Kernel Methods for Likelihood-Free Inference

Amrit Seshadri Diggavi

4th Year Project Report
Computer Science
School of Informatics
University of Edinburgh
2018

Abstract

In the statistical inference of a model's parameters, the likelihood function of observed data under a particular selection of model parameters is of great importance. However, for complex models, an analytical formula for the likelihood function might not be available and it's numerical evaluation might be intractable. Likelihood free inference methods are applicable when it is possible to sample data from the model for different parameter settings. We can consider using likelihood free inference methods to approximate the posterior pdf of a model's parameters and bypass the need for an analytical solution for the likelihood function.

The Likelihood free inference by ratio estimation (LFIRE) technique is a recent likelihood free inference technique that may be considered for estimation of the posterior of a model's parameters. For a data generating model, the LFIRE technique approximates the posterior pdf of a model's parameters by approximating the ratio between the data generating pdf and marginal pdf, by training logistic regression to classify data sets generated from the model under different parameter settings.

The training of a logistic regression classifier for the estimation of the ratio in the LFIRE technique can be replaced with training a Kernel Logistic Regression (KLR) classifier, so that we can powerfully and compactly consider likelihood free inference of a model's parameters with complex high dimensional summary statistics for generated data.

This thesis considers Gaussian and ARCH(1) data generating models for posterior estimation and applies KLR to the LFIRE technique, investigating how the cost of doing so can be minimized and how an appropriate choice of kernel function and associated hyper-parameters can be made for a given data generating model.

It found that for an appropriate choice of kernel function, we are able to guide an exact selection of kernel hyper-parameters that are appropriate for posterior estimation by maximizing the prediction accuracy of classification tