var}[\mathbf{y}]^{-1}\ \mathrm{Cov}[\mathbf{y},\mathbf{w}] \\
&= \psi\mathbf{I}_n - \psi^2\mathbf{H}_\lambda\mathbf{V}_y^{-1}\mathbf{H}_\lambda \\
&= \mathbf{V}_y^{-1} =: \tilde{\mathbf{V}}_w
\end{aligned} \tag{3.3}$$

where the last equality in the derivation of the conditional variance is obtained using the Woodbury matrix identity. Also, write the second posterior moment of the random effects as

$$\begin{aligned}
\mathrm{E}[\mathbf{w}\mathbf{w}^\top|\mathbf{y}] &= \mathrm{Var}[\mathbf{w}|\mathbf{y}] + \mathrm{E}[\mathbf{w}|\mathbf{y}]\,\mathrm{E}[\mathbf{w}|\mathbf{y}]^\top \\
&= \tilde{\mathbf{V}}_w + \tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top =: \tilde{\mathbf{W}}.
\end{aligned} \tag{3.4}$$

From the marginal distribution of the responses, we notice that the mean and variance parameters are separable (i.e., they are not dependent on each other). It is straightforward to see that the maximum likelihood estimate for α is $\hat\alpha = \bar y = \sum_{i=1}^n y_i/n$. We can use this fact and treat the intercept parameter as known.

With g denoting the relevant density functions, the complete data log-likelihood of $\boldsymbol{\lambda}$ and ψ is given by

$$\begin{aligned}
l(\boldsymbol{\lambda},\psi|\mathbf{y},\mathbf{w}) &= \log g(\mathbf{y},\mathbf{w};\hat\alpha,\boldsymbol{\lambda},\psi) \\
&= \log g(\mathbf{y};\hat\alpha,\boldsymbol{\lambda},\psi) + \log g(\mathbf{w};\psi) \\
&= -n\log 2\pi - \cancel{\frac{1}{2}\log|\psi^{-1}\mathbf{I}_n|} - \frac{\psi}{2}\|\mathbf{y} - \hat{\boldsymbol{\alpha}} - \mathbf{H}_\lambda\mathbf{w}\|^2 \cancel{- \frac{1}{2}\log|\psi\mathbf{I}_n|} - \frac{\psi}{2}\mathbf{w}^\top\mathbf{w} \\
&= -n\log 2\pi - \frac{\psi}{2}\|\mathbf{y} - \hat{\boldsymbol{\alpha}}\|^2 - \frac{\psi}{2}\mathbf{w}^\top\mathbf{H}_\lambda^2\mathbf{w} + \psi(\mathbf{y} - \hat{\boldsymbol{\alpha}})^\top\mathbf{H}_\lambda\mathbf{w} - \frac{1}{2\psi}\mathbf{w}^\top\mathbf{w} \\
&= -n\log 2\pi - \frac{\psi}{2}\|\mathbf{y} - \hat{\boldsymbol{\alpha}}\|^2 - \frac{1}{2}\mathbf{w}^\top(\underbrace{\psi\mathbf{H}_\lambda^2 + \psi^{-1}\mathbf{I}_n}_{\mathbf{V}_y})\mathbf{w} + \psi(\mathbf{y} - \hat{\boldsymbol{\alpha}})^\top\mathbf{H}_\lambda\mathbf{w} \\
&= -n\log 2\pi - \frac{\psi}{2}\|\mathbf{y} - \hat{\boldsymbol{\alpha}}\|^2 - \frac{1}{2}\operatorname{tr}\left(\mathbf{V}_y\mathbf{w}\mathbf{w}^\top\right) + \psi(\mathbf{y} - \hat{\boldsymbol{\alpha}})^\top\mathbf{H}_\lambda\mathbf{w}.
\end{aligned}$$

The EM algorithm at iteration $t \in \{0,1,\dots\}$ entails taking the expectation of the above complete data log-likelihood under $\mathbf{w}$ (the E-step, conditional on the responses $\mathbf{y}$ and some parameter values $(\boldsymbol{\lambda}^{(t)},\psi^{(t)})$. Making use of the results in (3.3) and (3.4), we denote the tth iteration E-step by the function

$$\begin{aligned}
Q(\boldsymbol\lambda, \psi) &= \operatorname{E}_{\mathbf{w}}\Big[l(\boldsymbol\lambda, \psi | \mathbf{y}, \mathbf{w}) \,\Big|\, \mathbf{y}, \boldsymbol\lambda^{(t)}, \psi^{(t)} \Big] \\
&= -n \log 2\pi - \frac{\psi}{2} \Vert \mathbf{y} - \hat{\boldsymbol\alpha} \Vert^2 - \frac{1}{2} \operatorname{tr}\left(\mathbf{V}_y \tilde{\mathbf{W}}^{(t)} \right) + \psi(\mathbf{y} - \hat{\boldsymbol\alpha})^\top \mathbf{H}_\lambda \tilde{\mathbf{w}}^{(t)}.
\end{aligned}$$

The M-step entails maximizing this Q function with respect to the parameters, which then boils down to solving the system of differential equations

$$\frac{\partial Q(\boldsymbol\lambda, \psi)}{\partial \lambda_k} = -\frac{1}{2} \operatorname{tr}\left(\frac{\partial \mathbf{V}_y}{\partial \lambda_k} \tilde{\mathbf{W}}^{(t)} \right) + \psi(\mathbf{y} - \hat{\boldsymbol\alpha})^\top \frac{\partial \mathbf{H}_\lambda}{\partial \lambda_k} \tilde{\mathbf{w}}^{(t)} \qquad (3.5)$$

$$\frac{\partial Q(\boldsymbol\lambda, \psi)}{\partial \psi} = -\frac{1}{2} \Vert \mathbf{y} - \hat{\boldsymbol\alpha} \Vert^2 - \operatorname{tr}\left(\frac{\partial \mathbf{V}_y}{\partial \psi} \tilde{\mathbf{W}}^{(t)} \right) + (\mathbf{y} - \hat{\boldsymbol\alpha})^\top \mathbf{H}_\lambda \tilde{\mathbf{w}}^{(t)} \qquad (3.6)$$

equated to zero for $k = 1, \dots, p$. The algorithm is made simpler by first conditioning on a value of ψ to obtain updated values for $\boldsymbol\lambda$, and then conditioning on these $\boldsymbol\lambda$ values to obtain an update for ψ. Given some starting values $(\boldsymbol\lambda^{(0)}, \psi^{(0)})$, the E-step and the M-step are iterated until convergence is obtained. A practical stopping criterion would be when there is no longer a sizeable increase in the marginal log-likelihood value, i.e., iterate until

$$\log g(\mathbf{y}; \boldsymbol\lambda^{(t+1)}, \psi^{(t+1)}) - \log g(\mathbf{y}; \boldsymbol\lambda^{(t)}, \psi^{(t)}) < \delta.$$

for some small value $\delta \in \mathbb{R}$.

For models with iid errors, such as the ones we are considering in this paper, the solution for ψ in (3.6) has the closed-form expression

$$\psi^{(t+1)} = \frac{\operatorname{tr}(\tilde{\mathbf{W}}^{(t)})}{\Vert \mathbf{y} - \hat{\boldsymbol\alpha} \Vert^2 + \operatorname{tr}(\mathbf{H}_\lambda^2 \tilde{\mathbf{W}}) - 2(\mathbf{y} - \hat{\boldsymbol\alpha})^\top \mathbf{H}_\lambda \tilde{\mathbf{w}}}.$$

This is generally not the case for the scale parameters $\boldsymbol\lambda$, but even so, the system of equations can still be solved using numerical methods. In the next section, we describe cases when closed-form solutions exists for $\boldsymbol\lambda$, and how to derive them.

3.2.2 Estimation of the scale parameters

As long as (A) there are no covariates involving square, cubic or any other higher order terms; and (B) the maximum order of interactions between all covariates is two, then the solution to the M-step in (3.5) involving $\boldsymbol\lambda$ can be found to be in closed-form. This includes models with either a single or multiple scale parameter(s) with two-way interactions between some or all of the terms. For any other models such as ones involving squared terms and three-way interactions, the M-step can still be solved using numerical methods such as a downhill simplex method. We proceed under the assumptions (A) and (B).

Assume further that there are p covariates, and each of the p kernel matrices $\mathbf{H}_1, \dots, \mathbf{H}_p$ for the covariates are calculated (depending on whether the data is continuous or nominal). If two-way interactions are present between any $k, j \in \{1, \dots, p\}$, then these are also calculated as $\mathbf{H}_{kj} = \mathbf{H}_k \circ \mathbf{H}_j$ (the Hadamard product).

Let the number of unique scale parameters be p, i.e., one for each covariate. While the number of scale parameters could actually be less than p, implying that some of the covariates share a scale parameter. This can be thought of as a multi-dimensional covariate. In any case, for a group of such covariates, the kernel matrix is simply the sum of each of the kernel matrices, and all we have to do is re-index everything based on the number of kernel matrices there are, and we are back to p (the number of kernel matrices).

In general, the scaled kernel matrix looks like

$$\mathbf{H}_\lambda = \sum_{k=1}^{p} \lambda_k \mathbf{H}_k + \sum_{k,j \in \mathcal{M}} \lambda_k \lambda_j \mathbf{H}_{kj}$$

where the set $\mathcal{M}$ is the index of all two way interaction terms between the p covariates, i.e., $\mathcal{M} = \{(k,j) : k \text{ interacts with } j, \text{ and } k < j, \ \forall k, j = 1, \dots, p\}$. Let the number of two-way interactions be $m = |\mathcal{M}|$. The total number of scale parameters is equal to $q = p + m$ when there are non-parsimonious interactions present, otherwise it is $q = p$. The non-parsimonious method of interactions assigns a new scale parameter for each of the Hadamard products of interacting kernel matrices[1]. In comparison, the parsimonious method multiplies the corresponding scale parameters together.

For a particular λ_k, $k = 1, \dots, q$, we partition the sum of the kernel matrix into parts which involve λ_k and parts which do not:

$$\begin{aligned}
\mathbf{H}_\lambda &= \overbrace{\lambda_k \mathbf{H}_k + \lambda_k \sum_{j \in \mathcal{M}} \lambda_j \mathbf{H}_{kj}}^{\lambda_k \text{ is here}} + \overbrace{\sum_{\substack{j=1 \\ j \neq k}}^{p} \lambda_j \mathbf{H}_j + \sum_{\substack{k',j \in \mathcal{M} \\ k' \neq k}} \lambda_{k'} \lambda_j \mathbf{H}_{k'j}}^{\text{no } \lambda_k \text{ here}} \\
&= \lambda_k \mathbf{P}_\mathbf{k} + \mathbf{R}_k + \mathbf{U}_k. \qquad (3.7)
\end{aligned}$$

$\mathbf{P}_k$ is the kernel matrix $\mathbf{H}_k$ plus the sum-product of the interaction kernel matrices with the scale parameters relating to covariate k, i.e., $\sum_j \lambda_j \mathbf{H}_{kj}$. $\mathbf{R}_k$ is the sum-product of the kernel matrices and scale parameters excluding $\lambda_k \mathbf{H}_k$. $\mathbf{U}_k$ is the sum of the interaction cross-product terms excluding those relating to covariate k. Thus, the squared

[1] The non-parsimonious method actually re-indexes both the kernel matrices and scale parameters from $\{1, \dots, p\}$ to $\{1, \dots, q\}$ where the Hadamard products are treated like “regular” kernel matrices, and the method proceeds as if there are no interactions present.

kernel matrix is

$$\begin{aligned}\mathbf{H}_\lambda^2 = \lambda_k^2\mathbf{P}_k^2 + \lambda_k \Big(\mathbf{P}_k\mathbf{R}_k + (\mathbf{P}_k\mathbf{R}_k)^\top + \mathbf{P}_k\mathbf{U}_k + (\mathbf{P}_k\mathbf{U}_k)^\top\Big) \\ + \mathbf{R}_k^2 + \mathbf{U}_k^2 + \mathbf{R}_k\mathbf{U}_k + (\mathbf{R}_k\mathbf{U}_k)^\top. \end{aligned} \tag{3.8}$$

The M-step from (3.5) for each of the λ_k now reduces to solving

$$\frac{\partial Q(\boldsymbol\lambda,\psi)}{\partial\lambda_k} = -\frac{\psi}{2}\operatorname{tr}\Big[\big(2\lambda_k\mathbf{P}_k^2 + \mathbf{S}_k\big)\,\tilde{\mathbf{W}}^{(t)}\Big] + \psi(\mathbf{y}-\hat{\boldsymbol\alpha})^\top\mathbf{P}_k\tilde{\mathbf{w}}^{(t)}$$

equal to zero, where we have defined $\mathbf{S}_k = \mathbf{P}_k\mathbf{R}_k + (\mathbf{P}_k\mathbf{R}_k)^\top + \mathbf{P}_k\mathbf{U}_k + (\mathbf{P}_k\mathbf{U}_k)^\top$. This yields the solution

$$\lambda_k^{(t+1)} = \frac{(\mathbf{y}-\hat{\boldsymbol\alpha})^\top\mathbf{P}_k\tilde{\mathbf{w}}^{(t)} - \frac{1}{2}\operatorname{tr}\big(\mathbf{S}_k\tilde{\mathbf{W}}^{(t)}\big)}{\operatorname{tr}\big(\mathbf{P}_k^2\tilde{\mathbf{W}}^{(t)}\big)}$$

for each $k = 1,\dots,p$.

For most cases, $\mathbf{P}_k$ and $\mathbf{S}_k$ only depend on the kernel matrices and not on the scale parameters, so can be calculated once and stored for efficiency. Further, $\mathbf{U}_k$ equals zero for most cases except in the parsimonious multiple scale parameter case thus simplifying calculations. In fact, we can avoid the expensive matrix multiplications involved in evaluating $\mathbf{P}_k$, its square, and $\mathbf{S}_k$, by calculating once and storing $\mathbf{H}_1,\dots,\mathbf{H}_p$, its squares and all possible two-way matrix multiplications of these kernel matrices, as the relevant calculation of the M-step merely involves a linear combination of these matrices. These operations are conducted by the `kernL()` function.

3.2.3 Calculation of inverse variance and the likelihood

Sometimes, the calculation of the inverse of $\mathbf{V}_y = \psi\mathbf{H}_\lambda + \psi^{-1}\mathbf{I}_n$ can become problematic, especially when ψ gets extremely large (or extremely small). Notice that $\mathbf{V}_y$ is of the form $\mathbf{A} + s\mathbf{I}_n$, where $\mathbf{A}$ is a symmetric and positive-definite matrix and s is a constant. An eigendecomposition of $\mathbf{A}$ yields $\mathbf{A} = \mathbf{V}\mathbf{U}\mathbf{V}^\top$, where $\mathbf{V}$ is the $n \times n$ matrix whose jth column is the normalised eigenvector $\mathbf{v}_j$ of $\mathbf{A}$ such that $\mathbf{V}\mathbf{V}^\top = \mathbf{I}_n$, and $\mathbf{U}$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues, i.e., $\mathbf{U}_{ii} = u_i$, for $i = 1,\dots,n$. Consider then the linear equation

$$\begin{aligned}(\mathbf{A} + s\mathbf{I}_n)\mathbf{a} &= (\mathbf{V}\mathbf{U}\mathbf{V}^\top + s\mathbf{V}\mathbf{V}^\top)\mathbf{a} \\ &= \mathbf{V}\operatorname{diag}(u_1{+}s,\dots,u_n{+}s)\,\mathbf{V}^\top\mathbf{a} =: \mathbf{b}.\end{aligned}$$

Solving for $\mathbf{a}$ yields

$$\mathbf{a} = \mathbf{V}\operatorname{diag}\left(\frac{1}{u_1{+}s},\dots,\frac{1}{u_n{+}s}\right)\mathbf{V}^\top\mathbf{b}. \tag{3.9}$$

With $\mathbf{A} = \psi\mathbf{H}_\lambda^2$ and $s = 1/\psi$, this is a much more stable way of computing $\mathbf{a} = \mathbf{V}_y^{-1}\mathbf{b}$, and is also useful for finding the inverse $\mathbf{a} = \mathbf{V}_y^{-1}$ by setting $\mathbf{b} = \mathbf{I}_n$.

This eigendecomposition is also used for a stable calculation of the log-likelihood. Firstly, the eigenvalues of $\mathbf{A} + s\mathbf{I}_n$ are simply $u_1+s, \dots, u_n+s$, and the log-determinant can be calculated as

$$\log|\mathbf{A} + s\mathbf{I}_n| = \sum_{i=1}^n \log(u_i + s).$$

Furthermore, the matrix $\mathbf{V}_y^{-1}(\mathbf{y} - \boldsymbol{\alpha})$ is given by $\mathbf{a}$ in equation (3.9) with $\mathbf{b} = (\mathbf{y} - \boldsymbol{\alpha})$. Thus, the marginal log-likelihood of the responses is given as

$$l(\alpha, \boldsymbol{\lambda}, \psi|\mathbf{y}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^n \log(u_i + s) - \frac{1}{2}(\mathbf{y} - \boldsymbol{\alpha})^\top\mathbf{a}.$$

3.2.4 Calculation of standard errors

Consider again the distribution $\mathbf{y} \sim \mathrm{N}(\boldsymbol{\alpha}, \mathbf{V}_y)$, where the mean and covariance matrix depends on different sets of parameters - namely α for the mean, and $\boldsymbol{\theta} = (\lambda_1, \dots, \lambda_p, \psi)$ for the covariance matrix. The Fisher information matrix $\mathcal{I}[\alpha, \boldsymbol{\theta}]$ then has the form

$$\mathcal{I}[\alpha, \boldsymbol{\theta}] = \mathrm{diag}\big(\mathcal{I}[\alpha], \mathcal{I}[\boldsymbol{\theta}]\big),$$

where

$$I[\alpha] = \frac{\partial\boldsymbol{\alpha}^\top}{\partial\alpha}\mathbf{V}_y^{-1}\frac{\partial\boldsymbol{\alpha}}{\partial\alpha} = \mathbf{V}_y^{-1} \circ \mathbf{J}_n$$

and

$$I[\boldsymbol{\theta}]_{ij} = \frac{1}{2}\operatorname{tr}\left(\mathbf{V}_y^{-1}\frac{\partial\mathbf{V}_y}{\partial\theta_i}\mathbf{V}_y^{-1}\frac{\partial\mathbf{V}_y}{\partial\theta_j}\right).$$

Recall that $\mathbf{V}_y = \psi\mathbf{H}_\lambda^2 + \psi^{-1}\mathbf{I}_n$ and that from (3.8), the derivative of the squared scaled kernel matrix has the form $\partial\mathbf{H}_\lambda^2/\partial\lambda_k = 2\lambda_k\mathbf{P}_k^2 + \mathbf{S}_k$. Therefore, the partial derivative of $\mathbf{V}_y$ with respect to λ_k for $k = 1, \dots, p$ is

$$\frac{\partial\mathbf{V}_y}{\partial\lambda_k} = \psi\left(2\lambda_k\mathbf{P}_k^2 + \mathbf{S}_k\right),$$

while the partial derivative of $\mathbf{V}_y$ with respect to ψ is

$$\begin{aligned}\frac{\partial \mathbf{V}_y}{\partial \psi} &= \mathbf{H}_\lambda^2 - \frac{1}{\psi^2}\mathbf{I}_n \\ &= \frac{1}{\psi}\left(\psi \mathbf{H}_\lambda^2 + \frac{1}{\psi}\mathbf{I}_n\right) - \frac{2}{\psi^2}\mathbf{I}_n \\ &= \frac{1}{\psi}\mathbf{V}_y - \frac{2}{\psi^2}\mathbf{I}_n.\end{aligned}$$

The Fisher information matrix can be obtained fairly inexpensively as the two matrices $\mathbf{P}_k^2$ and $\mathbf{S}_k$ have already been calculated through the EM algorithm (if the $\boldsymbol{\lambda}$ are in closed-form). Otherwise, there exist R functions to compute the Hessian of the (negative) log-likelihood numerically. The standard error for $\eta \in \{\alpha, \lambda_1, \dots, \lambda_p, \psi\} =: \mathcal{E}$ is then given by

$$\text{s.e.}(\eta) = \sqrt{\left(I[\alpha, \boldsymbol{\theta}]^{-1}\right)_{kk}},$$

where k is the index of the parameter in the set $\mathcal{E}$. The **iprior** package employs Wald tests of significance based off of these standard errors for the intercept and scale parameters in the `summary` of an `ipriorMod` object.

3.3 The EM algorithm pseudocode

Something I just realised...

Already have the eigendecomposition $\mathbf{H}_\lambda = \mathbf{V}\,\text{diag}(u_1, \dots, u_n)\mathbf{V}^\top$ with $\mathbf{V}^\top\mathbf{V} = \mathbf{I}_n$. Also, $\tilde{\mathbf{W}} = \mathbf{V}_y^{-1} + \tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top$, and $\mathbf{V}_y = \psi\mathbf{H}_\lambda^2 + \psi^{-1}\mathbf{I}_n$. Therefore$\mathbf{V}_y^{-1} = \mathbf{V}\,\text{diag}\left(\frac{1}{\psi u_i^2 + 1/\psi}\right)\mathbf{V}^\top$. Then,

$$\begin{aligned}\text{tr}(\mathbf{H}_\lambda^2\tilde{\mathbf{W}}) &= \text{tr}\left(\mathbf{H}_\lambda^2(\mathbf{V}_y^{-1} + \tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top)\right) \\ &= \text{tr}(\mathbf{H}_\lambda^2\mathbf{V}_y^{-1}) + \text{tr}(\mathbf{H}_\lambda^2\tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top) \\ &= \text{tr}\left(\mathbf{V}\,\text{diag}(u_i)\cancel{\mathbf{V}^\top\mathbf{V}}\,\text{diag}\left(\frac{1}{\psi u_i^2 + 1/\psi}\right)\mathbf{V}^\top\right) + \tilde{\mathbf{w}}^\top\mathbf{H}_\lambda\mathbf{H}_\lambda\tilde{\mathbf{w}} \\ &= \text{tr}\left(\cancel{\mathbf{V}^\top\mathbf{V}}\,\text{diag}\left(\frac{u_i}{\psi u_i^2 + 1/\psi}\right)\right) + \tilde{\mathbf{w}}^\top\mathbf{H}_\lambda\mathbf{H}_\lambda\tilde{\mathbf{w}} \\ &= \sum_{i=1}^n \frac{u_i}{\psi u_i^2 + 1/\psi} + \tilde{\mathbf{w}}^\top\mathbf{H}_\lambda\mathbf{H}_\lambda\tilde{\mathbf{w}}\end{aligned}$$

This way, no need to square any matrices.

Here is the pseudocode for the EM algorithm to estimate the I-prior model. An estimate of the percentage run time for some of the heavier computations are also given.

Algorithm 1 EM algorithm for the I-prior model

```
procedure Initialise (part of the Kernel Loader)
    Choose suitable $\boldsymbol{\lambda}^{(0)} = (\lambda_1^{(0)}, \dots, \lambda_l^{(0)})$ and $\psi^{(0)}$
    $t \leftarrow 0$
    $\hat{\alpha} \leftarrow \sum_{i=1}^n y_i/n$        ▷ The MLE for $\hat{\alpha}$
    $p \leftarrow$ no. of covariates
    $m \leftarrow$ no. of interactions
    $q \leftarrow$ no. of expanded scale parameters/kernel matrices $(p+m)$
    for $k = 1, \dots, p$ do
        Calculate $\mathbf{H}_k, \mathbf{H}_k^2$, and any Hadamard products (interactions).
        Calculate $\mathbf{P}_k, \mathbf{P}_k^2, \mathbf{S}_k$ using kernel matrices $\mathbf{H}_k$ and $\boldsymbol{\lambda}^{(0)}$.    ▷ time: 3.4%
    end for
    Index all the relevant kernel matrices from 1 to $q$.
end procedure

procedure Block A update (Iteration 0)
    Expand $\boldsymbol{\lambda}_{1:l}^{(t)} \rightarrow \boldsymbol{\lambda}_{1:q}^{(t)}$ depending on interactions and higher order terms
    $\mathbf{H}_\lambda \leftarrow \sum_{k=1}^q \boldsymbol{\lambda}_k^{(t)} \mathbf{H}_k$
    procedure Eigendecomposition        ▷ time: 51.6%
        $(\text{diag}(u_1, \dots, u_n), \mathbf{V}) \leftarrow \text{eigen}(\mathbf{H}_\lambda)$
    end procedure
    $s \leftarrow 1/\psi^{(t)}$
end procedure

function Linear solver and inverse (input $\mathbf{b}$)
    $\mathbf{a} \leftarrow \mathbf{V}\,\text{diag}\left[\frac{s}{u_1^2+s^2}, \dots, \frac{s}{u_n^2+s^2}\right] \mathbf{V}^\top \mathbf{b}$
end function

procedure Log-likelihood update
    call Linear solver and inverse (input $\mathbf{b} = \mathbf{y} - \hat{\boldsymbol{\alpha}}$)
    $l \leftarrow -\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^n \log(u_i^2/s + s) - \frac{1}{2}(\mathbf{y} - \boldsymbol{\alpha})^\top \mathbf{a}$
end procedure
```

```
procedure BLOCK B UPDATE ($t$)
    Update $\mathbf{P}_k$ and $\mathbf{S}_k$.        ▷ time: 15.6%
end procedure

while $l_{new} - l_{old} > \delta$ or $t < t_{max}$ do        ▷ The EM iterations
    procedure BLOCK C UPDATE (Iteration $t$)
        call LINEAR SOLVER AND INVERSE (input $\mathbf{b} = \mathbf{I}_n$)        ▷ time: 10.4%
        $\mathbf{V}_y^{-1} \leftarrow \mathbf{a}$
        $\tilde{\mathbf{w}} \leftarrow \psi^{(t)} \mathbf{H}_\lambda \mathbf{V}_y^{-1} (\mathbf{y} - \hat{\boldsymbol{\alpha}})$
        $\tilde{\mathbf{W}} \leftarrow \mathbf{V}_y^{-1} + \tilde{\mathbf{w}} \tilde{\mathbf{w}}^\top$
    end procedure

    procedure UPDATE FOR $\boldsymbol{\lambda}$: CLOSED-FORM EM
        for $k = 1, \dots, l$ do
            $T_1 \leftarrow \operatorname{tr} \left( \mathbf{P}_k^2 \tilde{\mathbf{W}} \right)$
            $T_2 \leftarrow (\mathbf{y} - \hat{\boldsymbol{\alpha}}) \mathbf{P}_k \tilde{\mathbf{w}} - \frac{1}{2} \operatorname{tr} \left( \mathbf{S}_k \tilde{\mathbf{W}} \right)$        ▷ time: 8.5%
            $\lambda_k^{(t+1)} \leftarrow T_2 / T_1$
        end for
    end procedure
    Note: If higher order terms and/or three-way (or more) interactions are present, then a numerical method is used.

    procedure UPDATE FOR $\psi$        ▷ time: 5.8%
        $T_3 \leftarrow (\mathbf{y} - \hat{\boldsymbol{\alpha}})^\top (\mathbf{y} - \hat{\boldsymbol{\alpha}}) + \operatorname{tr}(\mathbf{H}_\lambda^2 \tilde{\mathbf{W}}) - 2(\mathbf{y} - \hat{\boldsymbol{\alpha}})^\top \mathbf{H}_\lambda \tilde{\mathbf{w}}$
        $\psi^{(t+1)} \leftarrow \sqrt{\operatorname{tr}(\tilde{\mathbf{W}}) / T_3}$
    end procedure

    call BLOCK A UPDATE (Iteration $t + 1$)
    call LOG-LIKELIHOOD UPDATE
    $t \leftarrow t + 1$
end while

$(\hat{\boldsymbol{\lambda}}, \hat{\psi}) \leftarrow (\boldsymbol{\lambda}^{(t)}, \psi^{(t)})$        ▷ The maximum likelihood estimates
```

3.4 Markov chain Monte Carlo methods

3.5 Low-rank matrix approximation (Nyström method)

Chapter 4

Examples of I-prior models

4.1 Toy examples

Simple example showcasing canonical, fBm and Pearson kernel (and maybe some others? like polynomial and squared exponential). Good to compare a simulation scenario against regularisation and/or GPR.

4.2 Multilevel modelling

In this section, a comparison between a standard random effects model and the I-prior approach for estimating varying intercept and varying slopes model is illustrated. We consider a data set which accompanies the MLwiN software on the academic achievements of 4,059 pupils at 65 inner-London schools citeprasbash2012user, R2MLwiN. The response variable of interest are the pupils' (normalised) GCSE scores at age 16 encoded in the variable `normexam`. Also available in the data set is a pupil-specific regressor, which is the London reading test results (`standlrt`) for each pupil taken when they were aged 11.

```
R> data(tutorial, package = "R2MLwiN")
R> str(tutorial[, c("normexam", "school", "standlrt")])
```

```
## 'data.frame': 4059 obs. of  3 variables:
##  $ normexam: num  0.261 0.134 -1.724 0.968 0.544 ...
##  $ school  : Factor w/ 65 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1..
##  $ standlrt: num  0.619 0.206 -1.365 0.206 0.371 ...
```

First, we consider the varying intercept model. A standard approach to fitting this model is the random intercept model, which is based on the assumption that the intercepts are iid normal with zero mean. In R, packages such as **lme4** are able to fit these types of models. In the I-prior approach, the response variable normexam is regressed against the covariate school indicating the school that pupil had attended, which is assumed to have a nominal effect on GCSE scores. In other words, the regression function lies in the Pearson RKHS. As the variable school is a factor-type variable, the **iprior** package knows to treat this variable with the Pearson kernel automatically without user specification.

```
R> # Model 1: Varying intercept model
R> (mod1.fit <- iprior(normexam ~ school, data = tutorial))
```

(output omitted)

```
##
## Call:
## iprior(formula = normexam ~ school, data = tutorial)
##
## RKHS used: Pearson, with a single scale parameter.
##
## Parameter estimates:
##   (Intercept)        lambda           psi
## -0.0001139137  0.0006998747  1.1799071249
```

In Figure ??(a), the posterior means of the intercepts are plotted for the random effects model and the I-prior model. It can be seen that the estimates are in broad agreement, with conspicuously different estimates for schools 48 (-0.13 vs. -0.38) and 54 (-0.40 vs. -0.58), the I-prior giving the larger estimate in absolute values in both cases. The reason for this is that the I-prior variance for each school's regression function in a Pearson RKHS is inversely proportional to the sample size for that school. A proof of this is given in Appendix ??. Indeed, schools 48 and 54 have the smallest sample sizes of all schools, namely 2 and 8 respectively, whilst the next smallest is school 37 with 22 students.

Next we consider the varying slope model which regresses, for each school, the GCSE score on the results of the London reading test taken at age 11 (standlrt). A standard approach to fitting this model is the random intercept/slopes model, which is based on the assumption that the intercept/slope pairs are iid bivariate normal with zero means. To obtain an I-prior, we assume as above a nominal effect of school, and a linear effect of standlrt (using the canonical kernel on this variable). An interaction between the variables standlrt and school imply that the effect of the covariate standlrt varies with each school.

```
R> # Model 2: Varying slope model
R> (mod2.fit <- iprior(normexam ~ school * standlrt, data = tutorial))
```

(output omitted)

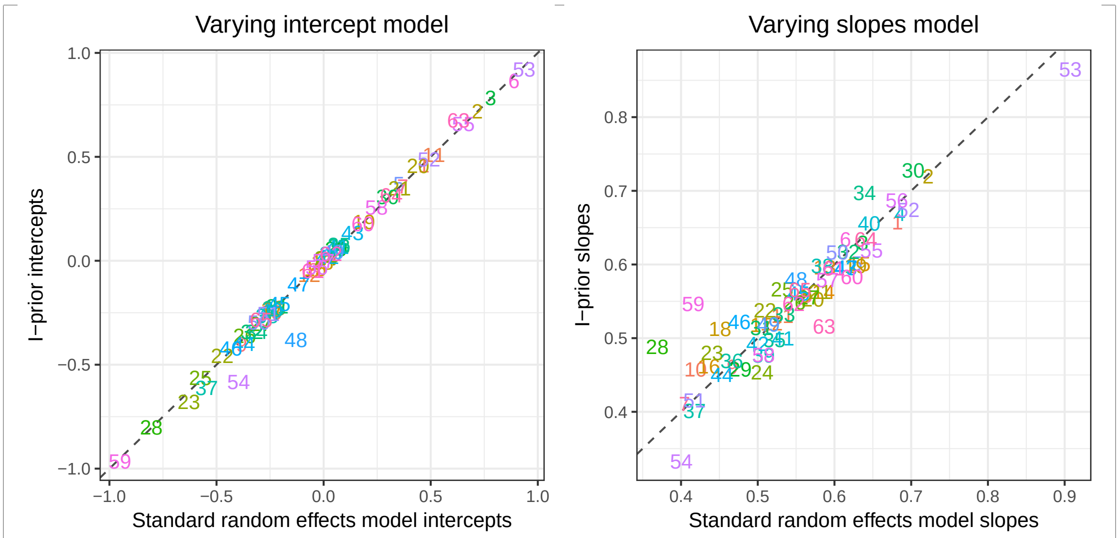


Figure 4.1: Estimated intercepts and slopes for school achievement data under (a) varying intercept (left); and (b) varying slope model. The numbers plotted are the school indices with the identity line for reference.

```
##
## Call:
## iprior(formula = normexam ~ school * standlrt, data = tutorial)
##
## RKHS used: Pearson & Canonical, with multiple scale parameters.
##
## Parameter estimates:
##   (Intercept)       lambda1       lambda2           psi
## -0.0001139137  0.0004234411  0.3731574626  1.8028198235
```

In Figure **??**(b), the posterior means of the slopes obtained using the standard random effects model are plotted against the ones obtained using the I-prior. Again, we see broad agreement of the estimates, but much less so than the varying intercept model.

A limited cross-validation study yielded on average a small advantage of the standard random effects approach in terms of mean squared error, in the order of half a percent, indicating the iid assumption in the random effects models is reasonable. However, an advantage of the I-prior is that no a priori assumption about the distribution of the parameters need to be made. Furthermore, our approach is more parsimonious and allows potentially simpler estimation and testing.

4.3 Longitudinal modelling

We consider a balanced longitudinal data set consisting of weights in kilograms of 60 cows, 30 of which were randomly assigned to treatment group A, and the remaining 30 to treatment group B. The animals were weighed 11 times over a 133-day period; the first 10 measurements for each animal were made at two-week intervals and the last measurement was made one week later. This experiment was reported by citekenward1987method, and the data set is included as part of the package **jmcm** in R.

```
R> data(cattle, package = "jmcm")
```

```
## # A tibble: 660 x 4
##         id  time  group weight
##     <fctr> <dbl> <fctr>  <int>
##   1      1     0      A    233
##   2      1    14      A    224
##   3      1    28      A    245
##   4      1    42      A    258
##   5      1    56      A    271
##   6      1    70      A    287
##   7      1    84      A    287
##   8      1    98      A    287
##   9      1   112      A    290
##  10      1   126      A    293
## # ... with 650 more rows
```

The response variable of interest are the `weight` growth curves, and the aim is to investigate whether a treatment effect is present. The usual approach to analyse a longitudinal data set such as this one is to assume that the observed growth curves are realizations of a Gaussian process. For example, citekenward1987method assumed a so-called ante-dependence structure of order k, which assumes an observation depends on the previous k observations, but given these, is independent of any preceeding observations.

Using the I-prior, it is not necessary to assume the growth curves were drawn randomly. Instead, it suffices to assume that they lie in an appropriate function class. For this example, we assume that the function class is the FBM RKHS, i.e., we assume a smooth effect of time on weight. The growth curves form a multidimensional (or functional) response equivalent to a “wide” format of representing repeated measures data. In our analysis using the **iprior** package, we used the “long” format and thus our (unidimensional) sample size n is equal to 60 cows × 11 repeated measurements. We also have two covariates potentially influencing growth, namely the cow subject `id` and also treatment `group`. The regression model can be thought of as

$$\texttt{weight} = f(\texttt{id}, \texttt{group}, \texttt{time}) + \text{error}.$$

We assume iid errors, and in addition to a smooth effect of `time`, we further assume a nominal effect of both cow `id` and treatment `group` using the Pearson RKHS. In the **iprior** package, factor type objects are treated with the Pearson kernel automatically, and the only `model` option we need to specify is the `kernel = "FBM"` option for the `time` variable. We have opted not to estimate the Hurst coefficient in the interest of computational time, and instead left it at the default value of 0.5. Table 4.1 explains the five models we have fitted.

Model	Explanation	Formula (`weight ~ ...`)
1	Growth does not vary with treatment nor among cows	`time`
2	Growth varies among cows only	`id * time`
3	Growth varies with treatment only	`group * time`
4	Growth varies with treatment and among cows	`id * time + group * time`
5	Growth varies with treatment and among cows, with an interaction effect between treatment and cow	`id * group * time`

Table 4.1: A brief description of the five models fitted using I-priors.

The simplest model fitted was one in which the growth curves do not depend on the treatment effect or individual cows. We then added treatment effect and the cow `id` as covariates, separately first and then together at once. We also assumed that both of these covariates are time-varying, and hence added also the interaction between these covariates and the `time` variable. The final model was one in which an interaction between treatment effect and individual cows was assumed, which varied over time.

All models were first loaded into a `ipriorKernel` object, and then fitted using the `ipriorOptim` function. Compared to the EM algorithm alone, we found that the combination of direct optimization with the EM algorithm in the `ipriorOptim` routine fits the model about six times faster for this data set due to slow convergence of EM algorithm. Here is the code and output for fitting the first model.

```
R> mod1 <- kernL(weight ~ time, data = cattle, model = list(kernel = "FBM"))
R> mod1.fit <- ipriorOptim(mod1)
```

```
## Iteration 0:    Log-likelihood = -40740.339
## Iteration 1:    Log-likelihood = -5416.4625 .......
## Iteration 2:    Log-likelihood = -3347.1579 .......
## Iteration 3:    Log-likelihood = -3096.7290 .......
## EM NOT CONVERGED!
##
```

```
## Now switching to optim...
##
## final  value 2789.600435
## converged
##
## Preparing iprior output... DONE.
```

The `iprior0ptim` routine (see Section ?? for details) performs three EM iterations from a random starting value of the parameters. After the initial EM steps, a direct optimization is carried out using R's built-in optimizer `optim()`. The user has several `control` options to choose from, such as specifying the number of initial EM steps to be performed.

Model	Formula (`weight ~ ...`)	Log-likelihood	Error S.D.	Number of λ parameters
1	`time`	-2789.60	16.22	1
2	`id * time`	-2792.15	16.18	2
3	`group * time`	-2295.16	3.68	2
4	`id * time + group * time`	-2270.85	3.39	3
5	`id * group * time`	-2250.88	3.77	3

Table 4.2: Summary of the five I-prior models fitted to the cow data set.

The results of the model fit are summarised in Table 4.2. We can test for a treatment effect by testing Model 4 against the alternative that Model 2 is true. The log-likelihood ratio test statistic is $D = -2(-2792.15 - (-2270.85)) = 1042.61$ which has an asymptotic chi-squared distribution with $3 - 2 = 1$ degree of freedom. The p-value for this likelihood ratio test is less than 10^{-6}, so we conclude that Model 4 is significantly better than Model 2.

We can next investigate whether the treatment effect differs among cows by comparing Model 5 against Model 4. As these models have the same number of parameters, we can simply choose the one with the higher likelihood, which is Model 5. We conclude that treatment does indeed have an effect on growth, and that the treatment effect differs among cows. We can use the `plot` function to plot the fitted regression curves onto the cow data set. This is shown in Figure 4.2.

4.4 Regression with functional covariates

We illustrate the prediction of a real valued response when one of the covariates is a function using a widely analysed data set for quality control in the food industry. The data[1]contain samples of spectrometric curve of absorbances of 215 pieces of finely

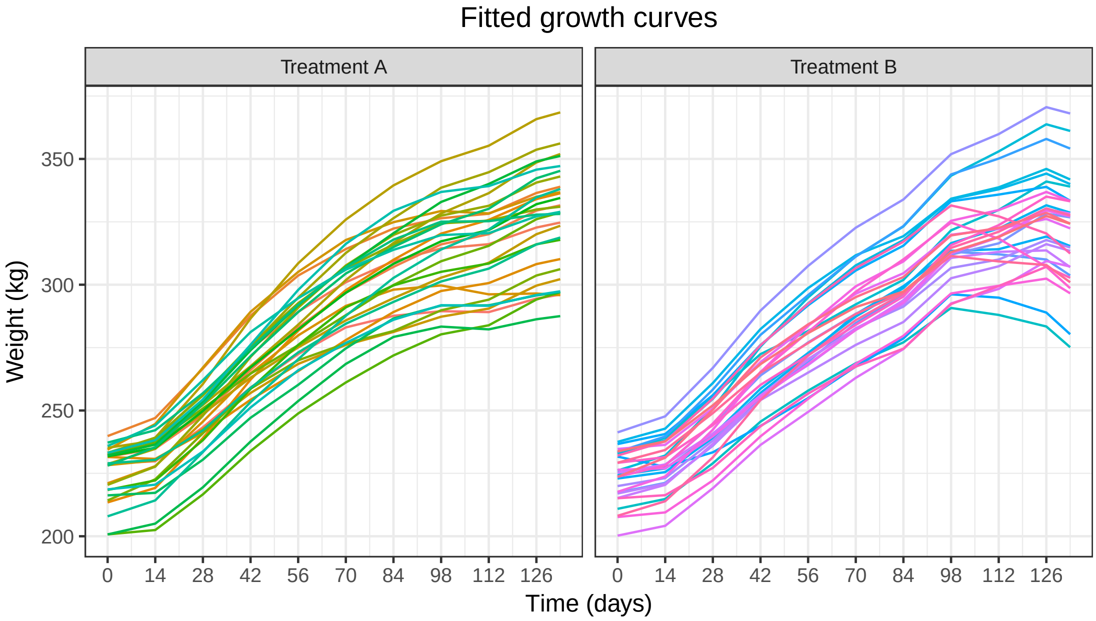


Figure 4.2: A plot of the I-prior fitted regression curves from Model 5. In this model, growth curves differ among cows and by treatment effect (with an interaction between cows and treatment effect), thus producing these 60 individual lines, one for each cow.

chopped meat, along with their water, fat and protein content. These data are recorded on a Tecator Infratec Food and Feed Analyzer working in the wavelength range 850 - 1050 nm by the Near Infrared Transmission (NIT) principle. Absorption data has not been measured continuously, but instead 100 distinct wavelengths were obtained. Figure 4.3 shows a sample of 10 such spectrometric curves.

For our analyses and many others' in the literature, the first 160 observations in the data set are used as a training sample for model fitting, and the remaining 55 observations as a test sample to evaluate the predictive performance of the fitted model. A summary of the various statistical methods applied to this data set, including various I-prior models, can be found in citebergsma2016. The focus here is to use the **iprior** package to fit various I-prior models to the Tecator data set.

Before we began, we preprocessed the spectral curves by taking their first differences . This leaves us with the 99-dimensional covariate, which is saved in the matrix object named `absorpTrain`. Our first modelling attempt is to estimate a linear effect by regressing the responses `fatTrain` against only a single high-dimensional covariate `absorpTrain` using the canonical RKHS. The model is loaded as an `ipriorKernel` object as follows:

7. Necessary for functional covariates - approximation of the Sobolev-Hilbert space inner product

[1] Used with permission from Tecator (see `http://lib.stat.cmu.edu/datasets/tecator` for details). We used the version made available in the dataframe `tecator` from the R package **caret** for our analyses.

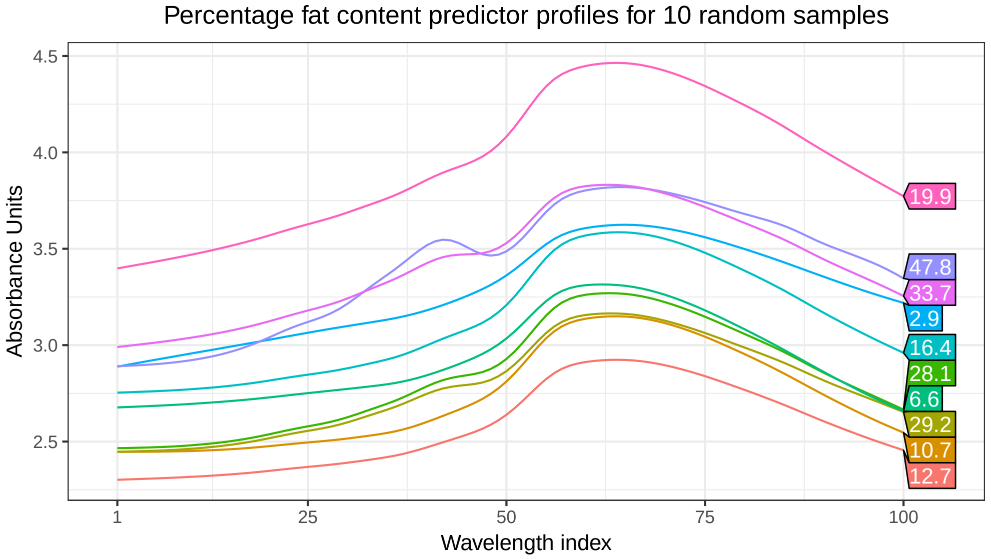


Figure 4.3: Sample of spectrometric curves used to predict fat content of meat. For each meat sample the data consists of a 100 channel spectrum of absorbances and the contents of moisture, fat (numbers shown in boxes) and protein measured in percent. The absorbance is $-\log 10$ of the transmittance measured by the spectrometer. The three contents, measured in percent, are determined by analytic chemistry.

```
R> # Model 1: canonical RKHS (linear)
R> (mod1 <- kernL(y = fatTrain, absorpTrain))
```

```
##
## Sample size =  160
## Number of x variables, p =  1
## Number of scale parameters, l =  1
## Number of interactions =  0
##
## Info on H matrix:
##
## List of 1
##  $ absorpTrain: Canonical [1:160, 1:160] 0.000254 0.0003 -0.000231 -..
```

Here, we have used the non-formula syntax because each object after the `y` argument is treated as a single covariate, even if it is multi-dimensional, i.e., a matrix. We could have also used the formula syntax and used the `model` option `one.lam = TRUE`. Note that the canonical RKHS is used by default.

Our second and third model uses a polynomial-type construction of the canonical RKHS, which allows us to add quadratic and cubic terms of the spectral curves. The syntax is as before with the addition of `absorpTrain^b`, which element-wise raises the entries of the matrix `absorpTrain` to the power `b`. To date, the only method to fit these models parsimoniously in **iprior** is by using non-formula syntax with `model` option `order` to control the scale parameters of the RKHS. Both models only have a single parameter. Without specifying the `order` option, additional scale parameters would be fitted, one for each quadratic and cubic term.

```
R> # Model 2: canonical RKHS (quadratic)
R> mod2 <- kernL(y = fatTrain, absorpTrain, absorpTrain ^ 2,
+                  model = list(order = c("1", "1^2")))
```

```
R> # Model 3: canonical RKHS (cubic)
R> mod3 <- kernL(y = fatTrain, absorpTrain, absorpTrain ^ 2, absorpTrain ^ 3,
+                  model = list(order = c("1", "1^2", "1^3")))
```

Next, we fitted a smooth dependence of fat content on the spectrometric curves using the FBM RKHS. By default, the Hurst coefficient for the FBM RKHS is set to be 0.5. However, we can use the function `fbmOptim()` which is able to compute the maximum likelihood estimate for the Hurst coefficient.

```
R> # Model 4: FBM RKHS (default Hurst = 0.5)
R> mod4 <- kernL(y = fatTrain, absorpTrain, model = list(kernel = "FBM"))
```

```
R> mod4

##
## Sample size =  160
## Number of x variables, p =  1
## Number of scale parameters, l =  1
## Number of interactions =  0
##
## Info on H matrix:
##
## List of 1
##  $ absorpTrain: FBM,0.5 [1:160, 1:160] 0.016192 -0.000775 -0.00346 -..
```

Finally, we add an extra covariate (meat moisture content) which is assumed to have a linear effect on fat content. Doing so adds one extra parameter to the model. To specify multiple kernels, we need to include the `model` option `kernel = c("FBM", "Canonical")` to indicate the effect of the respective covariates on the response. This is verified by inspecting the `print` output of the `ipriorKernel` object, and indeed we see that there are now two scale parameters, and the kernel loader correctly assigns the FBM and canonical RKHS to the spectrometric curves and moisture content respectively.

```
R> # Model 5: FBM RKHS + extra covariate
R> (mod5 <- kernL(y = fatTrain, absorpTrain, waterTrain,
+                  model = list(kernel = c("FBM", "Canonical"))))
```

```
##
## Sample size =  160
## Number of x variables, p =  2
## Number of scale parameters, l =  2
## Number of interactions =  0
##
## Info on H matrix:
##
## List of 2
##  $ absorpTrain: FBM,0.5 [1:160, 1:160] 0.016192 -0.000775 -0.00346 -..
##  $ waterTrain : Canonical [1:160, 1:160] 10.9 58.9 -23.8 -29.7 18.2 ..
```

All of the above models were fitted using `ipriorOptim`, except for the last two model, where we used `fbmOptim` in order to obtain the maximum likelihood estimate for the Hurst coefficient of the FBM RKHS. Predicted values of the test data set can be obtained using the `predict` function

```
R> fatTestPredicted <- predict(mod1.fit, list(absorpTest))
R> head(fatTestPredicted)
```

```
## [1] 14.12268 15.85864 15.84706 21.59324 25.22315 26.57978
```

and the root mean squared error (RMSE) calculated for each of the models. It was noted that for some models, different EM starting values gave slightly different results, and we suspect this is a due to numerical issues with the computation of the variance of marginal I-prior distribution. Nonetheless, the predicted values, and hence the RMSE, remain fairly robust.

The results are summarised in Table 4.3. Models 1-3 have the same number of parameters, so a direct comparison can be done, with the model giving the highest likelihood value preferred. In this case, it is the model with a quadratic effect, giving a test RMSE of 1.23. Models with the FBM RKHS gave better prediction still. A smooth effect (Hurst = 0.5) yields a test RMSE of 0.67, and this is improved only slightly by using the maximum likelihood estimate for the Hurst coefficient of 0.519. The best predictive model obtained was the final model, i.e., a smooth effect (Hurst = 0.934) with an additional covariate, giving a test RMSE of 0.68.

		Log-	RMSE	
Model	I-prior effect	likelihood	Train	Test
1	Linear	−409.32	2.85	3.24
2	Quadratic	−279.64	0.72	1.23
3	Cubic	−301.26	0.99	1.65
4	Smooth (Hurst = 0.5)	−148.34	0.00	0.67
4a	Smooth (Hurst = 0.519)	−146.23	0.00	0.66
5	Smooth (Hurst = 0.934) with additional covariate	−213.51	0.31	0.54

Table 4.3: A summary of the I-prior models fitted on the Tecator data set.

Chapter 5

I-priors for categorical responses

5.1 Preliminary

Observe data $\{(y_1,x_1),\dots,(y_n,x_n)\}$ where each $x_i \in \mathcal{X}$. Let $y_i \in \{1,\dots,m\}$ and write $y_i = (y_{i1},\dots,y_{im})$ where $y_{ij} = 1$ if $y_i = j$ and 0 otherwise. For $j = 1,\dots,m$, attempt to model

$$y_{ij} = \alpha_j + f_j(x_i) + \epsilon_{ij}$$
$$(\epsilon_{i1},\dots,\epsilon_{im})^\top \overset{\text{iid}}{\sim} \mathrm{N}_m(\mathbf{0}, \boldsymbol{\Psi}^{-1})$$

Define $f_j(x) = f(x,j)$ such that each f_j belong to some RKHS $\mathcal{F}$ (and not separate RKHSs $\mathcal{F}_j$). The reproducing kernel of $\mathcal{F}$ is $h : (\mathcal{X} \times \{1,\dots,m\})^2 \to \mathbb{R}$ as defined by

$$h\big((x,j),(x',j')\big) = a(j,j')h_\eta(x,x').$$

Choices for $a : \{1,\dots,m\} \times \{1,\dots,m\} \to \mathbb{R}$ include

1. The Pearson kernel
$$a(j,j') = \frac{\delta_{jj'}}{\mathrm{P}(X=j)} - 1$$

2. The Identity kernel
$$a(j,j') = \delta_{jj'}$$

The kernel h_η may be any of the usual kernels, i.e. canonical, fBm, Pearson, SE, polynomial, etc. and this kernel depends on the hyperparameters η. Denote $\mathbf{H}$ as the $n \times n$ matrix with (r,s) entries equal to $h_\eta(x_r,x_s)$ for $r,s \in \{1,\dots,\}$. Similarly, denote $\mathbf{A}$ as the $m \times m$ matrix with (k,j) entries equal to $a(k,j)$. Note that for the identity kernel, $\mathbf{A} = \mathbf{I}_m$.

The regression model in vector form:

$$\overbrace{\begin{pmatrix} y_{i1} \\ \vdots \\ y_{im} \end{pmatrix}}^{\mathbf{y}_i} = \overbrace{\begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{pmatrix}}^{\boldsymbol\alpha} + \overbrace{\begin{pmatrix} f_1(x_i) \\ \vdots \\ f_m(x_i) \end{pmatrix}}^{\mathbf{f}(x_i)} + \overbrace{\begin{pmatrix} \epsilon_{i1} \\ \vdots \\ \epsilon_{im} \end{pmatrix}}^{\boldsymbol\epsilon_i}$$
$$\boldsymbol\epsilon_i \overset{\text{iid}}{\sim} \mathrm{N}_m(\mathbf{0}, \boldsymbol\Psi^{-1}).$$

An I-prior on the regression function $f : \mathcal{X} \times \{1, \dots, m\} \to \mathbb{R}$ takes the form

$$f_j(x) = f(x, j) = \sum_{k=1}^m \sum_{i=1}^n a(j,k) h_\eta(x, x_i) w_{ij}$$

where $(w_{i1}, \dots, w_{im})^\top \overset{\text{iid}}{\sim} \mathrm{N}_m(\mathbf{0}, \boldsymbol\Psi)$.

Rearrange the n observations per class. Let $\mathbf{f}_j = \big(f_j(x_1), \dots, f_j(x_n)\big)^\top \in \mathbb{R}^n$. We can write the I-prior as $\mathbf{f}_j = \mathbf{A}_{jj} \cdot \mathbf{H}\mathbf{w}_j$ Therefore, $\mathbf{f}_j \sim \mathrm{N}_n(\mathbf{0}, \boldsymbol\Psi_{jj}\mathbf{A}_{jj}\mathbf{H}^2)$, and

$$\begin{aligned} \operatorname{Cov}(\mathbf{f}_j, \mathbf{f}_k) &= \operatorname{Cov}(\mathbf{A}_{jj} \cdot \mathbf{H}\mathbf{w}_j, \mathbf{A}_{kk} \cdot \mathbf{H}\mathbf{w}_k) \\ &= \mathbf{A}_{jj}\mathbf{A}_{kk} \cdot \mathbf{H} \operatorname{Cov}(\mathbf{w}_j, \mathbf{w}_k)\mathbf{H} \\ &= \mathbf{A}_{jj}\mathbf{A}_{kk}\boldsymbol\Psi_{jk}\mathbf{H}^2. \end{aligned}$$

Write $\mathbf{f} = (\mathbf{f}_1^\top, \dots, \mathbf{f}_m^\top)^\top\mathbb{R}^{nm}$. Then this is multivariate normal with mean $\mathbf{0}$ and covariance matrix equal to

$$\begin{aligned} \operatorname{Var}\mathbf{f} &= (\mathbf{A} \otimes \mathbf{H})(\boldsymbol\Psi \otimes \mathbf{I}_n)(\mathbf{A} \otimes \mathbf{H}) \\ &= (\mathbf{A}\boldsymbol\Psi\mathbf{A} \otimes \mathbf{H}^2) \\ &= \big(\boldsymbol\Omega_{jk}\mathbf{H}^2\big)_{j,k=1}^n \end{aligned}$$

where $\boldsymbol\Omega_{jk} = (\mathbf{A}\boldsymbol\Psi\mathbf{A})_{jk}$. Out of interest, this can be expressed as a matrix normal distribution. Write $\mathbf{w}$ as the $n \times m$ matrix with entries equal to w_{ij}. Then $\mathbf{w} \sim \mathrm{MN}_{n,m}(\mathbf{0}, \mathbf{I}_n, \boldsymbol\Psi)$ and $\mathbf{f} = \mathbf{H}\mathbf{w}\mathbf{A} \sim \mathrm{MN}_{n,m}(\mathbf{0}, \mathbf{H}^2, \mathbf{A}\boldsymbol\Psi\mathbf{A})$.

5.1.1 Special case: $\boldsymbol\Psi = \mathbf{I}_m$ with identity kernel

In this case, the matrix normal distribution for $\mathbf{f}$ is $\mathrm{MN}_{n,m}(\mathbf{0}, \mathbf{H}^2, \mathbf{I}_m)$. That is to say, the *columns* of $\mathbf{f}$, i.e. $\mathbf{f}_j$, are iid observations $\mathbf{f}_j \overset{\text{iid}}{\sim} \mathrm{N}_n(\mathbf{0}, \mathbf{H}^2)$.

Chapter 6

Estimation of I-probit models using variational inference

In this chapter we provide the details of the variational algorithm to estimate categorical I-prior models.

6.1 Relevant distributions

$$p(\mathbf{y}|\mathbf{y}^*) = \prod_{i=1}^{n}\prod_{j=1}^{m} p_{ij} = \prod_{i=1}^{n}\prod_{j=1}^{m} \mathbb{1}[y^*_{ij} = \max_k y^*_{ik}]^{\mathbb{1}[y_i=j]}$$

$$\begin{aligned}
p(\mathbf{y}^*|\mathbf{f}) &= \prod_{i=1}^{n}\prod_{j=1}^{m} \mathrm{N}(f_{ij}, 1) \\
&= \exp\left[-\frac{nm}{2}\log 2\pi - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}(y^*_{ij} - f_{ij})^2\right] \\
&= \exp\left[-\frac{nm}{2}\log 2\pi - \frac{1}{2}\|\mathbf{y}^* - \mathbf{f}\|^2\right]
\end{aligned}$$

$$f_{ij} = \alpha_j + \sum_{k=1}^{n} h_{\lambda_j}(x_i, x_k) w_{kj}$$

$$\begin{aligned}
p(\mathbf{w}) &= \prod_{i=1}^{n}\prod_{j=1}^{m} p(w_{ij}) \\
&= \prod_{i=1}^{n}\prod_{j=1}^{m} \mathrm{N}(0, 1) \\
&= \exp\left[-\frac{nm}{2}\log 2\pi - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}^2\right] \\
&= \exp\left[-\frac{nm}{2}\log 2\pi - \frac{1}{2}\mathbf{w}^\top\mathbf{w}\right]
\end{aligned}$$

$$p(\lambda, \alpha) \propto \text{const.}$$

6.2 Mean field distributions

$$
\begin{aligned}
p(\mathbf{y}^*, \mathbf{w}, \alpha, \lambda|\mathbf{y}) &\equiv q(\mathbf{y}^*)q(\mathbf{w})q(\lambda)q(\alpha) \\
&\equiv \prod_{i,j} q(y^*_{ij})q(\mathbf{w})q(\lambda)q(\alpha)
\end{aligned}
$$

The first line is by assumption, while the second line follows from an induced factorisation, as we will see later. Denote by $\tilde{q}$ the distributions which minimise the KL divergence (maximises the lower bound). Then, for each of $\xi \in \{\mathbf{y}^*, \mathbf{w}, \alpha, \lambda\}$, $\tilde{q}$ satisfies

$$\log \tilde{q}(\xi) = \mathrm{E}_{-\xi}[\log p(\mathbf{y}, \mathbf{y}^*, \mathbf{w}, \alpha, \lambda)] + \text{const.}$$

$\tilde{q}(\mathbf{y}^*)$

In this subsection, we use the notation $y^*_i = (y^*_{i1}, \dots, y^*_{im})$ to denote the vector of length m containing the latent variables for response i. The joint distribution for $\mathbf{y}^* = (y^*_1, \dots, y^*_n)^\top$ is a product of the distribution for each of the components y^*_i - this is a consequence of the independence structure across observations. Therefore, we can consider the variational density for each y^*_i separately.

Consider the case where y_i takes one particular value $j \in \{1, \dots, m\}$. The mean-field density $q(y^*_i)$ for each $i = 1, \dots, n$ is found to be

$$
\begin{aligned}
\log \tilde{q}(y^*_i) &= \mathbb{1}[y^*_{ij} = \max_k y^*_{ik}] \cdot \mathrm{E}_{\mathbf{w},\alpha,\lambda}\left[-\frac{1}{2}\sum_{k=1}^m (y^*_{ik} - f_{ik})^2\right] + \text{const.} \\
&= \mathbb{1}[y^*_{ij} = \max_k y^*_{ik}] \cdot \left[-\frac{1}{2}\sum_{k=1}^m (y^*_{ik} - \tilde{f}_{ik})^2\right] + \text{const.} \\
&\equiv \begin{cases} \prod_{k=1}^m \mathrm{N}(\tilde{f}_{ik}, 1) & \text{if } y^*_{ij} > y^*_{ik}, \forall k \neq j \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

where $\tilde{f}_{ik} = \mathrm{E}[\alpha_k] + \sum_{l=1}^m h_{\mathrm{E}[\lambda_k]}(x_i, x_l)\,\mathrm{E}[w_{il}]$, and expectations are taken under the optimal mean-field distribution $\tilde{q}$. The distribution for $q(y^*_i)$ is a truncated m-variate normal distribution such that the jth component is always largest. It is worth investigating the properties of this distribution, and we now present some relevant definitions and results.

Definition 6.1 (Conically-truncated multivariate normal distribution)**.** Let $\mathbf{X} = (X_1, \dots, X_d)$ be a d-dimensional random variable with pdf defined as

$$p(\mathbf{x}) = \begin{cases} \prod_{i=1}^d \mathrm{N}(\mu_i, \sigma_i) & \text{if } X_j > X_i, \forall i \neq j \\ 0 & \text{otherwise} \end{cases}$$

for some $j \in \{1, \ldots, d\}$. We denote the distribution of $\mathbf{X}$ by $\mathrm{N}^{(j)}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)$ and $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2)$. The pdf of $\mathbf{X}$ has support on the set $\{\mathbb{R}^d \,|\, x_j > x_i, \forall i \neq j\}$ and the following functional form:

$$p(\mathbf{x}) = \frac{C^{-1}}{\sigma_1 \cdots \sigma_d (2\pi)^{d/2}} \exp\left[-\frac{1}{2}\sum_{i=1}^{d}\left(\frac{x_i - \mu_i}{\sigma_i}\right)^2\right]$$

where ϕ is the pdf of a standard normal distribution and

$$C = \mathrm{E}_Z\left[\prod_{\substack{i=1\\ i\neq j}}^{d} \Phi\left(\frac{\sigma_j}{\sigma_i} Z + \frac{\mu_j - \mu_i}{\sigma_i}\right)\right]$$

where $Z \sim \mathrm{N}(0,1)$. In the case where all variances are unity, the pdf of $\mathbf{X} \sim \mathrm{N}^{(j)}(\boldsymbol{\mu}, \mathbf{I}_d)$ is

$$p(\mathbf{x}) = \left\{(2\pi)^{d/2}\, \mathrm{E}_Z\left[\prod_{\substack{i=1\\ i\neq j}}^{d} \Phi\left(Z + \mu_j - \mu_i\right)\right]\right\}^{-1} \exp\left[-\frac{1}{2}\sum_{i=1}^{d}(x_i - \mu_i)^2\right].$$

Proof. A derivation of the functional form for the pdf of $X \sim \mathrm{N}^{(j)}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is given. Using the fact that $\int p(x)\,\mathrm{d}x = 1$, and that

$$
\begin{aligned}
&\int \mathbb{1}[x_i < x_j, \forall i \neq j] \prod_{i=1}^{d} \mathrm{N}(\mu_i, \sigma_i^2)\,\mathrm{d}x_1 \cdots \mathrm{d}x_d \\
&\quad = \int \mathbb{1}[x_i < x_j, \forall i \neq j] \prod_{i=1}^{d} \left[\frac{1}{\sigma_i} \phi\left(\frac{x_i - \mu_i}{\sigma} \right) \right] \mathrm{d}x_1 \cdots \mathrm{d}x_d \\
&\quad = \int \mathbb{1}[x_i < x_j, \forall i \neq j] \frac{1}{\sigma_j} \phi\left(\frac{x_j - \mu_j}{\sigma_j} \right) \prod_{\substack{i=1\\ i\neq j}}^{d} \left[\frac{1}{\sigma_i} \phi\left(\frac{x_i - \mu_i}{\sigma_i} \right) \right] \mathrm{d}x_1 \cdots \mathrm{d}x_d \\
&\quad = \int \prod_{\substack{i=1\\ i\neq j}}^{d} \Phi\left(\frac{x_j - \mu_i}{\sigma_i} \right) \frac{1}{\sigma_j} \phi\left(\frac{x_j - \mu_j}{\sigma_j} \right) \mathrm{d}x_j \\
&\quad = \int \prod_{\substack{i=1\\ i\neq j}}^{d} \Phi\left(\frac{\sigma_j z_j + \mu_j - \mu_i}{\sigma_i} \right) \phi(z_j)\,\mathrm{d}z_j \\
&\qquad \text{(by using the standardisation } z_j = (x_j - \mu_j)/\sigma_j) \\
&\quad = \mathrm{E}\left[\prod_{\substack{i=1\\ i\neq j}}^{d} \Phi\left(\frac{\sigma_j}{\sigma_i} Z_j + \frac{\mu_j - \mu_i}{\sigma_i} \right) \right]
\end{aligned}
$$

the proof follows directly. □

Lemma 6.1. *Let $X \sim \mathrm{N}^{(j)}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with pdf $p(\mathbf{x})$ as defined in Definition 6.1. Then*

(i) The expectation $\mathrm{E}[\mathbf{X}] = \big(\mathrm{E}[X_1], \dots, \mathrm{E}[X_d]\big)$ is given by

$$
\mathrm{E}[X_i] = \begin{cases} \mu_i - \sigma_i C^{-1} \, \mathrm{E}_Z \left[\phi_i \prod_{k \neq i,j} \Phi_k \right] & \text{if } i \neq j \\ \mu_j - \sigma_j \sum_{i \neq j} \big(\mathrm{E}[X_i] - \mu_i \big) & \text{if } i = j \end{cases}
$$

(ii) The differential entropy $\mathcal{H}(p)$ is given by

$$
\mathcal{H}(p) = \log C + \frac{d}{2} \log 2\pi + \frac{1}{2} \sum_{i=1}^{d} \log \sigma_i^2 + \frac{1}{2} \sum_{i=1}^{d} \frac{1}{\sigma_i^2} \, \mathrm{E}[x_i - \mu_i]^2
$$

where $C = \mathrm{E}\left[\prod_{i \neq j} \Phi_i\right]$*, and we had defined*

$$\phi_i = \phi_i(Z) = \phi\left(\frac{\sigma_j Z + \mu_j - \mu_i}{\sigma_i}\right)$$
$$\Phi_i = \Phi_i(Z) = \Phi\left(\frac{\sigma_j Z + \mu_j - \mu_i}{\sigma_i}\right)$$

with $Z \sim \mathrm{N}(0,1)$*, and* $\phi(\cdot)$ *and* $\Phi(\cdot)$ *the pdf and cdf of* Z *respectively.*

As we know, y_i takes on any one value from the set $\{1, \dots, m\}$. Thus, we have that the distribution of $(y^*_{i1}, \dots, y^*_{im})$ is $\mathrm{N}^{(y_i)}(\boldsymbol{\mu}_i, \mathbf{I}_m)$, where $\boldsymbol{\mu}_i = (\tilde{f}_{i1}, \dots, \tilde{f}_{im})$. The expectation is given by

$$\mathrm{E}[y^*_{ik}] = \begin{cases} \tilde{f}_{ik} - C_i^{-1} \mathrm{E}_Z\left[\phi_{ik}(Z) \prod_{l \neq k, y_i} \Phi_{il}(Z)\right] & \text{if } k \neq y_i \\ \tilde{f}_{iy_i} - \sum_{k \neq y_i}\left(\mathrm{E}[y^*_{ik}] - \tilde{f}_{ik}\right) & \text{if } k = y_i \end{cases}$$

where

$$\phi_{ik}(Z) = \phi(Z + \tilde{f}_{iy_i} - \tilde{f}_{ik})$$
$$\Phi_{ik}(Z) = \Phi(Z + \tilde{f}_{iy_i} - \tilde{f}_{ik})$$
$$C_i = \mathrm{E}_Z\left[\prod_{\substack{i=1 \\ i \neq j}}^{d} \Phi\left(Z + \tilde{f}_{iy_i} - \tilde{f}_{ik}\right)\right]$$

and $Z \sim \mathrm{N}(0,1)$ with PDF and CDF $\phi(\cdot)$ and $\Phi(\cdot)$ respectively. In order to calculate these expectations, we need to compute the following integrals:

$$\mathrm{E}_Z\left[\phi_{ik}(Z) \prod_{l \neq k, j} \Phi_{il}(Z)\right] = \int \phi_{ik}(z) \prod_{l \neq k, j} \Phi_{il}(z)\phi(z)\,\mathrm{d}z, \quad \forall k \neq y_i$$
$$C_i = \mathrm{E}_Z\left[\prod_{l \neq j} \Phi_{il}(Z)\right] = \int \prod_{l \neq j} \Phi_{il}(z)\phi(z)\,\mathrm{d}z$$

Since these are functions of a Gaussian pdf, these can be computed rather efficiently using quadrature methods.

$\tilde{q}(\mathbf{w})$

For each $j = 1, \dots, m$, denote $\mathbf{y}_j^* = (y_{1j}^*, \dots, y_{nj}^*)^\top$ as the vector of length n containing all latent observations for each class. Then,

$$\begin{aligned}
\log \tilde{q}(\mathbf{w}) &= \mathrm{E}_{\mathbf{y}^*,\alpha,\lambda} \left[-\frac{1}{2} \sum_{j=1}^m \| \mathbf{y}_j^* - \alpha_j \mathbf{1}_n - \mathbf{H}_{\lambda_j} \mathbf{w}_j \|^2 - \frac{1}{2} \sum_{j=1}^m \| \mathbf{w}_j \|^2 \right] + \text{const.} \\
&= -\frac{1}{2} \sum_{j=1}^m \mathrm{E}_{\mathbf{y}^*,\alpha,\lambda} \left[\mathbf{w}_j^\top \mathbf{H}_{\lambda_j}^2 \mathbf{w}_j + \mathbf{w}_j^\top \mathbf{w}_j - 2(\mathbf{y}_j^* - \alpha_j \mathbf{1}_n)^\top \mathbf{H}_{\lambda_j} \mathbf{w}_j \right] + \text{const.} \\
&= -\frac{1}{2} \sum_{j=1}^m \Big(\mathbf{w}_j^\top (\mathrm{E}[\mathbf{H}_{\lambda_j}^2] + \mathbf{I}_n) \mathbf{w}_j - 2(\mathrm{E}[\mathbf{y}_j^*] - \mathrm{E}[\alpha_j] \mathbf{1}_n)^\top \mathrm{E}[\mathbf{H}_{\lambda_j}] \mathbf{w}_j \Big) + \text{const.}
\end{aligned}$$

Let $\mathbf{A}_j = \mathrm{E}[\mathbf{H}_{\lambda_j}^2] + \mathbf{I}_n$ and $\mathbf{a}_j = \mathrm{E}[\mathbf{H}_{\lambda_j}](\mathrm{E}[\mathbf{y}_j^*] - \mathrm{E}[\alpha_j]\mathbf{1}_n)$. Then, using the fact that

$$\mathbf{w}_j^\top \mathbf{A}_j \mathbf{w}_j - 2\mathbf{a}_j^\top \mathbf{w}_j = (\mathbf{w}_j - \mathbf{A}_j^{-1}\mathbf{a}_j)^\top \mathbf{A}_j (\mathbf{w}_j - \mathbf{A}_j^{-1}\mathbf{a}_j),$$

we see the $\log \tilde{q}(\mathbf{w})$ is a sum of quadratic terms in $\mathbf{w}_j$, and we recognise this as the kernel of the product of indepdendent multivariate normal densities. Therefore, for each $j = 1, \dots, m$,

$$\tilde{q}(\mathbf{w}_j) \equiv \mathrm{N}(\mathbf{A}_j^{-1}\mathbf{a}_j, \mathbf{A}_j^{-1}),$$

and $\tilde{q}(\mathbf{w}) = \prod_{j=1}^m \tilde{q}(\mathbf{w}_j)$. Because of this induced factorisation, we can obtain mean-field densities for each $\mathbf{w}_j$ separately. For convenience later in deriving the lower bound, we note that the second moment of $\tilde{q}(\mathbf{w}_j)$ is equal to $\mathrm{E}[\mathbf{w}_j\mathbf{w}_j^\top] = \mathbf{A}_j^{-1}(\mathbf{I}_n + \mathbf{a}_j\mathbf{a}_j^\top \mathbf{A}_j^{-1}) =: \widetilde{\mathbf{W}}_j$.

$\tilde{q}(\lambda)$

For $j = 1, \dots, m$,

$$\begin{aligned}
\log \tilde{q}(\lambda_j) &= \mathrm{E}_{\mathbf{y}^*,\mathbf{w},\alpha} \left[-\frac{1}{2} \sum_{j=1}^m \| \mathbf{y}_j^* - \alpha_j \mathbf{1}_n - \lambda_j \mathbf{H} \mathbf{w}_j \|^2 \right] + \text{const.} \\
&= -\frac{1}{2} \sum_{j=1}^m \mathrm{E}_{\mathbf{y}^*,\mathbf{w},\alpha} \left[\lambda_j^2 \mathbf{w}_j^\top \mathbf{H}^2 \mathbf{w}_j - 2\lambda_j (\mathbf{y}_j^* - \alpha_j \mathbf{1}_n)^\top \mathbf{H} \mathbf{w}_j \right] + \text{const.} \\
&= -\frac{1}{2} \sum_{j=1}^m \Big(\lambda_j^2 \operatorname{tr} \big(\mathbf{H}^2 \, \mathrm{E}[\mathbf{w}_j \mathbf{w}_j^\top] \big) - 2\lambda_j (\mathrm{E}[\mathbf{y}_j^*] - \mathrm{E}[\alpha_j] \mathbf{1}_n)^\top \mathbf{H} \, \mathrm{E}[\mathbf{w}_j] \Big) + \text{const.}
\end{aligned}$$

By completing the squares, we recognise this is as the kernel of the product of independent univariate normal densities. Thus, each $\lambda_j \sim \mathrm{N}(d_j/c_j, 1/c_j)$, where

$$c_j = \operatorname{tr}\left(\mathbf{H}^2 \operatorname{E}[\mathbf{w}_j\mathbf{w}_j^\top]\right) \quad \text{and} \quad d_j = (\operatorname{E}[\mathbf{y}_j^*] - \operatorname{E}[\alpha_j]\mathbf{1}_n)^\top \mathbf{H} \operatorname{E}[\mathbf{w}_j].$$

Supposing we use the same covariance kernel (and therefore scale parameter) for each regression class, the distribution for λ is easily seen as

$$\lambda \sim \mathrm{N}\left(\frac{\sum_{j=1}^m d_j}{\sum_{j=1}^m c_j}, \frac{1}{\sum_{j=1}^m c_j} \right).$$

$\tilde{q}(\alpha)$

For $j = 1, \dots, m$, denote $\mathbf{H}_i$ as the row vector of the kernel matrix $\mathbf{H}$. Then,

$$\begin{aligned}
\log \tilde{q}(\alpha) &= \operatorname{E}_{\mathbf{y}^*,\mathbf{w},\lambda}\left[-\frac{1}{2}\sum_{j=1}^m\sum_{i=1}^n \left(y_{ij}^* - \alpha_j - \lambda_j \textstyle\sum_{k=1}^n h(x_i,x_k)w_{kj}\right)^2 \right] + \text{const.} \\
&= -\frac{1}{2}\sum_{j=1}^m \operatorname{E}_{\mathbf{y}^*,\mathbf{w},\lambda}\left[n\alpha_j^2 - 2\alpha_j \sum_{i=1}^n (y_{ij}^* - \lambda_j \mathbf{H}_i \mathbf{w}_j) \right] + \text{const.} \\
&= -\frac{n}{2}\sum_{j=1}^m \left[\left(\alpha_j - \frac{1}{n}\sum_{i=1}^n (\operatorname{E}[y_{ij}^*] - \operatorname{E}[\lambda_j]\mathbf{H}_i\mathbf{w}_j) \right)^2 \right] + \text{const.}
\end{aligned}$$

which is of course the kernel of the product of m univariate normal densities, each with mean and variance

$$\tilde{\alpha}_j = \frac{1}{n}\sum_{i=1}^n \left(\operatorname{E}[y_{ij}^*] - \operatorname{E}[\lambda_j]\mathbf{H}_i \operatorname{E}[\mathbf{w}_j] \right) \quad \text{and} \quad v_{\alpha_j} = \frac{1}{n}.$$

Suppose that we use a single intercept parameter α. In this case, α is is also normally distributed with mean and variance

$$\tilde{\alpha} = \frac{1}{nm}\sum_{j=1}^m\sum_{i=1}^n \left(\operatorname{E}[y_{ij}^*] - \operatorname{E}[\lambda_j]\mathbf{H}_i \operatorname{E}[\mathbf{w}_j] \right) \quad \text{and} \quad v_\alpha = \frac{1}{nm}.$$

6.3 Monitoring the lower bound

A convergence criterion would be when there is no more significant increase in the lower bound $\mathcal{L}$, as defined by

$$\begin{aligned}
\mathcal{L} &= \int q(\mathbf{y}^*, \mathbf{w}, \lambda, \alpha) \log \left[\frac{p(\mathbf{y}, \mathbf{y}^*, \mathbf{w}, \lambda, \alpha)}{q(\mathbf{y}^*, \mathbf{w}, \lambda, \alpha)} \right] \mathrm{d}\mathbf{y}^* \, \mathrm{d}\mathbf{w} \, \mathrm{d}\lambda \, \mathrm{d}\alpha \\
&= \mathrm{E}[\log p(\mathbf{y}, \mathbf{y}^*, \mathbf{w}, \lambda, \alpha)] - \mathrm{E}[\log q(\mathbf{y}^*, \mathbf{w}, \lambda, \alpha)] \\
&= \mathrm{E}\left[\cancel{\log \prod_{i=1}^{n} \prod_{j=1}^{m} p(y_i | y_{ij}^*)} \right] + \mathrm{E}\left[\log p(\mathbf{y}^* | \mathbf{f})\right] + \mathrm{E}\left[\log p(\mathbf{w})\right] + \cancel{\mathrm{E}\left[\log p(\lambda)\right]} + \cancel{\mathrm{E}\left[\log p(\alpha)\right]} \\
&\quad - \mathrm{E}\left[\log q(\mathbf{y}^*)\right] - \mathrm{E}\left[\log q(\mathbf{w})\right] - \mathrm{E}\left[\log q(\lambda)\right] - \mathrm{E}\left[\log q(\alpha)\right]
\end{aligned}$$

Note that the categorical pmf $p(y_i | y_{ij}^*)$ becomes degenerate once the latent variables are known, so this term is cancelled out. With the exception of $q(\mathbf{y}^*)$, all of the distributions are Gaussian. The following results will be helpful.

Definition 6.2 (Differential entropy)**.** The differential entropy $\mathcal{H}$ of a pdf $p(x)$ is given by

$$\mathcal{H}(p) = -\int p(x) \log p(x) \, \mathrm{d}x = -\mathrm{E}_p[\log p(x)].$$

Lemma 6.2. *Let $p(x)$ be the pdf of a random variable x. Then if*

(i) p is a univariate normal distribution with mean μ and variance σ^2,

$$\mathcal{H}(p) = \frac{1}{2}(1 + \log 2\pi) + \frac{1}{2} \log \sigma^2$$

(ii) p is a d-dimensional normal distribution with mean μ and variance Σ,

$$\mathcal{H}(p) = \frac{d}{2}(1 + \log 2\pi) + \frac{1}{2} \log |\Sigma|$$

Terms involving distributions of $\mathbf{y}^*$

$$
\begin{aligned}
\mathrm{E}\left[\log p(\mathbf{y}^*|\mathbf{f})\right] - \mathrm{E}\left[\log q(\mathbf{y}^*)\right] &= \sum_{i=1}^{n}\sum_{j=1}^{m} \mathrm{E}\left[\log p(y^*_{ij}|f_{ij})\right] + \sum_{i=1}^{n} \mathcal{H}\big(q(y^*_i)\big) \\
&= \cancel{\sum_{i=1}^{n}\sum_{j=1}^{m}\left(-\frac{1}{2}\log 2\pi - \frac{1}{2}\,\mathrm{E}[y^*_{ij} - f_{ij}]^2\right)} \\
&\quad + \cancel{\sum_{i=1}^{n}\sum_{j=1}^{m}\left(\frac{1}{2}\log 2\pi + \frac{1}{2}\,\mathrm{E}[y^*_{ij} - f_{ij}]^2\right)} + \sum_{i=1}^{n}\log C_i
\end{aligned}
$$

Terms involving distributions of $\mathbf{w}$

$$
\begin{aligned}
\mathrm{E}\left[\log p(\mathbf{w})\right] - \mathrm{E}\left[\log q(\mathbf{w})\right] &= \sum_{j=1}^{m}\Big(\mathrm{E}\left[\log p(\mathbf{w}_j)\right] - \mathrm{E}\left[\log q(\mathbf{w}_j)\right]\Big) \\
&= \sum_{j=1}^{m}\left(-\frac{n}{2}\log 2\pi - \frac{1}{2}\,\mathrm{E}[\mathbf{w}_j^\top \mathbf{w}_j] + \mathcal{H}\big(q(\mathbf{w}_j)\big)\right) \\
&= \sum_{j=1}^{m}\left(\cancel{-\frac{n}{2}\log 2\pi} - \frac{1}{2}\,\mathrm{tr}\left(\mathrm{E}[\mathbf{w}_j\mathbf{w}_j^\top]\right) + \frac{n}{2}(1 + \cancel{\log 2\pi}) - \frac{1}{2}\log|\mathbf{A}_j|\right) \\
&= \frac{nm}{2} - \frac{1}{2}\sum_{j=1}^{m}\left(\mathrm{tr}\,\widetilde{\mathbf{W}}_j + \log|\mathbf{A}_j|\right)
\end{aligned}
$$

Terms involving distribution of $q(\lambda)$

$$
\begin{aligned}
-\,\mathrm{E}\left[\log q(\lambda)\right] &= \sum_{j=1}^{m}\mathcal{H}\big(q(\lambda_j)\big) \\
&= \sum_{j=1}^{m}\left(\frac{1}{2}(1 + \log 2\pi) - \frac{1}{2}\log c_j\right) \\
&= \frac{m}{2}(1 + \log 2\pi) - \frac{1}{2}\sum_{j=1}^{m}\log c_j
\end{aligned}
$$

or if using single λ

$$-\operatorname{E}[\log q(\lambda)] = \frac{1}{2}(1 + \log 2\pi) - \frac{1}{2}\log\sum_{j=1}^{m} c_j.$$

Terms involving distribution of $q(\alpha)$

$$\begin{aligned} -\operatorname{E}[\log q(\alpha)] &= \sum_{j=1}^{m} \mathcal{H}\big(q(\alpha_j)\big) \\ &= \frac{m}{2}(1 + \log 2\pi - \log n) \end{aligned}$$

or if using single α

$$-\operatorname{E}[\log q(\alpha)] = \frac{1}{2}(1 + \log 2\pi - \log nm).$$

The lower bound

$$\begin{aligned} \mathcal{L} &= \sum_{i=1}^{n} \log C_i + \frac{nm}{2} - \frac{1}{2}\sum_{j=1}^{m}\Big(\operatorname{tr}\widetilde{\mathbf{W}}_j + \log|\mathbf{A}_j|\Big) \\ &\quad + \frac{m}{2}(1 + \log 2\pi) - \frac{1}{2}\sum_{j=1}^{m}\log c_j + \frac{m}{2}(1 + \log 2\pi - \log n) \\ &= \frac{m}{2}\big(n + 2(1 + \log 2\pi) - \log n\big) - \frac{1}{2}\sum_{j=1}^{m}\Big(\operatorname{tr}\widetilde{\mathbf{W}}_j + \log|\mathbf{A}_j| + \log c_j\Big) + \sum_{i=1}^{n}\log C_i \end{aligned}$$

Of course, if using either single α or single λ, then the formula needs to be adjusted accordingly.

6.4 The variational algorithm

8. Is this variational EM... or CAVI?

Since there is a cyclic dependence of the parameters on each other, we employ a sequential update algorithm. In what follows, a tilde on the parameters indicate that these are the expectations of the parameters given the optimal factorised distributions $\tilde{q}$ derived earlier.

STEP 1: Update $\tilde{\mathbf{y}}^{*(t+1)}$ given $\tilde{\mathbf{w}}^{(t)}$, $\tilde{\lambda}^{(t)}$, and $\tilde{\alpha}^{(t)}$

STEP 2: Update $\tilde{\mathbf{w}}^{(t+1)}$ given $\tilde{\mathbf{y}}^{*(t+1)}$, $\tilde{\lambda}^{(t)}$, and $\tilde{\alpha}^{(t)}$

STEP 3: Update $\tilde{\lambda}^{(t+1)}$ given $\tilde{\mathbf{y}}^{*(t+1)}$, $\tilde{\mathbf{w}}^{(t+1)}$, and $\tilde{\alpha}^{(t)}$

STEP 4: Update $\tilde{\alpha}^{(t+1)}$ given $\tilde{\mathbf{y}}^{*(t+1)}$, $\tilde{\mathbf{w}}^{(t+1)}$, and $\tilde{\lambda}^{(t+1)}$

Algorithm 2 VB-EM algorithm for the probit I-prior model

```
procedure Initialise
    for $j = 1, \dots, m$ do
        $\tilde{\mathbf{w}}_j^{(0)} \leftarrow \mathbf{0}_n$
        $\tilde{\alpha}_j^{(0)} \leftarrow \text{N}(0,1)$
        $\tilde{\lambda}_j^{(0)} \leftarrow \text{N}(0,1)$
        $\tilde{\lambda}_j^{sq(0)} \leftarrow (\tilde{\lambda}_j^{(0)})^2$        ▷ this is $\text{E}[\lambda_j^2]$
        $\mathbf{H}_{\lambda_j}^{(0)} \leftarrow \tilde{\lambda}_j^{(0)}\mathbf{H}$
        $\mathbf{H}_{\lambda_j}^{sq(0)} \leftarrow \tilde{\lambda}_j^{sq(0)}\mathbf{H}^2$
    end for
end procedure

procedure Update for $\tilde{\mathbf{f}}$ (time $t$)
    for $j = 1, \dots, m$ do
        $\tilde{\mathbf{f}}_j^{(t+1)} \leftarrow \tilde{\alpha}_j^{(t)}\mathbf{1}_n + \mathbf{H}_{\lambda_j}\tilde{\mathbf{w}}_j^{(t)}$
    end for
    $\tilde{\mathbf{f}}^{(t+1)} \leftarrow \big(\tilde{\mathbf{f}}_1^{(t+1)}, \dots, \tilde{\mathbf{f}}_m^{(t+1)}\big)^\top$
end procedure

procedure Update for $y_{ij}^*$ (time $t$)
    for $i = 1, \dots, n$ do
        $j \leftarrow y_i$
        $C_i^{(t+1)} \leftarrow \prod_{k \neq j} \Phi\left((\tilde{f}_{ij}^{(t+1)} - \tilde{f}_{ik}^{(t+1)})/\sqrt{2}\right)$
        for $k = 1, \dots, j-1, j+1, \dots, m$ do
            $D_{ik} \leftarrow \text{E}_Z\left[\phi_k(Z + \tilde{f}_{ij}^{(t+1)} - \tilde{f}_{ik}^{(t+1)}) \prod_{l \neq k,j} \Phi_l(Z + \tilde{f}_{ij}^{(t+1)} - \tilde{f}_{ik}^{(t+1)})\right]$
            $\tilde{y}_{ik}^{*(t+1)} \leftarrow \tilde{f}_{ik}^{(t+1)} - D_{ik}/C_i^{(t+1)}$
        end for
        $\tilde{y}_{ij}^{*(t+1)} \leftarrow \tilde{f}_{ij}^{(t+1)} - \sum_{k \neq j}\big(\tilde{y}_{ik}^{*(t+1)} - \tilde{f}_{ik}^{(t+1)}\big)$
    end for
end procedure
```

```
procedure UPDATE FOR $\mathbf{w}_j$ (time $t$)
    for $j = 1, \ldots, m$ do
        $\tilde{\mathbf{y}}_j^{*(t+1)} \leftarrow (\tilde{y}_{1j}^{(t+1)}, \ldots, \tilde{y}_{nj}^{(t+1)})^\top$
        $\mathbf{A}_j \leftarrow \mathbf{H}_{\lambda_j}^{sq(t)} + \mathbf{I}_n$
        $\mathbf{a}_j \leftarrow \mathbf{H}_\lambda(\tilde{\mathbf{y}}_j^{*(t+1)} - \tilde{\alpha}_j^{(t)}\mathbf{1}_n)$
        $\tilde{\mathbf{w}}_j^{(t+1)} \leftarrow \mathbf{A}_j^{-1}\mathbf{a}_j$
        $\widetilde{\mathbf{W}}_j^{(t+1)} \leftarrow \mathbf{A}_j^{-1}(\mathbf{I}_n + \mathbf{a}_j\mathbf{a}_j^\top\mathbf{A}_j^{-1})$
        $\text{logdetA}_j^{(t+1)} \leftarrow \log|\mathbf{A}_j|$
    end for
end procedure

procedure UPDATE FOR $\lambda$ (time $t$)
    for $j = 1, \ldots, m$ do
        $c_j^{(t+1)} \leftarrow \mathrm{tr}\left(\mathbf{H}^2\widetilde{\mathbf{W}}_j\right)$
        $d_j \leftarrow (\tilde{\mathbf{y}}_j^{*(t+1)} - \tilde{\alpha}_j^{(t)}\mathbf{1}_n)^\top\mathbf{H}\tilde{\mathbf{w}}_j^{(t+1)}$
        $\tilde{\lambda}_j^{(t+1)} \leftarrow d_j/c_j^{(t+1)}$
        $\tilde{\lambda}_j^{sq(t+1)} \leftarrow 1/c_j^{(t)} + (d_i/c_i^{(t+1)})^2$
    end for
    if single $\lambda$ then $\forall j$
        $\tilde{\lambda}_j^{(t+1)} \leftarrow \sum_j d_j \Big/ \sum_j c_j^{(t+1)}$
        $\tilde{\lambda}_j^{sq(t+1)} \leftarrow 1 \Big/ \sum_j c_j^{(t+1)} + \left(\sum_j d_j \Big/ \sum_j c_j^{(t+1)}\right)^2$
    end if
    call UPDATE KERNEL MATRICES
end procedure

procedure UPDATE KERNEL MATRICES (time $t$)
    for $j = 1, \ldots, m$ do
        $\mathbf{H}_{\lambda_j}^{(t+1)} \leftarrow \tilde{\lambda}_j^{(t+1)}\mathbf{H}$
        $\mathbf{H}_{\lambda_j}^{sq(t+1)} \leftarrow \tilde{\lambda}_j^{sq(t+1)}\mathbf{H}^2$
    end for
end procedure
```

```
procedure UPDATE FOR $\alpha$ (time $t$)
    if single $\alpha$ then
        $\tilde{\alpha}^{(t+1)} \leftarrow \frac{1}{nm} \sum_{j=1}^{m} \sum_{i=1}^{n} \big(\tilde{y}_{ij}^{*(t+1)} - \tilde{\lambda}_j^{(t+1)} \mathbf{H}_i \tilde{\mathbf{w}}_j^{(t+1)}\big)$
    else
        for $j = 1, \ldots, m$ do
            $\tilde{\alpha}_j^{(t+1)} \leftarrow \frac{1}{n} \sum_{i=1}^{n} \big(\tilde{y}_{ij}^{*(t+1)} - \tilde{\lambda}_j^{(t+1)} \mathbf{H}_i \tilde{\mathbf{w}}_j^{(t+1)}\big)$
        end for
    end if
end procedure

procedure CALCULATE LOWER BOUND (time $t$)
    $\mathcal{L}^{(t)} \leftarrow \frac{1}{2}\big(nm - \log nm + 3(1 + \log 2\pi)\big) - \frac{1}{2}\left(\text{logdetA}^{(t)} + \operatorname{tr} \widetilde{\mathbf{W}}^{(t)} + \sum_{i=1}^{2} \log c_i^{(t)}\right) + \sum_{i=1}^{n} \log C_i^{(t)}$
end procedure

procedure THE VB-EM ALGORITHM
    $t \leftarrow 0$
    while $\mathcal{L}^{(t+1)} - \mathcal{L}^{(t)} > \delta$ or $t < t_{max}$ do
        call UPDATE FOR $\mathbf{y}^*$
        call UPDATE FOR $\mathbf{w}$
        call UPDATE FOR $\lambda$
        call UPDATE FOR $\alpha$
        call CALCULATE LOWER BOUND
        $t \leftarrow t + 1$
    end while
end procedure

return $(\hat{\mathbf{y}}^*, \hat{\mathbf{w}}, \hat{\lambda}, \hat{\alpha}) \leftarrow (\tilde{\mathbf{y}}^{*(t)}, \tilde{\mathbf{w}}^{(t)}, \tilde{\lambda}^{(t)}, \tilde{\alpha}^{(t)})$        ▷ converged parameter estimates
return $(\hat{y}_1, \ldots, \hat{y}_n) \leftarrow \left(\arg\max_{k=1}^{m} \hat{y}_{1k}^*, \ldots, \arg\max_{k=1}^{m} \hat{y}_{nk}^*\right)$        ▷ predicted classes
for $i = 1, \ldots, n$ do
    for $j = 1, \ldots, m$ do
        return $\hat{p}_{ij} \leftarrow \prod_{\substack{k=1 \\ k \neq j}}^{m} \Phi\left(\frac{\hat{y}_{ij}^* - \hat{y}_{ik}^*}{\sqrt{2}}\right)$        ▷ predicted probabilities
    end for
end for
```

Chapter 7

Examples of I-probit models

7.1 Toy examples

Let's look at some toy examples to illustrate classification using I-probit models. First is a binary classification task based on two predictors. This data set consists of 300 points from two spirals with some Gaussian noise added. A plot is shown below.

```
R> spiral <- gen_spiral(n = 300, sd = 0.07)
R> plot(spiral)
```

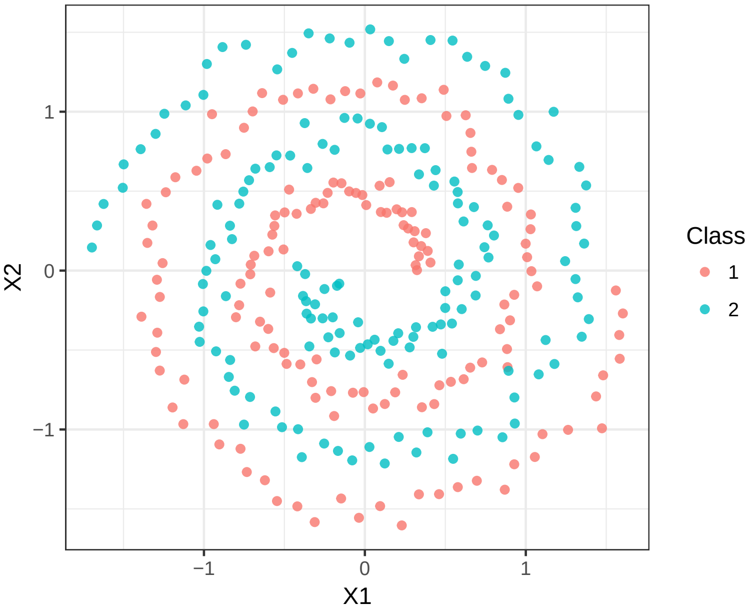


Figure 7.1: Spiral data set.

We tried a few models. First with the linear canonical kernel. This gave very poor results (training error rate of 50% is basically just guess-work). Not surprising because the problem hardly seems linear in nature. Best to go with a smooth function, so we tried the fBm kernel. This gave an improved training error rate (31.3%) but judging by the predictive plot, there still is room for improvement.

```
R> # Bad results, linear functions not able to predict spirals well
R> (mod1 <- iprobit(y ~ X1 + X2, spiral, kernel = "Canonical"))

## ===========
## Converged after 15 iterations.
## Training error rate: 50.00 %
## Lower bound value: -214.0725
##
##     alpha lambda[1] lambda[2]
##   0.00004   0.00000   2.88634

R> iplot_predict(mod1)
```

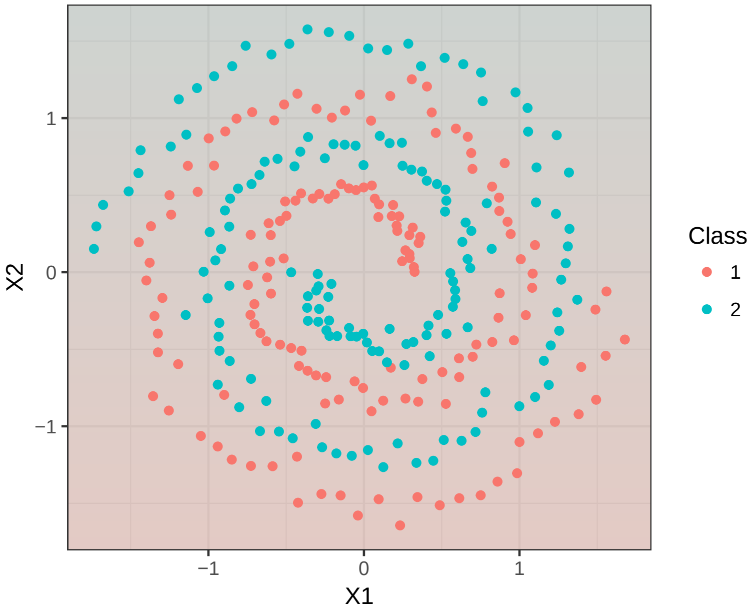


Figure 7.2: Canonical kernel with multiple scale parameters.

```
R> # Getting there, but still not nice
R> (mod2 <- iprobit(y ~ X1 + X2, spiral, kernel = "FBM"))

## ========================================================================
## Convergence criterion not met.
## Training error rate: 31.33 %
## Lower bound value: -204.2227
##
##     alpha lambda[1] lambda[2]
##   0.00484  -0.00263   1.26369
```

```
R> iplot_predict(mod2)
```

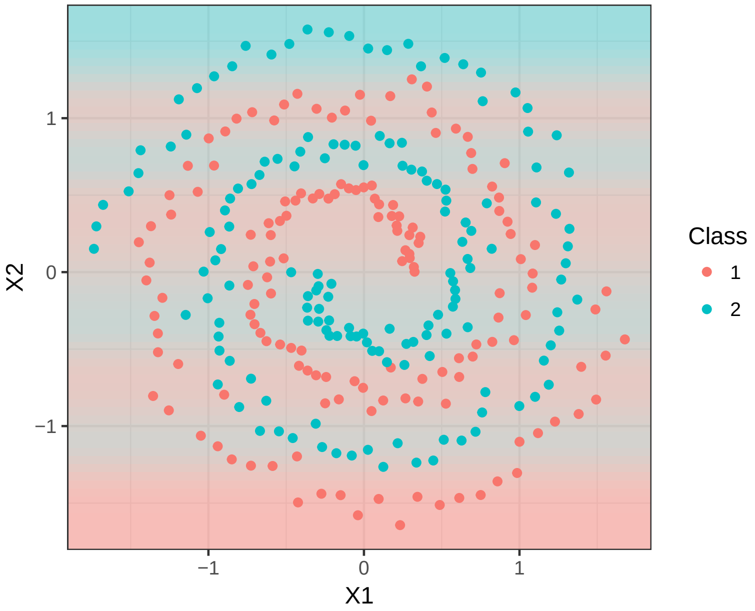


Figure 7.3: fBm kernel with multiple scale parameters.

```
R> # Turns out the scale parameters matter here
R> (mod3 <- iprobit(y ~ X1 + X2, spiral, kernel = "FBM", one.lam = TRUE))
```

```
## ============================================================
## Converged after 82 iterations.
## Training error rate: 1.67 %
## Lower bound value: -162.9976
##
##   alpha  lambda
## 0.00497 5.16273
```

```
R> iplot_predict(mod3)
```

It turns out that retstricting the model to have a single scale parameter works best, coupled with the fBm kernel. This seems to suggest that the two variables are similarly scaled and effects the latent response in a similar magnitude. Indeed, the X_1 and X_2 variables are quite similar in that they are points from two spirals mirroring each other. We are able to get a training error rate of 1.67%, and incidentally this model gives the

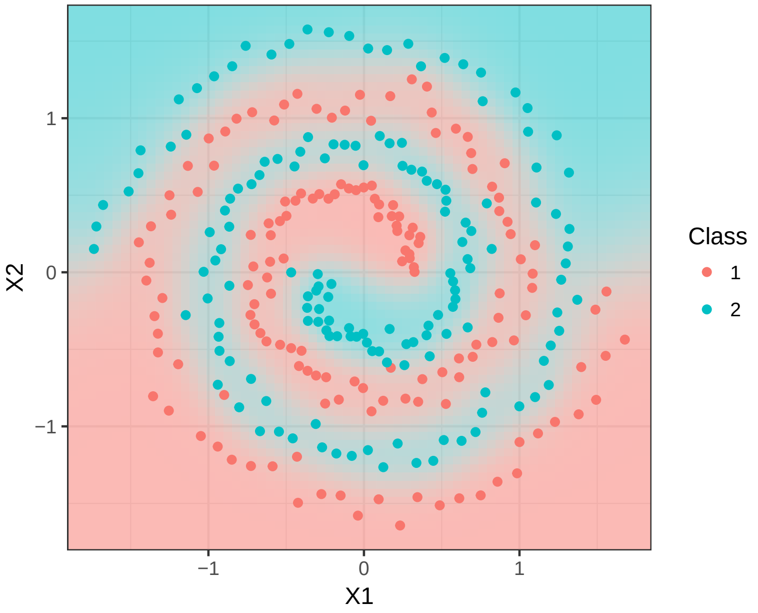


Figure 7.4: fBm kernel with a single shared scale parameter.

highest lower-bound value as well.

One thing that was noticed with this data set was that different starting values led to possibly different converged parameter estimates. This leads us to believe that the variational lower bound to be maximised has multiple local optima. One way to overcome this is to perform multiple restarts and keep the results from the highest lower bound value. This is something to look out for when analysing real-data examples.

The next example is a four-class classification data set that is meant to be linearly separable in two dimensions. Random noise was added to the X_1 and X_2 component from four equidistant points (representing four distinct classes) around a circle of radius three. 125 points were generated for each class, thereby giving a total of 500 data points altogether. Here is a plot of the data set.

```
R> mixture <- gen_mixture(n = 500, m = 4, sd = 1.5)
R> (mod <- iprobit(y ~ X1 + X2, mixture))

## ==========================================================================
## Convergence criterion not met.
## Training error rate: 8.80 %
## Lower bound value: -194.2465
##
##              Class = 1 Class = 2 Class = 3 Class = 4
## alpha         -0.12285  -0.71550  -0.72534  -0.07966
## lambda[1,]   1.28355   0.00000   0.70112   0.00000
## lambda[2,]   0.00000   0.25543   0.00000   1.02738
```

We fit a canonical I-probit model, and get the following results.

```
R> plot(mixture)
R> iplot_predict(mod)
```

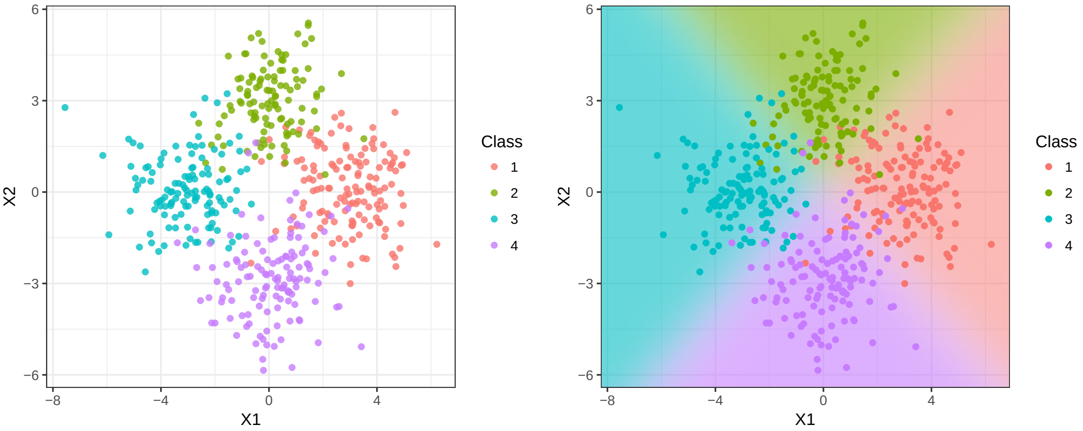


Figure 7.5: Canonical kernel is able to linearly separate the data points.

7.2 Predicting cardiac arrhythmia

Machine learning tools are being used in the field of medicine as a means to aid medical diagnosis of diseases. In this example, factors determining the presence or absence of heart diseses is studied. Traditionally, cardiologists may look at patients' cardiac activity (ECG data) to reach a diagnosis. This of course remains the so-called "gold standard" method of obtaining a diagnosis. The study by Guvenir et. al. aimed to predict cardiac abnormalities by way of machine learning and minimise the difference between the gold standard and computer-based classifications. This data set is made publicly available at [...]. It contains a myriad of ECG readings and other patient attributes such as age, height, and weight. Altogether there are 451 observations and 279 predictors. We excluded nominal covariates, leaving us with 194 continuous predictors, which we then standardised so that we can use a single-scale I-probit model. In the original data set, there are 13 distinct classes of cardiac arrhythmia. We had combined all of these to form a single class, thus reducing the problem to a binary classification task (normal vs. arrhythmia).

Fitting an I-probit model on the full data set takes about 2.5 seconds only, with convergence reached in at most 15 iterations. However, we do find that the training

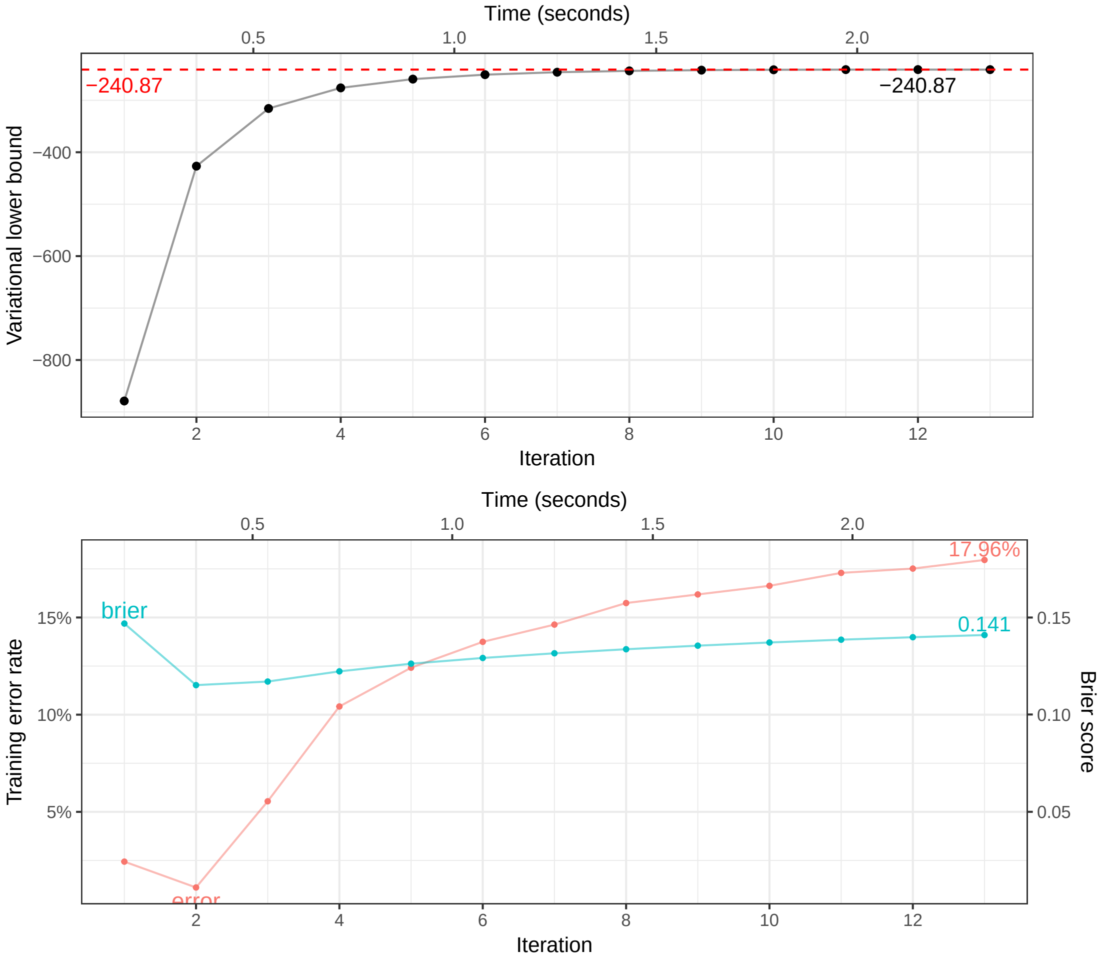


Figure 7.6: (a) Plot of variational lower bound over time. (b) Plot of training error rate and Brier scores over time.

error rates are much better if the model was not allowed to reach full convergence (i.e., stopped early at five iterations, say) Why is this? Need to look at a plot of marginal likelihood. . It is believed that local optima gives better predictive performance, rather than at the global maxima of the (approximate) likelihood.

Figure 7.6(a) plots the variational lower bound value over time and iterations for the cardiac arrhythmia data set. As expected, the lower bound value increases over time until a convergence criterion is reached. In Figure 7.6(b), the training error rate and the Brier score is plotted against time. What we see is that the training error rate worsens over time as the lower bound value reaches its maximum value. There is some reason to terminate the variational algorithm early - while compromising on the lower bound value, we hope to obtain parameter values which give good predictive performance.

To measure predictive ability, we fit the I-probit model with the canonical and fBm-0.5 kernel on a random subset of the data and obtain the out-of-sample test error rates from the remaining observations. We then compare the results against popular machine learning classifiers, namely: 1) k-nearest neighbours; 2) support vector machine; 3) Gaussian process classification (radial basis kernel); 4) random forests; 5) nearest shrunken centroids (Tibshirani et. al. 2003); and 6) L-1 penalised logistic regression. The final model also performs variable selection, something that the I-probit model can do as well, but for now we concentrate on using all the available predictors for training and testing. The experiment is set up as follows:

1. Form a training set by sub-sampling $n \in \{50, 100, 200\}$ observations.
2. The remaining unsampled data is used as the test set.
3. Fit model on training set, and obtain test error rates defined as

$$\text{test error rate} = \frac{1}{n}\sum_{i=1}^{n}[y_i^{\text{pred}} \neq y_i^{\text{test}}] \times 100\%.$$

4. Repeat steps 1-3 100 times to obtain the *average* test error rates and standard errors.

Results for the six methods listed above were obtained from Cannings and Samworth (2017). The results are shown in the plot below.

A plot of the mean test error rates together with the 95% confidence intervals for all models are shown in Figure 7.7. The methods shown in the plot are sorted from the best (top) to the worst (bottom), according to a weighted ranking system which favours better performance in smaller sub-samples. It can be seen that the I-probit models outperform the more popular machine learning algorithms out there including k-nearest neighbours, support vector machines and Gaussian process classification. The fBm I-probit model performed better than the canonical linear I-probit model, which is unsurprising. An underlying smooth function to model the latent variables is expected to generalise better than a rigid straight line function. The fBm I-probit model came second only to random forests, an ensemble learning method, which depending on the number of random decisions trees generated simultaneously, might be slow. The time complexity of a random forest algorithm is $O(pqn\log(n))$, where p is the number of variables used for training, q is the number of random decision trees, and n is the number of observations.

7.3 Meta-analysis of smoking cessation

Data from 27 separate smoking cessation studies in which participants are subjected to a nicotine gum treatment or a placebo. The interest is to estimate the treatment effect size, and whether it is statistically significant. The studies are conducted at different

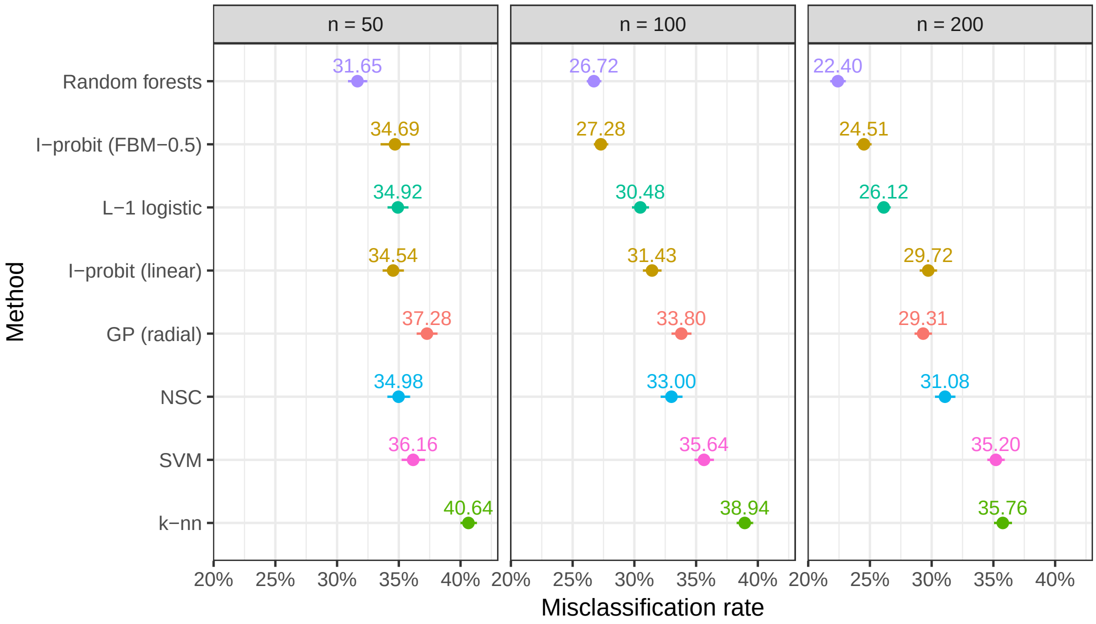


Figure 7.7: Plot of mean test error rates (points) together with the 95% confidence intervals for I-probit models and six popular classifiers.

times and due to various reasons such as funding and cultural effects, the results from all of the studies may not be in agreement. The number of effective participants plays a major role in determining the power of the statistical tests performed in individual studies. The question then becomes how do we meaningfully aggregate all the data to come up with one summary measure?

Several methods exist to analyse such data sets. One may consider a fixed-effects model, similar to a one-way ANOVA model to establish whether or not the effect size is significant. Because of the study-specific characteristics, it is natural to consider multilevel or random-effects models as a means to estimate the effect size. Regardless of method, the approach of analysing study-level treatment effects instead of patient-level data is the paradigm for meta-analysis. However, analysing study-level estimates of effect size can be problematic for various reasons, such as small group samples or rare occurences. Our approach using I-priors looks at patient-level data, but takes into account the levels due to the various study groups.

A summary of the data is displayed by the box-plot in Figure 7.8. On the whole, there are a total of 5908 patients, and they are distributed roughly equally among the control and treatment groups (46.33% and 53.67% respectively, on average). From the box-plots, it is evident that there are more patients who quit smoking in the treatment group as compared to the placebo control group. There are various measures of treatment effect size, such as risk ratio or risk differences, but we shall concentrate on *odds ratios* as

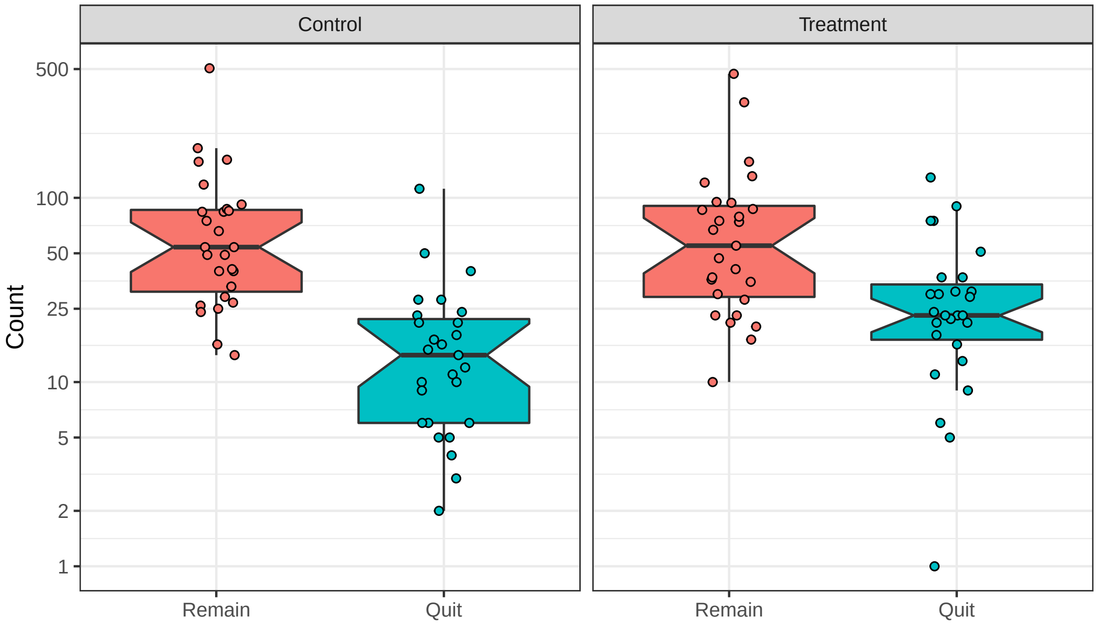


Figure 7.8: Comparative box-plots of the distribution of patients who successfully quit smoking and those who remained smokers, in the two treatment groups.

defined by

$$\text{odds ratio} = \frac{\text{odds of quitting smoking in } \textit{treatment} \text{ group}}{\text{odds of quitting smoking in } \textit{control} \text{ group}}.$$

The odds of quitting smoking in either group is defined as

$$\text{odds} = \frac{\text{P[quit smoking]}}{1 - \text{P[quit smoking]}},$$

and these probabilities, odds and ultimately the odds ratio can be estimated from sample proportions. This raw odds ratio for all study groups is calculated as $1.66 = e^{0.50}$. It is also common for the odds ratio to be reported on the log scale (usually as a remnant of logistic models). A value greater than one for the odds ratio (or equivalently, greater than zero for the log-odds ratio) indicates a significant treatment effect.

A random-effects analysis using a multilevel logistic model has been considered by cite Skrondal Rabe-Hasketh, Agresti and Hartzel . Let $i = 1, \ldots, n_j$ index the patients in study group $j \in \{1, \ldots, 27\}$. For patient i in study j, p_{ij} denotes the probability that the patient has successfully quit smoking. Additionally, x_{ij} is the centred dummy variable indicating patient i's treatment group in study j. These take on two values: 0.5

for treated patients and -0.5 for control patients. The logistic random-effects model is

$$\log\left(\frac{p_{ij}}{1-p_{ij}}\right) = \beta_{0j} + \beta_{1j}x_{ij}$$

with

$$\begin{pmatrix}\beta_{0j}\\ \beta_{1j}\end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix}\beta_0\\ \beta_1\end{pmatrix}, \begin{pmatrix}\sigma_0^2 & \sigma_{01}\\ \sigma_{01} & \sigma_1^2\end{pmatrix}\right)$$

Agresti also made the additional assumption that $\sigma_{01} = 0$ so that, coupled with the contrast coding used for x_{ij}, the total variance $\mathrm{Var}[\beta_{0j} + \beta_{1j}x_{ij}]$ would be constant in both treatment groups. The overall log odds ratio is represented by β_1, and this is estimated as $0.57 = \log 1.76$.

In an I-prior model, the Bernoulli probabilities p_{ij} are regressed against the treatment group indicators x_{ij} and also the patients' study group j via the regression function f and a probit link:

$$\begin{aligned}\Phi^{-1}(p_{ij}) &= f(x_{ij}, j)\\ &= f_1(x_{ij}) + f_2(j) + f_{12}(x_{ij}, j).\end{aligned}$$

We have decomposed our function f into three parts: f_1 represents the treatment effect, f_2 represents the effect of the study groups, and f_{12} represents the interaction effect between the treatment and study group on the modelled probabilities. As both x_{ij} and j are nominal variables, the functions f_1 and f_2 both lie in the Pearson RKHS of functions $\mathcal{F}_1$ and $\mathcal{F}_2$, each with RKHS scale parameters λ_1 and λ_2. As such, it does not matter how the x_{ij} variables are coded (dummy coding 0, 1 vs. centred coding -0.5, 0.5) as the scaling of the function is determined by the RKHS scale parameters. The interaction effect f_{12} lies in the RKHS tensor product $\mathcal{F}_1 \otimes \mathcal{F}_2$. In I-prior modelling, there are only two parameters to estimate, while in the standard logistic random-effects model, there are six. The results of the I-prior fit are summarised in the table below.

Table 7.1: Results of the I-prior model fit for three models.

Model	Lower bound	Brier score	No. of RKHS scale param.
f_1	-3210.76	0.179	1
$f_1 + f_2$	-3092.22	0.168	2
$f_1 + f_2 + f_{12}$	-3091.21	0.168	2

The approximated marginal log-likelihood value for the I-prior model (i.e. variational lower bound), the Brier score for each model and the number of RKHS scale parameters estimated in the model are reported in Table 7.1. Three models were fitted: : 1) A model with only the treatment effect; 2) A model with a treatment effect and a study group effect; and 3) Model 2 with the additional assumption that treatment effect varies across study groups. Model 1 disregards the study group effects, while Model 2 assumes that the effectiveness of the nicotine gum treatment does not vary across study groups (akin to a varying-intercept model). Although not soundly based in theory, we may compare variational lower bounds of the three models for model selection as a proxy to using the true log-likelihood value. In this case, Model 3 has the highest lower bound value. The Brier score indicates the predictive performance of the models, and there is not much to distinguish between the three.

Unlike in the logistic random-effects model, where the log odds ratio can be read off directly from the coefficients, with an I-prior probit model the log odds ratio needs to be calculated manually from the fitted probabilities. The probabilities of interest are the probabilities of quitting smoking under each treatment group for each study group j - call these $p_j(\text{treatment})$ and $p_j(\text{control})$. That is,

$$\begin{aligned} p_j(\text{treatment}) &= \Phi\big(f(\text{treatment}, j)\big) \\ p_j(\text{control}) &= \Phi\big(f(\text{control}, j)\big). \end{aligned}$$

The log odds ratio for each study group can then be calculated as usual. For the overall log odds ratio, the probabilities that are used are the averaged probabilities weighted according to the sample sizes in each group. This has been calculated as $0.49 = \log 1.63$, slightly lower than both the raw log odds ratio and the log odds ratio estimated by the logistic random-effects model. This can perhaps be attributed to some shrinkage of the estimated probabilities due to placing a prior with zero mean on the regression functions.

RE: Fiona's suggestion of discussing the variance, covariance/correlation of the random effects?

7.4 Vowel recognition data

cite Hastie Tibshirani elements of statistical learning . We illustrate multiclass classification using I-priors on a speech recognition data set[1]with $m = 11$ classes to be predicted from digitized low pass filtered signals generated from voice recordings. Each class corresponds to a vowel sound made when pronouncing a specific word. The words that make up the vowel sounds are shown in Table 7.2. Each word was uttered once by multiple speakers, and the data are split into a training and a test set. Four males and four female speakers contributed to the training set, while four male and three female speakers contributed to the test set. The recordings were manipulated using speech processing techniques, such that each speaker yielded six frames of speech from the eleven vowels, each with a corresponding 10-dimensional numerical input vector (the predictors). This means that the size of the training set is 528, while 462 data points are available for

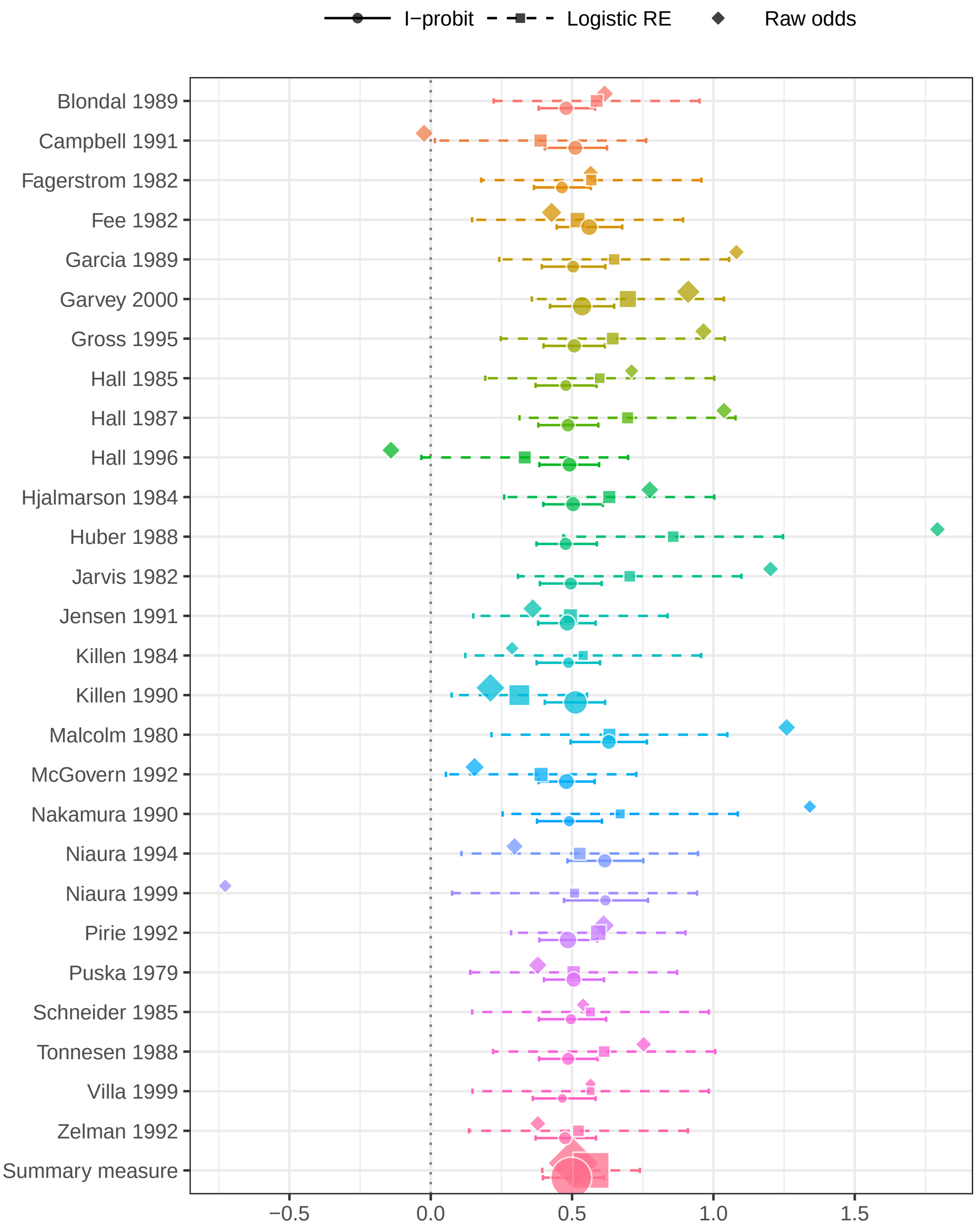


Figure 7.9: Forest plot of effect sizes (log odds ratios) in each group as well as the overall effect size together with their 95% confidence bands. The plot compares the raw log odds ratios, the logistic random-effect estimates, and the I-prior estimates. Sizes of the points indicate the relative sample sizes per study group.

Table 7.2: The eleven words that make up the classes of vowels.

Class	Label	Vowel	Word	Class	Label	Vowel	Word
1	`hid`	iː	heed	7	`hOd`	ɒ	hod
2	`hId`	ɪ	hid	8	`hod`	ɔː	hoard
3	`hEd`	ɛ	head	9	`hUd`	ʊ	hood
4	`hAd`	a	had	10	`hud`	uː	who'd
5	`hYd`	ʌ	hud	11	`hed`	əː	heard
6	`had`	ɑː	hard				

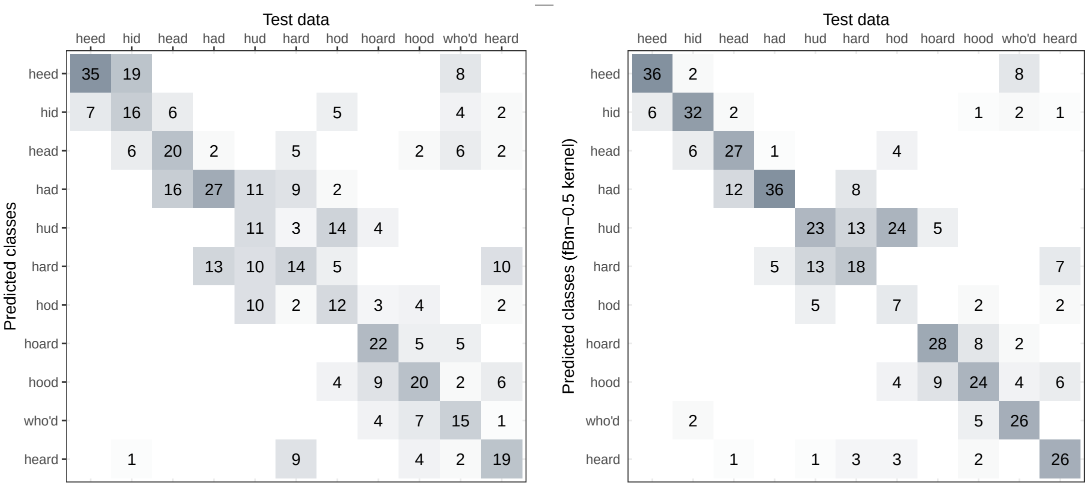
testing the predictive performance of the models. This data set is also known as Deterding's vowel recognition data (after the original collector, cite) or the Connectionist Bench data. Machine learning methods such as neural networks and nearest neighbour methods were analysed by Robinson (cite).

We will fit the data using an I-probit model with the canonical linear kernel and also the fBm-0.5 kernel. We assume $m = 11$ distinct I-priors corresponding to the latent variables in each class, thus there are 11 unique intercepts and 11 RKHS scale parameters to estimate in each model. Each model took roughly 6 seconds per iteration to complete. The canonical kernel model took a long time to converge, with each variational EM iteration improving the lower bound only slighly each time. In contrast, the fBm-0.5 model was quicker to converge, and this is something that we noticed happening for most other data sets as well. Multiple restarts from different random seeds were conducted, and we found that they all converged to a similar lower bound value. This alleviates any worry that the model might have converged to different multiple local optima.

A good way to visualise the performance of model predictions is through a confusion matrix, as shown in Figure 7.10. The numbers in each row indicate the instances of a predicted class, while the numbers in the column indicate instances of the actual classes. Nil values are indicated by blank cells. A quick glance of the plots seem to favour the fBm-0.5 kernel as having better predictions. There are a lot more misclassifications when using the canonical kernel. Under the fBm-0.5 model, the model makes understandable mistakes - confusing very similar words, especially 'hod' and 'hud'.

Comparisons to other methods that had been used to analyse this data set is given in Table 7.3. In particular, the I-probit model is compared against 1) linear regression; 2) logistic linear regression; 3) linear and quadratic discriminant analysis; 4) decision trees; 5) neural networks; 6) k-nearest neighbours; and 7) flexible discriminant analysis. All of these methods are described in further detail in citeHastie Tibshirani. The I-probit model using the fBm-0.5 kernel offers one of the best out-of-sample classification error rates (38.7%) of all the methods compared. The linear I-probit model is seen to

[1] Data is publicaly available from the UCI Machine Learning Repository, URL: `https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+(Vowel+Recognition+-+Deterding+Data)`.



	heed	hid	head	had	hud	hard	hod	hoard	hood	who'd	heard
heed	35	19								8	
hid	7	16	6				5			4	2
head		6	20	2		5			2	6	2
had			16	27	11	9	2				
hud					11	3	14	4			
hard				13	10	14	5				10
hod					10	2	12	3	4		2
hoard								22	5	5	
hood							4	9	20	2	6
who'd								4	7	15	1
heard		1				9			4	2	19

(a) Canonical kernel

	heed	hid	head	had	hud	hard	hod	hoard	hood	who'd	heard
heed	36	2								8	
hid	6	32	2						1	2	1
head		6	27	1			4				
had			12	36		8					
hud					23	13	24	5			
hard				5	13	18					7
hod					5		7		2		2
hoard								28	8	2	
hood							4	9	24	4	6
who'd		2							5	26	
heard			1		1	3	3		2		26

(b) fBm-0.5 kernel

Figure 7.10: Confusion matrices for the vowel classification problem in which predicted values were obtained from the I-probit models. The maximum value for any one cell is 42. Blank cells indicate nil values.

be comparable to logistic regression, linear and quadratic discriminant analysis, and also decision trees. It also provides significant improvement over multiple linear regression.

7.5 Spatio-temporal modelling of bovine tubercolosis in Cornwall

Data containing the number of breakdows of bovine tubercolosis (BTB) in Cornwall, the locations of the infected animals, and the year of occurence is analysed. The interest, as motivated by veterinary epidimiology, is to understand whether or not there is spatial segregation between the herds, and whether there is a time-element to presence or absence of this spatial segregation. There have been previous work done to analyse this data set: cite Diggle et al. (2005) developed a non-parametric method to estimate spatial segregation using a multivariate point process. The occurrences are modelled as Poisson point processes, and spatial segregation is said to have occured if the model-estimated type-specific breakdown probabilities at any given location are not significantly different from the sample proportions I think this is what they did - recheck . The authors estimated the probabilities via kernel regression, and the resulting test statistic had to be estimated via Monte Carlo methods. Other work includes Taylor et al. (2015), who used a fully Bayes scheme for spatio-temporal multivariate log-Gaussian Cox processes.

Table 7.3: Results of various classification methods for the vowel data set.

	Error rates	
Method	Training	Test
Linear regression	48	67
Logistic regression	22	51
Linear discriminant analysis	32	56
Quadratic discriminant analysis	1	53
Decision trees	5	54
Neural networks		45
k-Nearest neighbours		44
FDA/BRUTO	6	44
FDA/MARS	13	39
I-probit (fBm-0.5)	22	39
I-probit (linear)	28	54

Explain data set . $n = 919$ cases in total. Originally there are 11 spoligotypes, but of these, four are most common. Therefore, the rest are combined into a separate class of 'Others'. Total 14 years of data, so total number of classes is $m = 5$.

We are able to investigate any spatio-temporal patterns of infection using I-priors rather simply. Let p_{ij} denote the probability that a particular animal i is infected with the disease with spoligotype $j \in \{1, \dots, m\}$. We model the transformed probabilities $g(p_{ij})$ (as described in the categorical response chapter) as following a smooth function f which takes two covariates: the spatial data x_1 (Northings and Eastings, measured in kilometres), and the temporal data x_2 (year of infection):

$$\begin{aligned} g(p_{ij}) &= f_j(x_1, x_2) \\ &= f_{1j}(x_1) + f_{2j}(x_2) + f_{12j}(x_1, x_2) \end{aligned}$$

We assume a smooth effect of space and time on the probabilities, and an appropriate RKHS for the functions f_1 and f_2 are the fBm-0.5 RKHS. Alternatively, as per Diggle et al., divide the data into four distinct time periods: 1) 1996 and earlier; 2) 1997 to 1998; 3) 1999 to 2000; and finally 4) 2001 to 2002. In this case, x_2 would indicate which period the infection took place in, and thus would have a nominal effect on the probabilities. An appropriate RKHS for f_2 in such a case would be the Pearson RKHS. In either case, the function f_{12} would be the "interaction effect", meaning that with such an effect present, the spatial distribution of the diseases are assumed to vary across the years.

Let h_k, $k \in \{1, 2\}$ denote the reproducing kernel of the spatial and temporal RKHSs

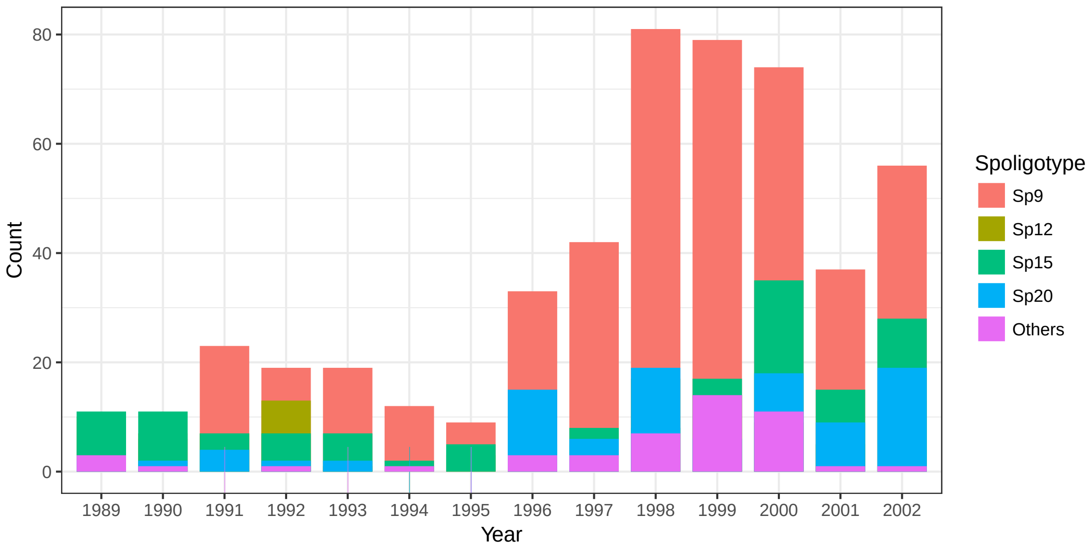


Figure 7.11: Distribution of the different types (Spoligotypes) of bovine tubercolosis affecting herds in Cornwall over the period 1989 to 2002.

respectively. Then, an I-prior on f_j takes the form

$$\begin{aligned} f_j(x_1, x_2) &= \lambda_{1j} \sum_{i=1}^n h_1(x_1, x_{i1}) w_{ij} + \lambda_{2j} \sum_{i=1}^n h_2(x_2, x_{i2}) w_{ij} \\ &\quad + \lambda_{1j}\lambda_{2j} \sum_{i=1}^n h_1(x_1, x_{i1}) h_2(x_2, x_{i2}) w_{ij} \end{aligned}$$

where $\mathbf{w}_j = (w_{1j}, \dots, w_{nj})^\top \sim \text{N}(0, \mathbf{I}_n)$ and each of the $\mathbf{w}_j$ are also independent of each other. The parameters λ_{1j} and λ_{2j} are the RKHS scale parameters for the spatial and temporal covariates respectively. Notice that the functions are indexed by the classes j, such that there would be $2m$ scale parameters to estimate. This is the more general case, in which we assume *separate scale* parameters in each class. However, we may also restrict the scale parameters to be equivalent in each class, so that this so-called *shared scale* model has only two parameters to estimate, which is simpler to do inference. Note that there are also intercept parameters to estimate (one in each class), but these will not be reported as they are irrelevant to the discussion at hand.

Spatio-temporal effects of the BTB breakdowns can be easily inferred through the RKHS scale parameters. The hypothesis of temporal significance is the same as testing the significance of the λ_2 parameter, while the test of both spatial and temporal effects are conducted on λ_1 and λ_2 simultaneously (equivalent to modelling f with a constant). For these tests, it is simpler to infer from the shared scale model, for which we can read the results directly of off Table 7.4. The said table displays the posterior mean estimate of

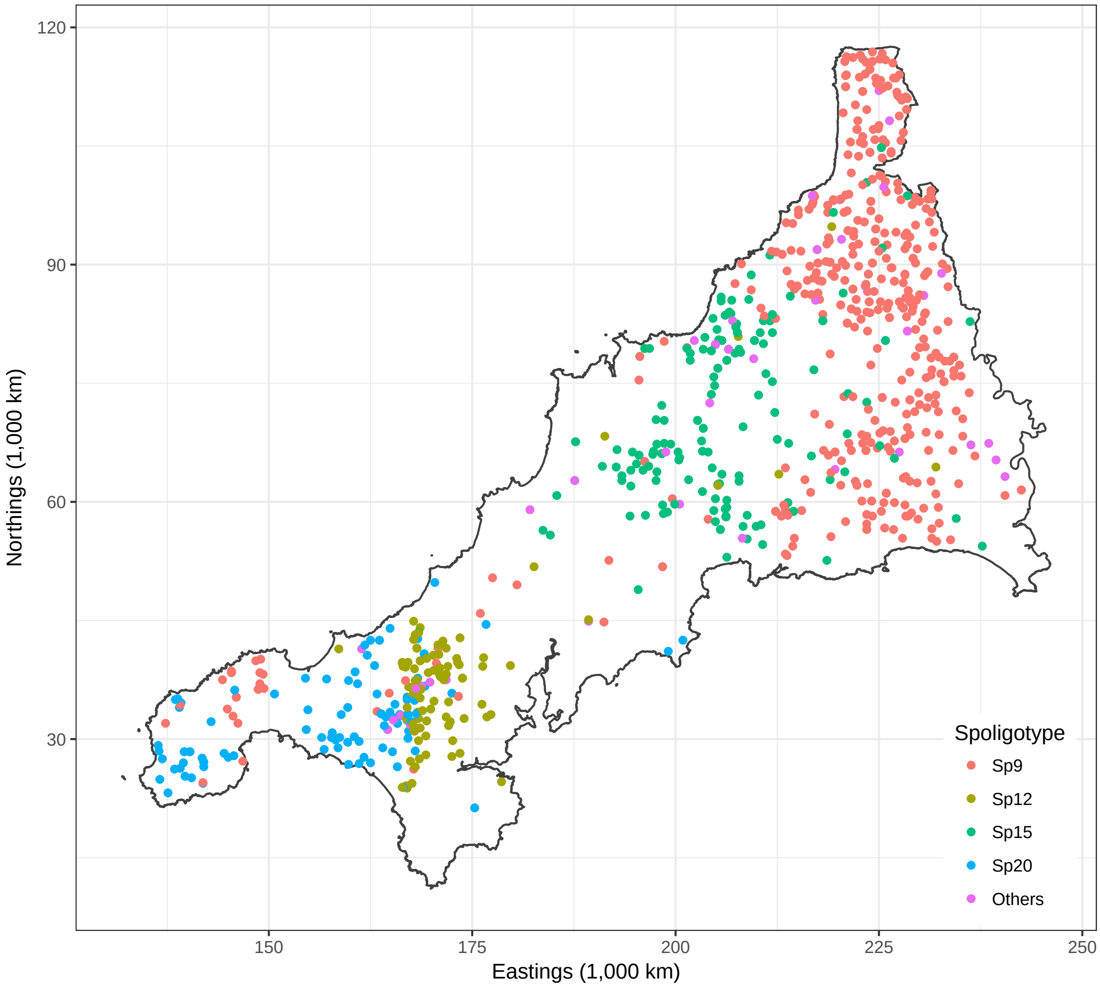


Figure 7.12: Spatial distribution of all cases over the 14 years.

the scale parameters, and together with its posterior standard deviation. From Chapter X, we know that these scale parameters follow a normal posterior distribution, so we can calculate the Z-scores by dividing the mean by its corresponding s.d.. Absolute values greater than three would satisfy a Bayesian hypothesis test of significance at the 0.01 level, for which we see all parameters satisfy in the shared scale model.

Table 7.4: Results of the fitted I-probit models.

	Model											
	Spatial				Spatio-temporal				Spatio-period			
	Estimate	S.D.	$\|Z\|$-score		Estimate	S.D.	$\|Z\|$-score		Estimate	S.D.	$\|Z\|$-score	
Shared scale model												
Spatial	0.19	0.003	64.9	***	0.18	0.003	67.4	***	0.19	0.003	65.6	***
Temporal					0.01	0.000	16.5	***	0.00	0.000	12.0	***
Separate scale model												
Spatial (Sp9)	0.47	0.014	33.5	***	0.48	0.014	33.1	***	0.47	0.014	33.9	***
Spatial (Sp12)	0.19	0.007	29.2	***	0.26	0.008	31.4	***	0.23	0.007	31.3	***
Spatial (Sp15)	0.17	0.005	33.9	***	0.17	0.005	33.6	***	0.17	0.005	33.9	***
Spatial (Sp20)	0.16	0.004	44.2	***	0.17	0.004	39.6	***	0.17	0.004	40.7	***
Spatial (Others)	0.00	0.004	0.0		0.00	0.004	0.0		0.00	0.004	0.0	
Temporal (Sp9)					0.00	0.002	0.1		0.00	0.001	6.3	***
Temporal (Sp12)					0.01	0.001	17.8	***	0.01	0.001	12.4	***
Temporal (Sp15)					0.02	0.001	12.3	***	0.00	0.001	0.0	
Temporal (Sp20)					0.00	0.002	0.1		0.00	0.001	0.1	
Temporal (Others)					0.00	0.002	0.0		0.01	0.001	10.9	***

[*] Lower-bound values (Brier scores) for the shared scale model are -664.8 (0.143), -654.9 (0.135), and -663.7 (0.136) respectively.

[†] Lower-bound values (Brier scores) for the separate scale model are -660.8 (0.138), -667.9 (0.129), and -678.3 (0.130) respectively.

A similar conclusion is reached when inferring from the separate scale model. Instead of individual tests of significance, we now need to test

$$\mathrm{H}_0 : \lambda_1 = \cdots = \lambda_m = 0.$$

We know that by the mean-field approximation used, the λ_js are independent of each other, and therefore a χ^2 test statistic can be built via

$$\chi^2 = \sum_{j=1}^{m} Z_j^2$$

which is then compared against extreme values of the χ^2_m-distribution. As is often the case, separate scale models tend to fit the data better as it gives more generality due to having different scale parameters in each class. This is also the case for the BTB data, where we see from the footnotes of Table 7.4 that the Brier scores for the separate scale models are better than the Brier scores in the shared scale models. For all following plots, we made use of the separate scale model for predicting the surface probabilities. Another comment regarding the models is that the conclusion remains the same if we had used the periodic formulation for x_2.

For a more visual approach, we can look at the plots of the surface probabilities. To obtain these probabilities, we first determined the spatial points (Northings and Eastings) which fall inside the polygon which makes up Cornwall. We then obtained predicted probabilities for each class of disease at each location. Figure 7.13 was obtained using the model with spatial covariates only, thus ignoring any temporal effects. In the case of the spatio-temporal model, we used the model which had the period formulation for time. This way, we can obtain the surface probabilities in only four time periods, although there is no issue with using the continuous time model. It is more economical to display four plots rather than the 14 yearly plots within the margins of this thesis.

As the model suggests, there is indeed spatial segregation for the four most common spoligotypes, and this is also very prominently seen from Figure 7.13. In comparing the distribution of the spoligotypes over the years, we may refer to Figure 7.14. For each time period, we superimpose the actual observations onto the predicted surface probabilities. In addition, coloured dotted lines are displayed to indicate the "decision boundaries" for each of the four spoligotypes. The most evident change is seen to the spatial distribution of spoligotype 12, with the decision boundary giving it a large area in years 1996 and earlier, but this steadily shrunk over the years. Spoligotype 9, which is most commonly seen in the east of Cornwall, seems to have made its way down to the south-west over the years. The other two spoligotypes seem to be rather constant over the years. This is supported also by the spatio-period model results in Table 7.4, where the test of nullity for the scale parameters of these two spoligotypes are not rejected.

7.6 Multi-class multivariate longitudinal data

Classification of psychotropic drugs based on EEG data (brain activity) of rats experiment. It is a longitudinal problem because the effect of the drugs are over a period of time in which the drug is in the system.

I would love to analyse this, but can't find the data set! Tempted to just create simulated data to back analyse, just as a proof of concept.

`http://onlinelibrary.wiley.com/doi/10.1111/j.1467-9876.2007.00575.x/abstract`

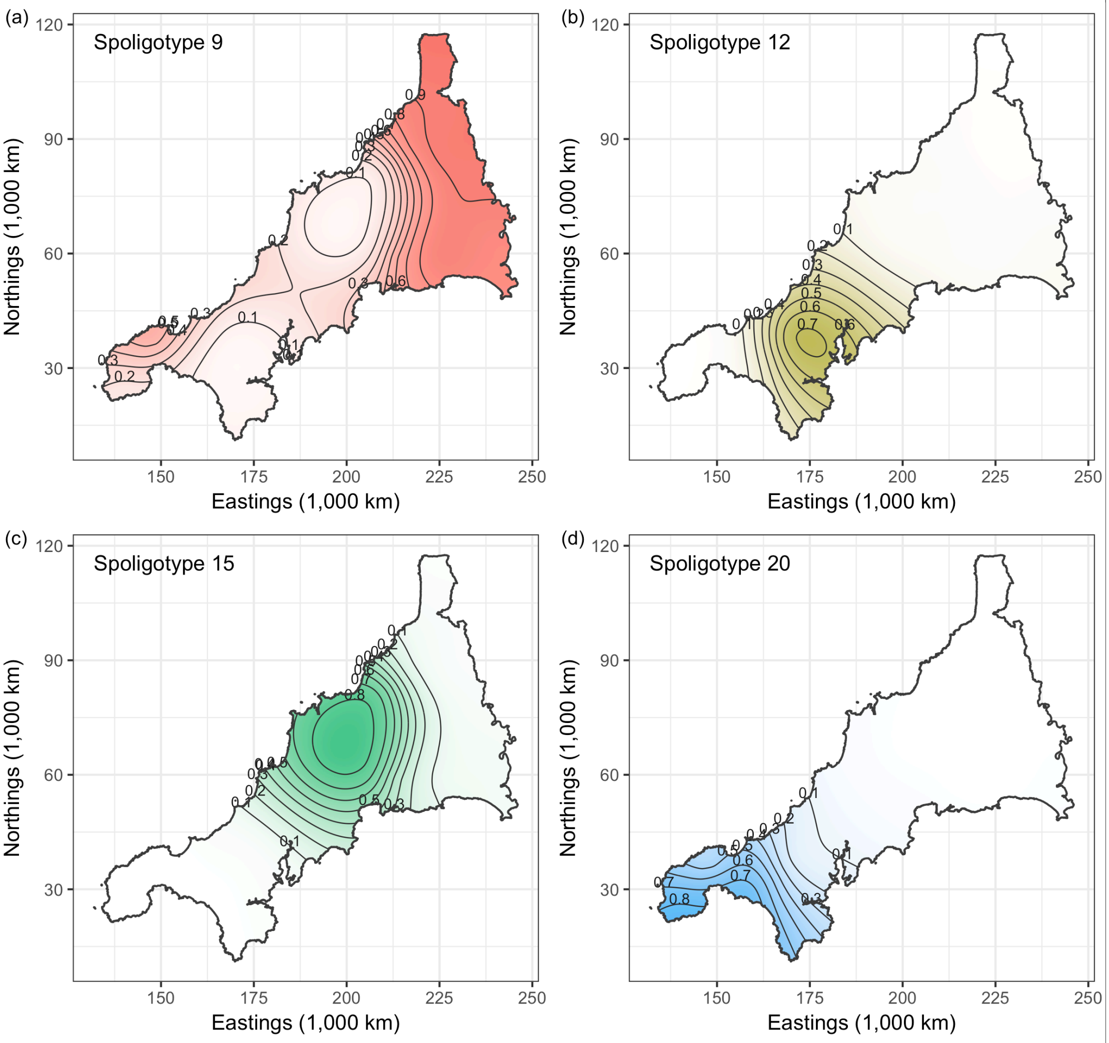


Figure 7.13: Caption.

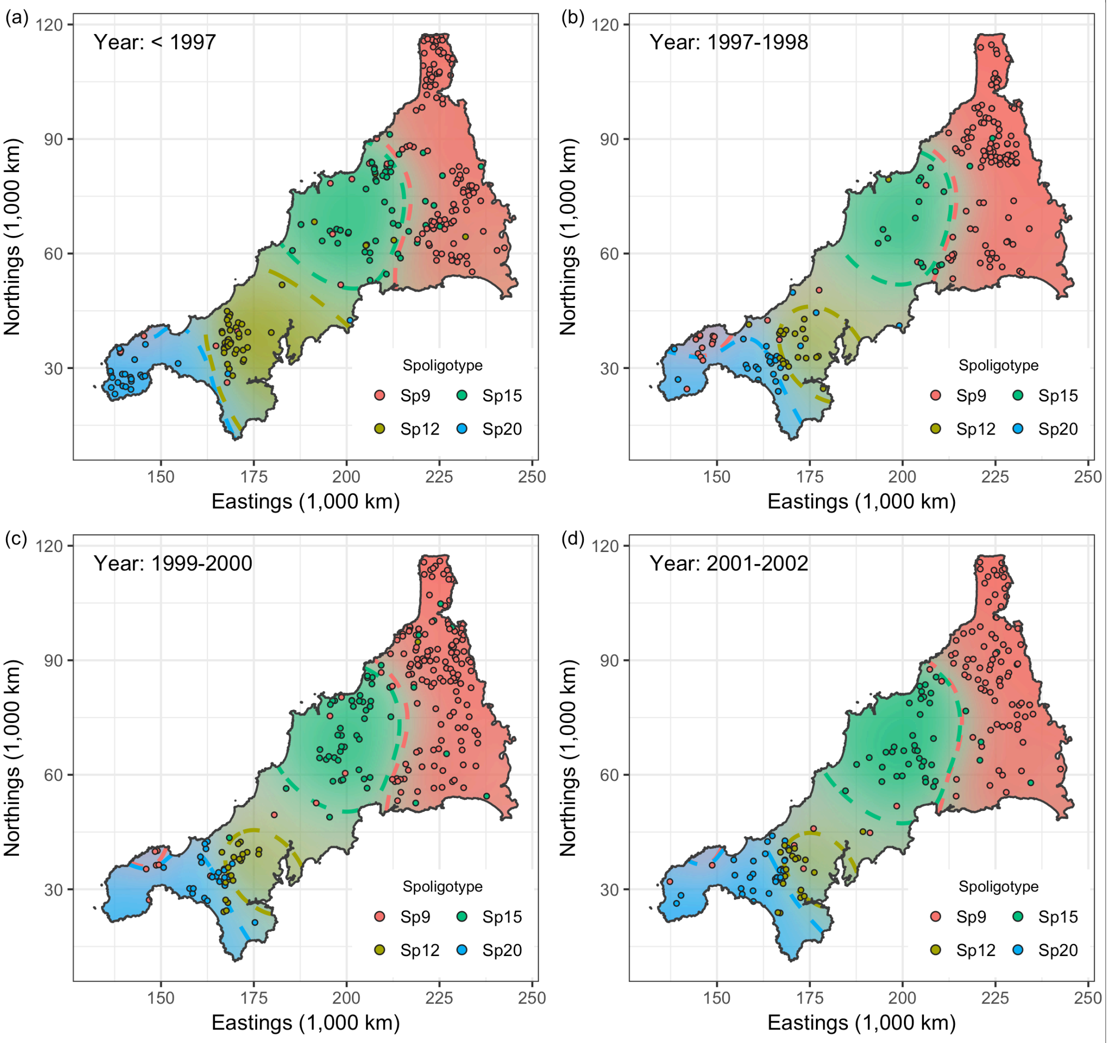


Figure 7.14: Caption.

Chapter 8

Variable selection using I-priors

8.1 Model selection (Empirical Bayes Factors)

A lot of material here. Need to sort it out .

Split into two: BVS by calculating explicit Bayes factors, and when p is large, need to use MCMC methods. Discussion on differences.

Consider the linear regression model, where an $n \times 1$ response vector $\mathbf{y} = (y_1, \dots, y_n)$ relates to several predictors or covariates linearly through the following equation:

$$\begin{aligned} \mathbf{y} &= \boldsymbol{\alpha} + \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \\ \boldsymbol{\epsilon} &\sim \text{N}(\mathbf{0}, \psi^{-1}\mathbf{I}_n) \end{aligned} \tag{8.1}$$

where $\boldsymbol{\alpha}$ is the vector of intercepts ($\boldsymbol{\alpha} = \alpha\mathbf{1}_n$, with $\mathbf{1}_n$ being a vector of ones), $\mathbf{X}$ is an $n \times p$ matrix containing (column-wise) the p observed explanatory variables, and $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_n)$ represents the errors. This linear model is undoubtedly familiar to any statistician, albeit written slightly differently. The constant term, or intercept, α, is segregated from the vector of coefficients $\boldsymbol{\beta}$, thereby allowing us to discard the column of ones typically reserved for the intercept in the design matrix $\mathbf{X}$. Also, we have chosen to work with the precision of the errors ψ, instead of the usual variance $\sigma^2 = 1/\psi$. These errors are assumed to be identically distributed as normal with mean zero and variance $1/\psi$, although one could of course choose to abandon this assumption by specifying $\boldsymbol{\Psi} = (\psi_{ij})$ as the variance-covariance matrix instead. All of these are chosen as a matter of convenience, especially on notation, as we will see later on.

The ordinary least squares (OLS) estimates for the regression coefficients are given as $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$. This is obtained by maximising the normal likelihood of $\boldsymbol{\beta}$, but interestingly, the exact same solution is obtained by minimising the sum of squared errors - without having to set any distributional assumption on the errors. The form of the solution comes from only what is known to us: the data, $\mathbf{X}$ and $\mathbf{y}$.

The Bayesian approach to estimating the linear model takes a different outlook, in that it supplements what is already known from the data with additional information in the form of prior beliefs about the parameters, or simply, priors. Inference about the parameters are then performed on the posterior

$$f(\boldsymbol{\Theta}|\mathbf{y}) \propto \overbrace{f(\mathbf{y}|\boldsymbol{\Theta})}^{\text{likelihood}} \times \overbrace{f(\boldsymbol{\Theta})}^{\text{prior}}$$

such as taking the mean, which is known as the Minimum Mean Squared Error estimate (MMSE), or the mode, which corresponds to the maximum a posteriori estimate (MAP). The Bayesian approach of MAP is similar to maximum likelihood, but differs only in the fact that the optimisation objective (the likelihood function) is augmented with a prior distribution about the parameters. It is critical then, that the prior chosen does not deter us in our cause of finding the correct estimates.

There are many ways of categorising different types of priors, but we like to think that priors can either be pure beliefs (subjective), or chosen according to some principle (objective). For instance, in estimating the chance of rain tomorrow, one might have their own personal feeling about this and elicit a certain probability based on no particular reason, but simply intuition. This is a subjective probability. However, one could also take into account historical data about the chances of rain on a particular day, somewhat more objectively.

In any case, we would also like to categorise priors as either being informative or uninformative, although one could always question the actual informative value in eliciting subjective priors. As the name implies, informative priors aim to help nudge the parameter estimation in the right direction, assuming the prior itself is correct. On the other hand, uninformative priors provide little or vague information about the parameters, and in these cases, the data take over and the prior has little influence on the outcome. One example is the transformation invariant **Jeffreys1946**' (**Jeffreys1946**) prior: $f(\theta) \propto \sqrt{I(\sigma)}$, where $I(\sigma)$ is the Fisher information for σ. For a scale parameter[1] $\sigma \in \mathbb{R}$, the **Jeffreys1946**' prior can be shown to be $f(\sigma) \propto 1/\sigma$, which isn't truly a distribution being a uniform distribution on the real line. Such distributions are known as improper priors. Regardless, these typically yield a proper posterior distribution which we can work with.

The type of prior that is of interest, at least for the purposes of Bayesian variable selection, is one which is objective and ideally informative. The I-prior fits this bill perfectly. A Gaussian I-prior on the regression coefficients $\boldsymbol{\beta}$ has some prior mean $\boldsymbol{\beta}_0$ and covariance matrix equal to the Fisher information for $\boldsymbol{\beta}$. This information theoretic prior for linear models has an intuitive appeal: when there is much Fisher information about the parameters, the covariance matrix for the prior will be large, and thus there will be little influence of the prior mean on the posterior estimate, and vice versa. We

[1] A scale parameter σ for a family of probability distributions satisfies $F(x;\boldsymbol{\theta},\sigma) = F(x/\sigma;\boldsymbol{\theta},1)$, where F is its cumulative distribution function.

typically set the prior mean to be zero for this intuition to work favourably.

We realise there is an oddity in the classification of I-priors as informative. Previously, we alluded that a prior is said to be informative if it helps zone in on the "correct" estimate with the help of this prior, e.g. a normal prior assigned to a parameter with a small variance and prior mean close to the true value (assuming this is known somehow). Conversely, an uninformative prior would have a large variance. The I-prior is either informative or uninformative depending on the amount of Fisher information. Strictly speaking, since the informative-ness of the I-prior depends on the Fisher information, which in turn depends on the data, then technically the I-prior is considered to be uninformative as there really isn't any new information that the prior brings[2]. Any mention of the informativeness of I-priors is then just semantic - in fact, the 'I' in I-prior stands for information.

Circling back to the topic of interest: variable selection, or more generally, model selection. In an ideal world, model selection entails searching the entire model space to find the "best" model based on minimising a certain criterion. There are many such criteria, making model selection a huge topic to cover fully. These include criteria such as (adjusted) R^2, Akaike's information criteria (AIC) and other similar information criteria, Mallow's C_p, (k-fold) cross-validation error, and many others. The obvious issue is that when the dimension of the full model is large, then a search of the entire model space may be computationally prohibitive or even downright unfeasible.

The Bayesian philosophy to model evaluation may be thought of as follows: it is believed that a dataset $\mathbf{Y}$ had been generated from the pdf $f(\mathbf{y}|m_k, \boldsymbol{\Theta}_k)$, where m_k is one of a set of $M = \{m_1, \dots, m_K\}$ models[3], and $\boldsymbol{\Theta}_k$ are the parameters associated with this model. The goal of model selection is then to infer which of the K models had generated the data. The Bayesian approach allows us to assign priors to the parameters and the model index, i.e. $f(\boldsymbol{\Theta}_k|m_k)$ and $f(m_k)$ respectively, and thereby computing the posterior model distribution as

$$\begin{aligned} f(m_k|\mathbf{y}) &\propto f(\mathbf{y}|m_k)f(m_k) \\ &\propto \int f(\mathbf{y}|m_k, \boldsymbol{\Theta}_k) f(\boldsymbol{\Theta}_k|m_k) \, \mathrm{d}\boldsymbol{\Theta}_k \, f(m_k). \end{aligned}$$

The natural criteria for choosing m_k is the one which gives the highest posterior probability. We refer to this model as the maximum probability model.

If we are lucky, our problem may be simple enough that we are able to calculate all of the posterior probabilities, in which case the task is as simple as reading off the maximum probability model from a list of models with their corresponding probabilities.

[2] This is a similar argument as to why the **Jeffreys1946**' prior is considered uninformative. **Liu2014** studied the Kullback-Leibler divergence between the prior and posterior, and noted that this divergence is maximised using **Jeffreys1946**' prior. In other words, this is the prior for which the data brings the maximal amount of information.

[3] We refer to these as models not in the usual sense - more precisely, each m_k is a model class.

However, this is likely not the case, and we often have a large model set to consider. Even if the model set is small, we might find that the integral in the posterior is not analytically tractable. In either of these cases, Markov chain Monte Carlo (MCMC) methods is suitable to be used to overcome these issues of calculating the required posterior probabilities. In fact, MCMC methods can be quite efficient in the exploration of the model space because it will favour models which have great potential of being the true model, and will tend to ignore those that have little to no potential.

While the description we have just given for model selection is generic for most statistical models, variable selection is just a special case of model selection to where the model at hand is defined by the inclusion or exclusion of a finite number of variables. The linear regression model we were describing earlier is such an example. Much work has been done on Bayesian variable selection: **George1993 Kuo1998** and **Dellaportas2002** to name a few. We will be reviewing these methods later on, comparing similarities and differences, strengths and weaknesses, for it is these methods that we intend to improve on by using I-priors. The main motivation behind using I-priors in Bayesian variable selection is its suitability in accommodating to datasets with strong multicollinearity and being able to run with little to no prior information about the parameters.

In this section, we put our Bayesian thinking caps on. Earlier, we introduced the concept of Bayesian model evaluation. Variable selection is just a special case of this whereby a model is defined by the inclusion or exclusion of variables. The linear model (8.1) defined at the beginning is an example of this, and for the remainder of this paper, we will only consider models of this type.

A model is defined as a subset of variables selected from the full set of variables $\{X_1, \dots, X_p\}$ and is linearly related to the response variables through the model equation in (8.1). As each of the p variables can either be selected or not selected, the size of the model space is 2^p. Even for moderate p we can see how the size of the model space can become exponentially large, such that a search of the entire space would be impractical. Note that we do not consider the intercept to be selectable. If this were the case, this would imply a model as having intercept equal to zero as being possible. For most practical modelling purposes, the intercept is almost always non-zero.

It would be useful to be able to index each of these 2^p possible models somehow. We do this by introducing the model identifier vector

$$\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_p).$$

This vector of length p contains elements which indicate whether or not that particular variable was selected. In other words, $\gamma_j = 1$ if X_j was selected in the model, and $\gamma_j = 0$ otherwise for $j = 1, \dots, p$. The full model, where all the variables are included in the model, is denoted by $\boldsymbol{\gamma} = (1, \dots, 1)$. The intercept only model is denoted by $\boldsymbol{\gamma} = (0, \dots, 0)$.

With this in mind, we can then assign priors to the model $f(\boldsymbol{\gamma})$, and also to the parameters of the model $f(\boldsymbol{\Theta}|\boldsymbol{\gamma})$. Ultimately, we are interested in two things:

1. **Posterior inclusion probabilities** $\text{P}[\gamma_j = 1|\mathbf{y}]$ for variable X_j, for $j = 1, \dots, p$. This gives us an indication of how often each variable was selected in the posterior models.
2. **Posterior model probabilities** $\text{P}[\boldsymbol{\gamma} = \boldsymbol{\gamma}_k|\mathbf{y}]$. This gives us a sense of how likely a particular model would appear a posteriori.

The posterior inclusion probabilities can be thought of as the marginals of the posterior model probabilities across each variable. Also, as the distribution on the model probabilities are on a finite set, the posterior distribution is that of a probability distribution function, hence we speak of probabilities instead of densities.

These two types of quantities can be obtained by deriving the posterior distributions for the variable selection model if they are simple enough to be obtained. Sometimes, the relevant expressions are not available in closed form. Alternatively, MCMC methods such as Gibbs sampling can be employed to provide estimates of the quantities of interest. This is perhaps the preferred option, especially when p is large such that the computation all of the 2^p posterior model probabilities takes an unfeasible amount of time. MCMC usually does not list out all of the 2^p probabilities, but instead just the ones which are substantial enough to be deemed important. Models not visited in the MCMC posterior state space are assigned probability zero. Monte-Carlo errors are inevitably introduced into the estimates, but a large enough MCMC run can control these errors.

8.1.1 Overview of Bayesian variable selection methods

We start with an overview of the available methods, in chronological order of appearance in the literature. There are many good in-depth reviews to these methods and the reader may find **OHara2009** or **Chipman2008** useful.

George1993's Stochastic Search Variable Selection (SSVS)

$$\begin{gathered}
y_i = \alpha + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i \\
\epsilon_i \sim \text{N}(0, \psi^{-1}) \text{ iid} \\
i = 1, \dots, n \\
\\
\underline{\text{Prior for } \boldsymbol{\beta}} \\
\beta_j | \gamma_j \sim \gamma_j \text{N}(0, c_j^2 t_j^2) + (1 - \gamma_j)\text{N}(0, t_j^2) \\
j = 1, \dots, p
\end{gathered} \tag{8.2}$$

One of the early works on Bayesian variable selection for linear models come from the **George1993** paper by **George1993** In it, they augmented the indicator variables $\boldsymbol{\gamma}$ into the prior for $\boldsymbol{\beta}$, while the linear model itself remained the same. The prior for β_j is essentially one of two normal distributions, depending on whether or not variable X_j was selected.

The idea behind this type of prior is this: when variable X_j is not important, then γ_j should be equal to zero and the coefficient associated with it β_j should be small and close to zero as possible. Therefore, the prior on β_j should be normal with mean zero and have a small variance t_j^2. Conversely, when the variable X_j is important, then γ_j is one and β_j should be non-zero, and thus the prior on β_j should have a large variance $c_j^2 t_j^2$. In essence, t_j and c_j are tuning parameters that the user must choose. The authors give some suggested values for these tuning parameters: $\big(\text{SE}(\hat\beta_j)/t_j, c_j\big) = (1,5)$, $(1,10)$, $(10,100)$, or $(10,500)$, where $\text{SE}(\hat\beta_j) = \sqrt{\hat\psi^{-1}(\mathbf{X}^\top\mathbf{X})_{jj}}$ under the full model.

The priors on β_j need not be independent of each other. Perhaps a more convenient notation for this prior is

$$\boldsymbol{\beta}|\boldsymbol{\gamma} \sim \text{N}(\mathbf{0}, \mathbf{R}_{\boldsymbol{\gamma}}\mathbf{D}\mathbf{R}_{\boldsymbol{\gamma}}),$$

where $\mathbf{D} = \mathbf{I}_p$, $\mathbf{R}_{\boldsymbol{\gamma}} = \text{diag}[a_j t_j]$ and $a_j = \gamma_j c_j + (1-\gamma_j)$, for $j = 1, \dots, p$. The matrix $\mathbf{D}$ determines the independence of the β_js. Setting this to be the identity matrix implies independence. On the other hand, **George1993** proposed setting this proportional to the inverse sample correlation matrix in order to capture the design correlation.

Kuo1998's sampler (KM)

$$\begin{gathered}
y_i = \alpha + \gamma_1\beta_1 x_{i1} + \cdots + \gamma_p\beta_p x_{ip} + \epsilon_i \\
\epsilon_i \sim \text{N}(0, \psi^{-1}) \text{ iid} \\
i = 1, \dots, n \\
\\
\underline{\text{Prior for } \boldsymbol{\beta}} \\
\beta_j \sim \text{N}(b_j, d_j^2) \\
j = 1, \dots, p
\end{gathered} \tag{8.3}$$

Several years later in **Kuo1998 Kuo1998** published their Bayesian variable selection model, commonly referred to as the independent sampler, so-called because of the independence of the β_js and the γ_js. Instead of having the γ_js augmented into the prior, these are augmented into the model equation itself. Each term $\beta_j x_{ij}$ has its corresponding γ_j multiplied to it. Therefore, when $\gamma_j = 0$, the corresponding term drops out from the model.

The only hyperparameters one needs to choose for this model are the prior means and variances for the normal distributions of the β_js, similar to the Bayesian approach for estimating linear models. These choices reflect one's prior beliefs about the coefficients. In the absence of prior information, one can simply set $b_j = 0$, and choose $d_j = d$ such that $1/2 \leq d \leq 4$ after standardising the $\mathbf{X}$ variables. Otherwise, the user must choose an appropriate value of d_j for each j that would reflect the uncertainty of the estimate β_j being zero.

The appeal of this method is its simplicity, and that also benefits the Gibbs sampling procedure, as the Gibbs conditional densities are easily worked out, and available in a recognisable closed form.

Dellaportas2002's Gibbs Variable Selection (GVS)

$$\begin{gathered}
y_i = \alpha + \gamma_1\beta_1 x_{i1} + \cdots + \gamma_p\beta_p x_{ip} + \epsilon_i \\
\epsilon_i \sim \mathrm{N}(0, \psi^{-1}) \text{ iid} \\
i = 1, \ldots, n \\
\\
\underline{\text{Prior for } \boldsymbol{\beta}} \\
\beta_j | \gamma_j \sim \gamma_j \mathrm{N}(b_j, d_j^2) + (1-\gamma_j)\mathrm{N}(u_j, s_j^2) \\
j = 1, \ldots, p
\end{gathered} \tag{8.4}$$

The authors **Dellaportas2002** worked on an improvement to the current Bayesian variable selection methods, a method which they call the Gibbs Variable Selection (GVS). **Ntzoufras2008** provides an excellent reading about this method in his book, which also provides a good tutorial on using WinBUGS to estimate such models.

At first glance, their model looks like a cross between SSVS and KM, in that the $\boldsymbol{\gamma}$ indicators appear both in the model and in the prior. There are two priors for $\boldsymbol{\beta}$: one is the actual prior, and one which they call the "pseudo prior". This pseudo prior does not make its way into the posterior, and therefore does not influence the estimate at all. Instead, it is there just to make sampling more efficient, according to the authors. Why? When γ_j is one, then β_j is sampled from the posterior with the actual prior. This, coupled with the appropriate hyperparameters b_j and d_j, should encourage β_j to be non-zero. On the off-chance that γ_j is zero when the variable X_j is important, then β_j is sampled from the posterior with a pseudo prior which is designed such that good values for β_j are proposed. If the data (likelihood) also encourages β_j to be non-zero, then there is a high chance that γ_j will flip back to being one. In short, the pseudo prior helps flip the gamma in the right direction, if and when it needs to be flipped, and therefore spends less time being in the wrong state space.

With this model you do need to choose several tuning parameters. As before, we can choose $b_j = 0$ and $d_j = d$ with large d (after standardising $\mathbf{X}$) if no prior information. As for the pseudo prior hyperparameter, **Dellaportas2002** suggests the following choices:

1. $u_j = \hat{\beta}_j$, the estimates of a full pilot MCMC run, and correspondingly $s_j^2 = \widehat{\text{Var}}(\hat{\beta}_j)$.
2. $u_j = 0$ and $s_j^2 \propto d_j^2$, but kept low.

Remark 6. In the long run, we expect the KM and GVS methods to give identical results if the same prior $N(b_j, d_j^2)$ for the β_js are used. As mentioned, the pseudo prior in the GVS method merely improves efficiency of the Gibbs sampler.

Choices of priors

The main difference between the the three methods above, apart from the model structure itself, is the prior specified for $\boldsymbol{\beta}$, but the priors for the rest of the common parameters are and can be similar. The priors for the intercept α and precision ψ are chosen as the conjugate normal-gamma prior, and the prior for each $\gamma_1, \dots, \gamma_p \in \{0, 1\}$ is of course chosen to be Bernoulli:

$$\underline{\text{Priors for } \boldsymbol{\gamma},\ \alpha, \text{ and } \psi}$$

$$\gamma_j \sim \text{Bern}(p_j),\ \ j = 1, \dots, p$$

$$\alpha \sim \text{N}(a, b^2)$$

$$\psi \sim \Gamma(c, d)$$

Typically, in the absence of any prior knowledge, the hyperparameters are set to reflect an uninformative prior. For the normal-gamma, this implies the normal having mean $a = 0$ and large variance b^2, while the gamma having both shape and scale parameters c and d small. Note that a $\Gamma(c, d)$ distribution becomes the Jeffrey's prior as c and d approaches zero. On the other hand one may actually have some prior knowledge about these and may set these hyperparameters accordingly. In any case, we are not too concerned about estimating the intercept and precision parameters.

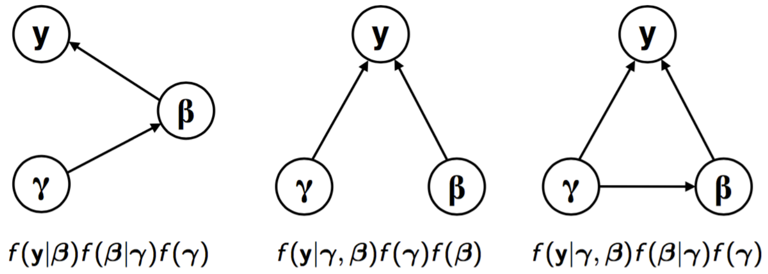
For the Bernoulli prior on the indicator variables, we can appeal to the principle of indifference and set all $p_j = 1/2$, as each variable may either be selected or not selected. Another possibility is to let the model estimate this common probability $p_1 = \cdots = p_p = p$ by assigning a hyperprior such as a beta distribution. The beta hyperprior can be chosen to be uninformative, such as Beta(1,1) (Uniform distribution) or Beta(1/2,1/2) (Jeffrey's prior). The user may also choose to code more complex relationship between the variables - e.g. if variable X_1 is included, then X_2 must be included - useful when performing variable selection on interaction effects. This way, the priors on $\gamma_1, \dots, \gamma_p$ will not be independent, and care must be taken when deriving the posteriors.

Estimate common p

8.1.2 The I-prior Bayesian variable selection model (I-prior)

When we compare the three models side-by-side, there are obvious differences in the structure of the models. Things such as the structure of the parameter space, the amount of tuning parameters that need to be set, and how the two sets of parameters of interest $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ behave in each model. Figure 8.1 summarises these differences.

However, in practice there isn't generally much to distinguish between these three models. It is more than likely that each of these methods will be optimal for a specific problem that the user faces, rather than one having an all-out advantage over the other in all situations. Having said that, these three methods have one thing in common, which is that they do not perform very well when faced with selection scenarios with correlated variables. This may be because of the use of independent $\boldsymbol{\beta}$ priors which lessens the posterior correlations, and thus smaller models tend to be selected (**George1993**).


y
β
γ
$f(\mathbf{y}|\beta)f(\beta|\gamma)f(\gamma)$
y
γ
β
$f(\mathbf{y}|\gamma,\beta)f(\gamma)f(\beta)$
y
γ
β
$f(\mathbf{y}|\gamma,\beta)f(\beta|\gamma)f(\gamma)$

	SSVS	**KM**	**GVS**
Parameter space	Retains original	Does not retain original	
Tuning parameters	Many	None	Some
Priors for β	$\beta\|\gamma \sim \mathrm{N}(\mathbf{0}, \mathbf{R}_\gamma \mathbf{D} \mathbf{R}_\gamma)$ $\mathbf{D} = \mathbf{I}_p$ $\mathbf{R}_\gamma = \mathrm{diag}(a_j t_j)$ $a_j = (1-\gamma_j) + \gamma_j c_j$	$\beta \sim \mathrm{N}(\mathbf{0}, \mathbf{D})$ $\mathbf{D} = d^2\mathbf{I}_p$	$\beta\|\gamma \sim \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ $\boldsymbol{\mu}_j = (1-\gamma_j)u_j$ $\boldsymbol{\Sigma}_{jk} = \gamma_j\gamma_k(d^2\mathbf{I}_p)_{jk}$ $+(1-\gamma_j\gamma_k)\mathbf{1}_{[j=k]}s_j^2$

Figure 8.1: A summarised comparison of the three Bayesian variable selection methods. Graphical models are also illustrated for each method.

We now see an opportunity to use I-priors in the Bayesian variable selection methods, by simply replacing the prior covariance matrix $\mathbf{D} = d^2\mathbf{I}_p$, or in the case of SSVS $\mathbf{D} = \mathbf{I}_p$, by that of the I-prior covariance matrix $\psi\boldsymbol{\Lambda}\mathbf{X}^\top\mathbf{X}\boldsymbol{\Lambda}$ (see the end of Section 2.4.2). **George1993** have already suggested to use $\mathbf{D} \propto (\mathbf{X}^\top\mathbf{X})^{-1}$ as a means of replicating the design correlation, and this turns out to be a generalisation of the g-prior for $\boldsymbol{\beta}$, though it seems to have the opposite effect. This is discussed in Section **??**.

The question is which of the three methods shall we peruse I-priors? The unappealing feature of SSVS is the need to set the tuning parameters before running the model. **George1993** do give four possible suggestions in their paper, but this is thought to be non-exhaustive. In other words, the user must really know the optimal settings for their problem at hand before running the variable selection model. On this note, GVS also has some tuning parameters to set, but not as many in our opinion. As the prior for $\boldsymbol{\beta}$ comprises of a true prior and pseudo prior. The obvious choice for the true prior is the I-prior (if we want to employ I-priors, that is). As for the pseudo prior, **Ntzoufras2008** uses the estimates obtained from a full pilot MCMC run (see Section 8.1.1) in his examples, and this seems reasonable.

Out of all these methods, KM stands out as being the simplest. I-priors fit straight into the story by replacing the prior on $\boldsymbol{\beta}$ in the model. We think this simplicity outweights the efficiency claimed to be brought about by introducing a pseudo prior in GVS. Further, the KM entails only specifying choices for the hyperparameters of the model as we would if we were estimate the linear model in a Bayesian manner. Since we are using the I-prior, there is no more hyperparameters to choose. This is a nice feature seeing it from a "hands-free plug-and-play" perspective.

More justification required

The I-prior Bayesian variable selection model is given below:

$$
\begin{gathered}
y_i = \alpha + \gamma_1\beta_1 x_{i1} + \cdots + \gamma_p\beta_p x_{ip} + \epsilon_i \\
\epsilon_i \sim \mathrm{N}(0, \psi^{-1}) \\
i = 1, \dots, n \\
\\
\underline{\text{Priors}} \\
\boldsymbol{\beta} \sim \mathrm{N}(\mathbf{0}, \psi\boldsymbol{\Lambda}\mathbf{X}^\top\mathbf{X}\boldsymbol{\Lambda}), \ \text{where } \boldsymbol{\Lambda} = \mathrm{diag}[\lambda_1, \dots, \lambda_p] \\
\gamma_j \sim \mathrm{Bern}(p_j), \ j = 1, \dots, p \\
\alpha \sim \mathrm{N}(a, b^2) \\
\psi, \lambda_1^{-2}, \dots, \lambda_p^{-2} \sim \Gamma(c, d)
\end{gathered}
\tag{8.5}
$$

By virtue of the I-prior being a maximum entropy prior, meaning that it is suitable to be used in the absence of prior information, we then complete the model specification above by also choosing uninformative hyperpriors (see Section 8.1.1).

The scale parameters $\lambda_1, \dots, \lambda_p$ originally came from I-prior modelling in a function space framework, whereby these scale parameters help resolve the arbitrary scale of the space of functions over the set of covariates. As these scale parameters make their way into the covariance matrix of the $\boldsymbol{\beta}$ prior, we can interpret them as follows: if no scale parameters are introduced, or equivalently all scale parameters are equal to one, then the covariance matrix is proportional to $\mathbf{X}^\top\mathbf{X}$. As the covariates are likely to be measured on differing scales, such as age in years, height in metres, weight in kilograms, etc., the entries of $\mathbf{X}^\top\mathbf{X}$ will be large for measurements on a large scale range (e.g. body weight in grams), and small for measurements on a small scale range (e.g. body height in metres).

This in turn affects the precision of the prior and consequently the estimation of the $\boldsymbol{\beta}$ parameter. For instance, a high precision (small variance) supports the predictor not being selected. What is ideal for us is that important variables should have β_js estimated as non-zero and vice-versa, but simply putting $\mathbf{X}^\top\mathbf{X}$ as the covariance matrix does not contribute towards this goal. Therefore, scaling the prior covariance matrix of $\boldsymbol{\beta}$ is necessary, and the I-prior method of scaling is a natural choice here.

An alternative solution, as is practiced by the three methods in the previous section, is to standardise both the $\mathbf{X}$ and $\mathbf{y}$ variables. This is indeed a good idea, but is slightly unsatisfactory - scaling the variables so that each has variance one feels ad-hoc in the face of it. Having scale parameters estimated through the model seems more elegant and conforms more to the original I-prior methodology. Having said that, standardising the variables while using a single estimable scale parameter λ is certainly an option, as all the variables would then have been scaled equally via standardisation. This has the advantage of taming extremely large entries of $\mathbf{X}^\top\mathbf{X}$ which may be problematic computationally when we require the inverse.

Remark 7. On another note, it might also possible to treat the scale parameters $\lambda_1, \dots, \lambda_p$ as fixed, having being estimated from the full model using the original I-prior framework described in Section **??**. This is an idea yet to be explored, and is not known whether this would yield good results. There is also a convergency and accuracy issue in obtaining reliable estimates of a large number of scale parameters through the EM procedure of maximising the likelihood of the I-prior model.

We can estimate this model by Gibbs sampling. Unlike the KM model however, one of the Gibbs conditional posterior was not found to be in closed form, which was the posterior for the precision ψ. So to estimate this model, one has to incorporate a Metropolis-Hastings step for the estimation of ψ. The conditional posterior densities are given in Appendix **??**. We can also feed this model into WinBUGS or JAGS which is then able to estimate this model for us.

8.1.3 Simulation study

In this section, we compare the performance of the four methods of Bayesian variable selection: SSVS, KM, GVS, and I-prior by means of a simulation study. The experiment is to select from $p = 100$ variables of a $n = 150$ sample size artificial dataset which has pairwise correlations between the variables. This was inspired by the studies done by **George1993** and **Kuo1998** in their respective papers, albeit on a larger scale (in theirs, $p = 30$).

More simulations required

Compare LASSO

The data was generated as follows:

- Draw $\mathbf{Z}_1, \dots, \mathbf{Z}_{100} \sim \mathrm{N}(\mathbf{0}, \mathbf{I}_{150})$.
- Draw $\mathbf{U} \sim \mathrm{N}(\mathbf{0}, \mathbf{I}_{150})$.
- Let $\mathbf{X}_j = \mathbf{Z}_j + \mathbf{U}$. This induces pairwise correlations of about 0.5.[4]

- Draw $\boldsymbol{\epsilon} \sim \mathrm{N}(\mathbf{0}, 2^2\mathbf{I}_{150})$.
- Generate response variables $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta}_{\text{true}} + \boldsymbol{\epsilon}$.

Let $\boldsymbol{\beta}_{\text{true}} = (\boldsymbol{\beta}_{-k}, \boldsymbol{\beta}_k)$, where $\boldsymbol{\beta}_{-k} = (\beta_1, \ldots, \beta_k) = (0, \ldots, 0)$ and $\boldsymbol{\beta}_k = (\beta_{k+1}, \ldots, \beta_{100}) = (1, \ldots, 1)$. In other words, only variables X_{k+1} to X_{100} are used. The experiment involves varying the value of k between 10, 25, 50, 75 and 90 to create five scenarios, which we label as Scenarios A to E respectively. The two extremes, Scenarios A and E, are meant to simulate situations in which there are a lot of non-zero betas in the true model (Scenario A) and situations in which there are very few non-zero betas in the true model (Scenario E). The variable selection is conducted with many correlated variables.

10,000 MCMC were samples obtained for each scenario, and the metric of interest is the number false choices the models make, i.e. selecting variables which were not in the true model and failing to select variables which were in the true model. Each experiment was repeated 10 times and results averaged, as this ensures that a good result was not simply due to chance of a good random seed in the data generation step. This experiment was conducted in R using JAGS, a variation of WinBUGS, and the results presented in the form of histograms in Figure 8.2. Note that the same prior for $\boldsymbol{\beta}$ was used in the KM and GVS method, so we expect the results to be similar for these two methods.

The ideal picture would be a histogram with a lot of mass towards the left side of the graph, indicating models which produced little false choices. The histograms indicate similar behaviour for the SSVS, KM and GVS methods. These methods perform poorly in the presence of many non-zero $\boldsymbol{\beta}$s, but perform slightly better in the presence of few non-zero $\boldsymbol{\beta}$s. For I-priors however, it is the opposite situation. The I-prior method seems to work quite well given many non-zero $\boldsymbol{\beta}$s, but performance worsens when there are actually few non-zero $\boldsymbol{\beta}$s in the true model. However, in the defence of I-priors, when it does well in Scenarios A, B and C, it does much better (fewer false choices) than when the other three methods do well (Scenarios D and E). I-prior is also less worse than when the other methods do terribly (maximum number of false choice for I-prior is 30, compared to ≈ 50 for any of the other three methods).

One possible explanation here is that when there are a lot of zero $\boldsymbol{\beta}$s in the true model, the Fisher information in the covariance matrix of the prior only serves to confuse with all this evidently unnecessary information. Hence, we can't expect I-priors to benefit in situations like these. Scenarios D and E bode well for the other three methods because the lack of a correlation structure in the covariance matrix of the priors causes these methods to select fewer variables, and thus make fewer false choices.

[4] $\mathrm{Cov}(X_j, X_k) = \mathrm{Cov}(Z_j + U, Z_k + U) = \mathrm{Var}\ U = 1$, and $\mathrm{Var}(X_j) = \mathrm{Var}(Z_j + U) = 2$. Thus, $\mathrm{Corr}(X_j, X_k) = \mathrm{Cov}(X_j, X_k)/(\mathrm{Var}(X_j)\mathrm{Var}(X_k))^{1/2} = 1/2$.

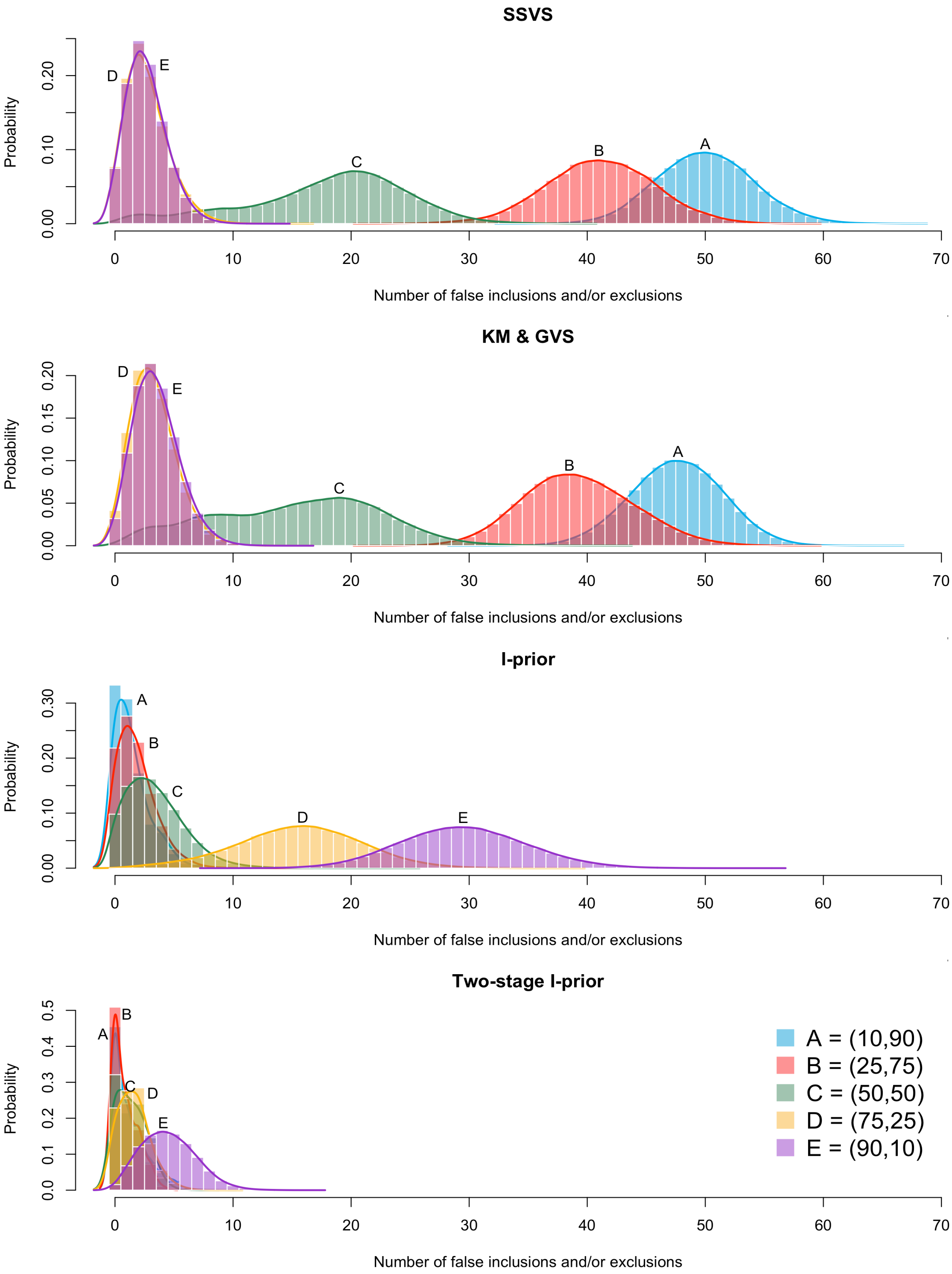


Figure 8.2: Histograms of false choices for SSVS, KM (equivalent to GVS), I-prior and two-stage I-prior compared across the five scenarios A to E. Each scenario is labelled as "$(k, 100 - k)$", where k denotes the number of zeros in the true value of $\boldsymbol{\beta}$ used.

8.1.4 Two-stage procedure

Since I-priors do quite well in Scenarios A-C, but not in D and E, why don't we try to make Scenarios D and E a bit more like Scenarios A-C? This requires some sort of pre-selection of the variables in order to trim off the unwanted variables before running the variable selection model. Without appealing to other pre-selection methods, there is some information from the Bayesian variable selection models that we can make use of - the posterior inclusion probabilities for each variable. As this gives an overall indication as to how valuable a particular variable is, we look for ways to incorporate this into our decision-making process of pre-selecting the variables. The obvious solution is to run the model twice:

1st Run the model. Discard variables with posterior inclusion probabilities less than τ, a treshold value.

2nd Re-run the model on the set of reduced variables.

A natural choice for τ would be 0.5. Setting it at 0.5 would mean that we only keep variables which have a better than equal chance of being selected. The model such that all posterior inclusion probabilities are greater than or equal to 0.5 is known as the median probability model. While this value of 0.5 may seem slightly arbitrary to some, **Barbieri2004** had done some work on median probability models, for which they had shown that under certain strict conditions, these models are also the most optimally predictive models that is able to be selected. The notion of two-stage procedures are not new, as many variable selection methods in the literature generally employ a pre-selection method before running their methods proper. For a two-stage procedure based on posterior inclusion probabilities, **Fouskakis2008** and **Ntzoufras2008** have employed this in their work.

The histogram at the bottom of Figure 8.2 shows that this two-stage procedure does indeed improve on the I-prior method. We see that a shift in the histogram towards the left-hand side of the graph for the second stage run of the model. Interestingly, not only does this improve the Scenarios D and E, we also seem improvements for Scenarios A-C. *Remark* 8. Among the reasons for a pre-selection of the variables are to remove highly correlated variables, removing variables which have no theoretical benefit, or simply to reduce the large number of variables for large especially when $p > n$. There is probably no justification why a two-stage procedure would work better than just a one-stage procedure other than for convenience. At the end of the day, it is the responsibility of the user to interpret the results of the variable selection methods carefully and not make inference blindly on the results of the model.

8.1.5 Real world applications

Here we look at three real-world applications on problems where there are some degree of multicollinearity in the datasets, and have been looked at before from a variable selection standpoint so that we are able to compare results to I-priors.

Aerobic fitness data

This dataset appeared in the *SAS/STAT® User's Guide* **SAS2008** and was also analysed by **Kuo1998** It involves understanding the factors which affect aerobic fitness, which is measured by the ability to consume oxygen. A sample of $n = 30$ male participants' had their physical fitness measured by means of simple exercise tests. The response variable `Oxygen` contains measurement of oxygen uptake rate in mL/kg body weight per minute. The six covariates were the participants' `Age`, `Weight`, time taken to run one mile (`RunTime`), resting heart rate (`RestPulse`), heart rate while running (`RunPulse`), and maximum heart rate during the exercise (`MaxPulse`). This dataset, although small in size, is interesting to analyze because of the correlations between the variables, mainly due to the measurements being taken during the same exercise test. The sample correlations of interest are shown in Figure 8.3 below:

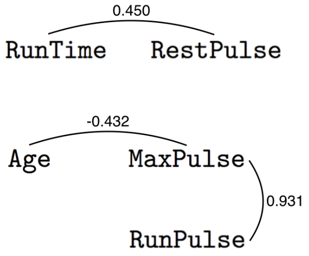


Figure 8.3: The sample correlations of interest in the aerobic fitness dataset. These show variables with correlations greater than 0.4 in magnitude.

The SAS analysis employed a forward selection and backwards elimination procedure and concluded that variables `Weight` and `RestPulse` were to be deleted. The KM procedure concurred with this finding. The I-prior method also did not choose the `Age` variable in addition to `Weight` and `RestPulse` in the maximum probability model. The variable `Age` only had a probability of 0.05 of being included in any posterior model, and also failed to appear in the top 92% of likely models. This can be explained by the correlation between `Age` and `MaxPulse` - supposedly the information encoded in `Age` has already been taken care of by `MaxPulse`, so the I-prior deemed this as surplus. However, the models that had `Age` selected performed better in terms of AIC, Mallows C_p, and 5-fold cross validation root mean squared error (RMSE). Despite this, the strength of the coefficients for the variables are comparable to that of the I-prior method, which is settling if one wishes to do inference on this.

	Full model	**I-prior**	**Forward sel.**	**Back elim.**
`Intercept`	104.2 (0.00)	80.8 (0.00)	103.3 (0.00)	98.6 (0.00)
`Age`	-0.24 (0.03)		-0.25 (0.02)	-0.21 (0.05)
`Weight`	-0.08 (0.15)		-0.08 (0.15)	
`RunTime`	-2.59 (0.00)	-2.97 (0.00)	-2.64 (0.00)	-2.75 (0.00)
`RestPulse`	-0.02 (0.72)			
`RunPulse`	-0.38 (0.00)	-0.38 (0.01)	-0.39 (0.00)	-0.36 (0.01)
`MaxPulse`	0.32 (0.03)	0.36 (0.02)	0.32 (0.03)	0.28 (0.05)
C_p	7.0	7.7	5.1	5.3
AIC	56.8	58.5	54.9	55.6
5f-CV RMSE	2.59	2.71	2.50	2.54

Table 8.1: The OLS estimates for each variable are given in the table above, along with the standard errors in parantheses. The table also shows the value for Mallow's C_p, AIC and 5-fold cross validation RMSE given for each model.

Mortality and air pollution data

The next real world application comes from a paper by **McDonald1973** In it, the effects of air pollution on mortality in a US metropolitan area ($n = 60$ and $p = 15$) were studied. The response variable is `Mortality`, a total age adjusted mortality rate, and the main pollution effects of interest were that of hydrocarbons (`HC`), oxides of nitrogen (`NOx`) and sulphur dioxide (`SO2`). Several other environmental and socioeconomic considerations were taken into account, otherwise the model may include unexplained variation which may have been caused by factors other than pollution. For example, a metropolitan area with a high proportion of the elderly should expect to have a higher mortality rate than one with a lower proportion. All of the variables can be considered as continuous and real. A full description of the data can be found in Appendix **??**.

This dataset also contains several highly correlated variables. When the full model is fitted, none of the pollutant effects were found to be significant. Clearly a variable selection method was required. **McDonald1973** used ridge regression analysis to determine which variables to select. We also have results from a backwards elimination procedure (using AIC as the selection criterion) for comparison. The results are summarised in Table 8.2.

		Full model	**I-prior**	**Ridge**	**Back elim.**
Environmental & demographic variables selected		All, but only `Rain`, `JanTemp`, `NonW` significant	`Rain`, `JanTemp`, `JulTemp`, `Over65`, `Popn`, `Hous`, `NonW`, `Poor`, `Humid`	`Rain`, `JanTemp`, `Educ`, `Dens`, `NonW`	`JanTemp`, `Educ`, `NonW`
Pollution effect	`HC`	✗	✗	✗	✓ $\beta = -0.98$
	`NOx`	✗	✗	✗	✓ $\beta = 1.99$
	`SO2`	✗	✓ $\beta = 0.33$	✓ $\beta = 0.24$	✗
C_p		16.0	13.4	5.6	8.7
AIC		439.8	439.2	431.3	435.0
BIC		49.5	36.5	20.3	21.2
5f-CV RMSE		50.6	41.7	39.3	38.6

Table 8.2: The results of the various variable selection methods compared. For each method, the variable selection procedure was conducted on the set of all variables, and then an OLS was fit on the resulting selected variables. The environmental and demographic variables selected are shown in the table for each model, but those in gray are the ones found to be not significant (at the 10% level).

It is noted that the I-prior method selected some variables which turned out to be insignificant, with only three significant variables selected in total. However, of importance is learning which of the three pollution factors has an effect on mortality rate. It is nice to see that the I-prior agrees with the ridge analysis done by **McDonald1973** on this, with only sulphur dioxide having a significant effect. The strength of this effect is also comparable (I-prior 0.33 c.f. ridge 0.24). The method of backwards elimination was found not only inconsistent with I-prior and ridge analysis, but also erroneous in that it seems to imply an increase in levels of hydrocarbons would bring about a reduction in mortality rate. Once again, we see that the I-prior is outperformed in terms of Mallow's C_p, AIC, BIC and 5-fold CV RMSE, but it is noted that the 5-fold CV RMSE is not too far off from its competitors.

Ozone data

In this section, we replicate the Bayesian variable selection analysis of the Ozone dataset done by **Casella2006** which appeared initially in **Breiman1985** and also show how Bayesian variable selection can help select important interaction terms. The data consists of daily ozone readings and various meteorological quantities, and the aim was to see which of these quantities contributed to the ozone concentration. The variables are explained in Appendix **??**.

The data contains 366 points, one for each day of the leap year 1976. For our analysis, we ignore the 163 missing data in the set, and use the remaining 203 datapoint for our analysis. Out of these 203, we randomly set aside 25 to use for validation. So the n used for the Bayesian variable selection methods was $n = 178$. **Casella2006** removed the variables `TempElMon` and `ibtLAX` before running their selection model, citing multicollinearity. We won't do this, as we would like to see how well I-priors do in the presence of multicollinearity. On another note, the variables `Month`, `DayMonth` and `DayWeek` were presumably intended to be categorical as in modelling seasonality in a time series data, but these were treated as continuous, as did **Casella2006** This is just as well, as our I-prior model is not able to handle categorical variables which have more than two levels. The results are compared below:

Method	Model	Post. prob.	R^2	RMSE
I-prior	`Month HumLAX TempElMon`	0.544	0.72	3.86
CM (MPM)	`HumLAX TempSand ibhLAX`	<0.001	0.69	4.47
CM (MSE)	`Month HumLAX TempSand ibhLAX`	<0.001	0.70	4.04
BF	`TempSand ibhLAX PresGrad VisLAX`	<0.001	0.66	4.27

Table 8.3: Table showing the comparison between the I-prior, Casella and Moreno (CM) analysis, and the ACE method by Breiman and Friedman (BF). MPM stands for maximum probability model, and MSE is the lowest RMSE model.

The maximum probability I-prior model was found to be much better in terms of RMSE compared to the maximum probability model of **Casella2006** The variables selected using I-prior corresponded to the significant variables when the full OLS model was fitted. Our I-prior selected model also had a lower RMSE than **Casella2006**'s lowest RMSE model. The ACE method by **Breiman1985** was found to be the worst model for prediction. It is noted that neither **Casella2006**'s nor **Breiman1985**'s model were found in the posterior model space using the I-prior method. Out of interest, if we had removed the two variables `TempElMon` and `ibtLAX` at the beginning, then we arrive at the same results as **Casella2006**

We now use the I-prior method to select between the squared terms and all level two interactions in an effort to improve model prediction. For 12 such variables, the number of variables to select becomes $12 + 12 + 12(12-1)/2 = 90$. By doing so, we were able to improve the model to give a slightly better predictive ability. The results are shown below in Table 8.4. The maximum probability model for I-prior method selected fewer variables as compared to **Casella2006**'s maximum probability model, yet was superior in terms of RMSE. For comparison, the backwards elimination resulted in a very complicated model which did not seem to improve on RMSE.

Remark 9. As the model fit involved randomly leaving out 25 data points which were later used for validation, the results between our analysis and **Casella2006**'s are bound to differ, as we did not use the same 25 data points for training and testing.

Remark 10. For this particular dataset, running the model without squared terms and linear predictors was straightforward. However, we ran into a numerical issue in the second part, whereby some entries of $\mathbf{X}^\top\mathbf{X}$ were found to be so large compared to others, that its inverse could not be computed. Thus, we standardised the $\mathbf{X}$ and $\mathbf{y}$

Method	Model	Post. prob.	R^2	RMSE
I-prior	`Month Month^2 WindLAX HumLAX TempElMon TempElMon^2 ibtLAX PresGrad^2 ibtLAX:HumLAX`	0.103	0.83	3.74
CM	`DayMonth Month^2 TempSand^2 PresGrad^2 Month:WindLAX DayMonth:HumLAX DayWeek:TempSand PresVand:HumLAX HumLAX:ibhLAX HumLAX:VisLAX`	<0.001	0.76	3.88
Back. elim.	`Month Month^2 DayWeek PresVand HumLAX TempSand TempSand^2 TempElMon ibhLAX VisLAX PresVand^2 WindLAX^2 PresGrad^2 DayMonth:Month WindLAX:Month HumLAX:Month ibhLAX:Month ibtLAX:Month HumLAX:DayMonth TempSand:DayMonth TempElMon:DayMonth VisLAX:DayMonth WindLAX:PresVand HumLAX:PresVand TempSand:PresVand TempSand:WindLAX ibtLAX:WindLAX TempSand:HumLAX TempElMon:HumLAX VisLAX:HumLAX PresGrad:TempSand VisLAX:TempSand PresGrad:TempElMon VisLAX:TempElMon ibtLAX:PresGrad VisLAX:PresGrad`	<0.001	0.87	4.29

Table 8.4: Results of variable selection for to look for squared and interaction terms in the ozone dataset.

variables and used a single scale parameter λ. See the second last paragraph of Section 8.1.2 on page 94.

Remark 11. The method that we employed was a naive I-prior variable selection method, whereby each of the 90 terms was considered independently. If one wishes a model such that its level one term is included when an interaction is present, then the variable selection needs to be adjusted accordingly. **Kuo1998** gives an example of this:

$$y_i = \alpha + \max(\gamma_1, \gamma_3)\beta_1 x_{i1} + \max(\gamma_2,$$

8.2 Special case: The canonical RKHS

8.3 Bayesian model selection

8.4 BVS using I-priors

8.5 Simulation study

8.6 Real-data examples

Chapter 9

Conclusion

Bibliography

Bergsma, W. (2017). “Regression with I-priors”. In: *Unpublished manuscript.*

Appendix A

Appendix for I-probit

A.1 Proof of Lemma 6.1

Proof. (i) Due to the independence structure in the pdf of $\mathbf{X}$, it is easy to consider the expectations of each of the components separately and marginalising out the rest of the components. For $i \neq j$, we have

$$\begin{aligned}
\mathrm{E}[x_i] &= C^{-1} \int \cdots \int \mathbb{1}[x_k < x_j, \forall k \neq j] \cdot x_i \prod_{k=1}^{d} \frac{1}{\sigma_k} \phi\left(\frac{x_k - \mu_k}{\sigma_k}\right) \mathrm{d}x_1 \cdots \mathrm{d}x_d \\
&= C^{-1} \iint \mathbb{1}[x_i < x_j] \frac{x_i}{\sigma_i} \phi\left(\frac{x_i - \mu_i}{\sigma_i}\right) \prod_{k \neq i,j} \Phi\left(\frac{x_j - \mu_k}{\sigma_k}\right) \frac{1}{\sigma_j} \phi\left(\frac{x_j - \mu_j}{\sigma_j}\right) \mathrm{d}x_i \, \mathrm{d}x_j \\
&= C^{-1} \iint \mathbb{1}[\sigma_i z_i + \mu_i < \sigma_j z_j + \mu_j](\sigma_i z_i + \mu_i)\phi(z_i) \prod_{k \neq i,j} \Phi\left(\frac{\sigma_j z_j + \mu_j - \mu_k}{\sigma_k}\right) \phi(z_j) \, \mathrm{d}z_i \, \mathrm{d}z_j \\
&= \mu_i C^{-1} \iint \mathbb{1}[z_i < (\sigma_j z_j + \mu_j - \mu_i)/\sigma_i] \phi(z_i) \prod_{k \neq i,j} \Phi\left(\frac{\sigma_j z_j + \mu_j - \mu_k}{\sigma_k}\right) \phi(z_j) \, \mathrm{d}z_i \, \mathrm{d}z_j \\
&\quad + \sigma_i C^{-1} \iint \mathbb{1}[z_i < (\sigma_j z_j + \mu_j - \mu_i)/\sigma_i] z_i \phi(z_i) \prod_{k \neq i,j} \Phi\left(\frac{\sigma_j z_j + \mu_j - \mu_k}{\sigma_k}\right) \phi(z_j) \, \mathrm{d}z_i \, \mathrm{d}z_j \\
&= \mu_i C^{-1} \overbrace{\int \prod_{k \neq j} \Phi\left(\frac{\sigma_j z_j + \mu_j - \mu_k}{\sigma_k}\right) \phi(z_j) \, \mathrm{d}z_j}^{C} \\
&\quad + \sigma_i C^{-1} \int \mathbb{1}[z_i < (\sigma_j z_j + \mu_j - \mu_i)/\sigma_i] z_i \phi(z_i) \prod_{k \neq i,j} \Phi\left(\frac{\sigma_j z_j + \mu_j - \mu_k}{\sigma_k}\right) \phi(z_j) \, \mathrm{d}z_i \, \mathrm{d}z_j
\end{aligned}$$

The integral involving z_i in the second part of the sum is recognised as the (unnormalised) expectation of the lower-tail of a univariate standard normal distribution truncated at $\tau_{ij} = (\sigma_j z_j + \mu_j - \mu_i)/\sigma_i$. That is,

$$\operatorname{E}[Z_i | Z_i < \tau_{ij}] = \big[\Phi(\tau_{ij})\big]^{-1} \int \mathbb{1}[z_i < \tau_{ij}] z_i \phi(z_i) \, \mathrm{d}z_i = -\frac{\phi(\tau_{ij})}{\Phi(\tau_{ij})}$$

Plugging this expression back into the derivation of this expectation, we get

$$\begin{aligned}
\operatorname{E}[X_i] &= \mu_i - \sigma_i C^{-1} \int \phi\left(\frac{\sigma_j z_j + \mu_j - \mu_i}{\sigma_i}\right) \prod_{k \neq i,j} \Phi\left(\frac{\sigma_j z_j + \mu_j - \mu_k}{\sigma_k}\right) \phi(z_j) \, \mathrm{d}z_j \\
&= \mu_i - \sigma_i C^{-1} \operatorname{E}\left[\phi\left(\frac{\sigma_j Z_j + \mu_j - \mu_i}{\sigma_i}\right) \prod_{k \neq i,j} \Phi\left(\frac{\sigma_j Z_j + \mu_j - \mu_k}{\sigma_k}\right)\right].
\end{aligned}$$

The expectation for the jth component is

$$\begin{aligned}
\operatorname{E}[X_j] &= C^{-1} \int \cdots \int \mathbb{1}[x_k < x_j, \forall k \neq j] \cdot x_j \prod_{k=1}^{d} \frac{1}{\sigma_k} \phi\left(\frac{x_k - \mu_k}{\sigma_k}\right) \mathrm{d}x_1 \cdots \mathrm{d}x_d \\
&= C^{-1} \int x_j \prod_{k \neq j} \Phi\left(\frac{x_j - \mu_k}{\sigma_k}\right) \cdot \frac{1}{\sigma_j} \phi\left(\frac{x_j - \mu_j}{\sigma_j}\right) \mathrm{d}x_j \\
&= C^{-1} \int (\sigma_j z_j + \mu_j) \prod_{k \neq j} \Phi\left(\frac{\sigma_j z_j + \mu_j - \mu_k}{\sigma_k}\right) \cdot \phi(z_j) \, \mathrm{d}z_j \\
&= \mu_j C^{-1} \overbrace{\int \prod_{k \neq j} \Phi\left(\frac{\sigma_j z_j + \mu_j - \mu_k}{\sigma_k}\right) \cdot \phi(z_j) \, \mathrm{d}z_j}^{C} \\
&\quad + \sigma_j C^{-1} \int \prod_{k \neq j} \Phi\left(\frac{\sigma_j z_j + \mu_j - \mu_k}{\sigma_k}\right) \cdot z_j \phi(z_j) \, \mathrm{d}z_j \\
&= \mu_j + \sigma_j C^{-1} \operatorname{E}\left[Z_j \prod_{k \neq j} \Phi\left(\frac{\sigma_j Z_j + \mu_j - \mu_k}{\sigma_k}\right)\right] \\
&= \mu_j + \sigma_j \sum_{\substack{i=1 \\ i \neq j}}^{d} \sigma_i C^{-1} \operatorname{E}\left[\phi\left(\frac{\sigma_j Z_j + \mu_j - \mu_i}{\sigma_i}\right) \prod_{\substack{k=1 \\ k \neq i,j}}^{d} \Phi\left(\frac{\sigma_j Z_j + \mu_j - \mu_k}{\sigma_k}\right)\right] \\
&= \mu_j - \sigma_j \sum_{i \neq j} \big(\operatorname{E}[X_i] - \mu_i\big)
\end{aligned}$$

where we have made use of Lemma A.1 in the second last step of the above.

(ii) The differential entropy is given by

$$\begin{aligned}
\mathcal{H}(p) &= -\int p(\mathbf{x}) \log p(\mathbf{x}) \, \mathrm{d}\mathbf{x} = -\operatorname{E}\left[\log p(\mathbf{x})\right] \\
&= -\operatorname{E}\left[-\log C - \frac{d}{2}\log 2\pi - \frac{1}{2}\sum_{i=1}^{d} \log \sigma_i^2 - \frac{1}{2}\sum_{i=1}^{d} \left(\frac{x_i - \mu_i}{\sigma_i}\right)^2 \right] \\
&= \log C + \frac{d}{2}\log 2\pi + \frac{1}{2}\sum_{i=1}^{d} \log \sigma_i^2 + \frac{1}{2}\sum_{i=1}^{d} \frac{1}{\sigma_i^2} \operatorname{E}[x_i - \mu_i]^2.
\end{aligned}$$

□

Lemma A.1. *Let $Z \sim \mathrm{N}(0,1)$. Then for all $m \in \{\mathbb{N} \,|\, m > 1\}$ and $(\mu, \sigma) \in \mathbb{R} \times \mathbb{R}^+$,*

$$\operatorname{E}\left[Z \prod_{\substack{k=1\\k\neq j}}^{m} \Phi(\sigma_k Z + \mu_k) \right] = \sum_{\substack{i=1\\i\neq j}}^{m} \operatorname{E}\left[\sigma_i \phi(\sigma_i Z + \mu_i) \prod_{\substack{k=1\\k\neq i,j}}^{m} \Phi(\sigma_k Z + \mu_k) \right]$$

for some $j \in \{1, \ldots, m\}$.

Proof. Use the fact that for any differentiable function g, $\operatorname{E}[Zg(Z)] = \operatorname{E}[g'(Z)]$, and apply the result with the function $g_m : z \mapsto \prod_{k\neq j} \Phi(\sigma_k z + \mu_k)$. All that is left is to derive the derivative of g, and we use an inductive proof to do this.

We adopt the following notation for convenience:

$$\begin{aligned}
\phi_i &= \phi(\sigma_i z + \mu_i) \\
\Phi_i &= \Phi(\sigma_i z + \mu_i)
\end{aligned}$$

The simplest case is when $m = 2$, which can be trivially shown to be true. Without loss of generality, let $j = 1$. Then

$$\begin{aligned}
g_2(z) &= \Phi_2 \\
\Rightarrow g_2'(z) &= \sigma_2 \phi_2 = \sum_{\substack{i=1\\i\neq 1}}^{2} \left[\sigma_i \phi_i \sum_{\substack{k=1\\k\neq 1,2}}^{2} \Phi_k \right].
\end{aligned}$$

Now assume that the inductive hypothesis holds for some $m \in \{\mathbb{N} \,|\, m > 1\}$. That is, the derivative of

$$g_m(z) = \prod_{\substack{k=1\\k\neq j}}^{m} \Phi_k$$

which is

$$g'_m(z) = \sum_{\substack{i=1\\i\neq j}}^{m} \left[\sigma_i\phi_i \prod_{\substack{k=1\\k\neq i,j}}^{m} \Phi_k\right],$$

is assumed to be true. Assume that without loss of generality, $j \neq m+1$. Then the derivative of

$$g_{m+1}(z) = \prod_{\substack{k=1\\k\neq j}}^{m+1} \Phi_k = g_m(z)\Phi_{m+1}$$

is found to be

$$\begin{aligned}
g'_{m+1}(z) &= \sigma_{m+1}\phi_{m+1}g_m(z) + g'_m(z)\Phi_{m+1}\\
&= \sigma_{m+1}\phi_{m+1}\prod_{\substack{k=1\\k\neq j}}^{m}\Phi_k + \sum_{\substack{i=1\\i\neq j}}^{m}\left[\sigma_i\phi_i\prod_{\substack{k=1\\k\neq i,j}}^{m}\Phi_k\right]\Phi_{m+1}\\
&= \sigma_{m+1}\phi_{m+1}\prod_{\substack{k=1\\k\neq j,m+1}}^{m+1}\Phi_k + \sum_{\substack{i=1\\i\neq j}}^{m}\left[\sigma_i\phi_i\prod_{\substack{k=1\\k\neq i,j}}^{m+1}\Phi_k\right]\\
&= \sum_{\substack{i=1\\i\neq j}}^{m+1}\left[\sigma_i\phi_i\prod_{\substack{k=1\\k\neq i,j}}^{m+1}\Phi_k\right]\\
&= g'_{m+1}(z).
\end{aligned}$$

Thus, by induction and linearity of expectations, the proof is complete. □

A.2 Proof for ...

Lemma A.2. *Let $p(x)$ be the pdf of a random variable x. Then if*

(i) p is a univariate normal distribution with mean μ and variance σ^2,

$$\mathcal{H}(p) = \frac{1}{2}(1+\log 2\pi) + \frac{1}{2}\log\sigma^2$$

(ii) p is a d-dimensional normal distribution with mean μ and variance Σ,

$$\mathcal{H}(p) = \frac{d}{2}(1+\log 2\pi) + \frac{1}{2}\log|\Sigma|$$

(iii) p is distribution of the ***upper-tail*** *of a univariate, one-sided normal distribution*

truncated at zero with mean μ and variance 1,

$$\mathcal{H}(p) = \frac{1}{2}\log 2\pi + \frac{1}{2}\left(\mathrm{E}[x^2] + \mu^2 - 2\mu\,\mathrm{E}[x]\right) + \log\Phi(\mu)$$

*(iv) p is distribution of the **lower-tail** of a univariate, one-sided normal distribution truncated at zero with mean μ and variance 1,*

$$\mathcal{H}(p) = \frac{1}{2}\log 2\pi + \frac{1}{2}\left(\mathrm{E}\,x^2 + \mu^2 - 2\mu\,\mathrm{E}\,x\right) + \log\left(1 - \Phi(\mu)\right)$$

Proof.

Case (i): $-\log p(x) = \frac{1}{2}\log 2\pi + \frac{1}{2}\log\sigma^2 + \frac{1}{2}(x-\mu)^2$. Then

$$\begin{aligned}\mathcal{H}(p) &= \mathrm{E}_x\left[\frac{1}{2}\log 2\pi + \frac{1}{2}\log\sigma^2 + \frac{1}{2\sigma^2}(x-\mu)^2\right]\\ &= \frac{1}{2}\log 2\pi + \frac{1}{2}\log\sigma^2 + \frac{1}{2\sigma^2}\cancelto{\sigma^2}{\mathrm{E}(x-\mu)^2}\\ &= \frac{1}{2}(1+\log 2\pi) + \frac{1}{2}\log\sigma^2\end{aligned}$$

Case (ii): $-\log p(x) = \frac{d}{2}\log 2\pi + \frac{1}{2}\log|\Sigma| + \frac{1}{2}(x-\mu)^\top\Sigma^{-1}(x-\mu)$. Then

$$\begin{aligned}\mathcal{H}(p) &= \frac{d}{2}\log 2\pi + \frac{1}{2}\log|\Sigma| + \frac{1}{2}\mathrm{E}_x\left[(x-\mu)^\top\Sigma^{-1}(x-\mu)\right]\\ &= \frac{d}{2}\log 2\pi + \frac{1}{2}\log|\Sigma| + \frac{1}{2}\mathrm{tr}\left(\Sigma^{-1}\,\mathrm{E}_x\cancelto{\Sigma}{\left[(x-\mu)(x-\mu)^\top\right]}\right)\\ &= \frac{d}{2}(1+\log 2\pi) + \frac{1}{2}\log|\Sigma|\end{aligned}$$

For the next two cases, we state the following properties of a truncated normal distribution without proof.

Lemma A.3. *Let $x \sim \mathrm{N}(\mu, \sigma^2)$ with x lying in the interval (a, b). Then we say that x follows a truncated normal distribution, and*

(i) the mean of x (conditional on $a < x < b$) is

$$\mathrm{E}[x] = \mu + \sigma\frac{\phi(\alpha) - \phi(\beta)}{Z},$$

(ii) the variance of x (conditional on $a < x < b$) is

$$\mathrm{Var}[x] = \sigma^2 \left[1 + \frac{\alpha\phi(\alpha) - \beta\phi(\beta)}{Z} - \left(\frac{\phi(\alpha) - \phi(\beta)}{Z} \right)^2 \right], \textit{ and}$$

(iii) the entropy of the pdf of x (conditional on $a < x < b$) is

$$\mathcal{H} = \frac{1}{2}\log 2\pi + \frac{1}{2}\log\sigma^2 + \log Z + \frac{1}{2} + \frac{\alpha\phi(\alpha) - \beta\phi(\beta)}{2Z},$$

where $\alpha = (a-\mu)/\sigma$, $\beta = (b-\mu)/\sigma$, and $Z = \Phi(\beta) - \Phi(\alpha)$, and ϕ and Φ are the pdf and cdf of a standard normal distribution respectively.

In the special case when $\sigma = 1$ (the case we are interested in), then with some manipulation, one arrives at the following expression for the entropy of the pdf p of a truncated normal distribution:

$$\begin{aligned}
\mathcal{H}(p) &= \frac{1}{2}\log 2\pi + \cancel{\frac{1}{2}\log\sigma^2} + \log Z + \frac{1}{2}\left(1 + \frac{\alpha\phi(\alpha) - \beta\phi(\beta)}{Z}\right) \\
&= \frac{1}{2}\log 2\pi + \log Z + \frac{1}{2}\left(\mathrm{Var}[x] + \left(\frac{\phi(\alpha) - \phi(\beta)}{Z}\right)^2\right) \\
&= \frac{1}{2}\log 2\pi + \log Z + \frac{1}{2}\left(\mathrm{E}[x^2] - \mathrm{E}^2[x] + (\mathrm{E}[x] - \mu)^2\right) \\
&= \frac{1}{2}\log 2\pi + \log Z + \frac{1}{2}\left(\mathrm{E}[x^2] + \mu^2 - 2\mu\,\mathrm{E}[x]\right)
\end{aligned}$$

We now continue with the proof.

Case (iii): Using Lemma A.3 with $a = 0$, $b = +\infty$, and $\sigma = 1$, we get that $Z = 1 - \Phi(-\mu) = \Phi(\mu)$. Therefore, the entropy of p is given by

$$\mathcal{H}(p) = \frac{1}{2}\log 2\pi + \frac{1}{2}\left(\mathrm{E}[x^2] + \mu^2 - 2\mu\,\mathrm{E}[x]\right) + \log\Phi(\mu)$$

Case (iv): Again, using Lemma A.3 with $a = -\infty$, $b = 0$, and $\sigma = 1$, we get that $Z = \Phi(-\mu) = 1 - \Phi(\mu)$. Therefore, the entropy of p is given by

$$\mathcal{H}(p) = \frac{1}{2}\log 2\pi + \frac{1}{2}\left(\mathrm{E}[x^2] + \mu^2 - 2\mu\,\mathrm{E}[x]\right) + \log\left(1 - \Phi(\mu)\right)$$

□

A.3 Distribution of $\tilde{q}(\mathbf{y}^*)$ for binary case

Case: $y_i = 1$

$$\begin{aligned}
\log \tilde{q}(y_i^*) &= \mathbb{1}[y_i^* \geq 0] \cdot \operatorname{E}_{\mathbf{w},\alpha,\lambda}\left[-\frac{1}{2}(y_i^* - \alpha - \lambda \mathbf{H}_i \mathbf{w})^2 \right] + \text{const.} \\
&= \mathbb{1}[y_i^* \geq 0] \cdot \left[-\frac{1}{2}\Big(y_i^{*2} - 2\operatorname{E}_{\mathbf{w},\alpha,\lambda}[\alpha + \lambda \mathbf{H}_i \mathbf{w}] y_i \Big) \right] + \text{const.} \\
&= \mathbb{1}[y_i^* \geq 0] \left[-\frac{1}{2}(y_i^* - \tilde{\eta}_i)^2 \right] + \text{const.} \\
&\equiv \begin{cases} \mathrm{N}(\tilde{\eta}_i, 1) & \text{if } y_i^* \geq 0 \\ 0 & \text{if } y_i^* < 0 \end{cases}
\end{aligned}$$

where

$$\tilde{\eta}_i = \operatorname{E}\alpha + \operatorname{E}\lambda \mathbf{H}_i \operatorname{E}\mathbf{w}$$

by independence of $q(\mathbf{w})$, $q(\alpha)$ and $q(\lambda)$. $\tilde{q}(y_i^*)$ is recognised as being the upper-tail of a one-sided normal distribution truncated at zero. The mean is

$$\operatorname{E}[y_i^* | y_i^* \geq 0] = \tilde{\eta}_i + \frac{\phi(\tilde{\eta}_i)}{\Phi(\tilde{\eta}_i)}$$

where ϕ and Φ are, respectively, the pdf and cdf of a standard normal distribution.

Case: $y_i = 0$

Following the same argument, we can deduce that $q(y_i^*)$ in this case would be the lower-tail of a one-sided normal distribution truncated at zero. The mean is

$$\operatorname{E}[y_i^* | y_i^* < 0] = \tilde{\eta}_i + \frac{\phi(\tilde{\eta}_i)}{\Phi(\tilde{\eta}_i) - 1}.$$

Index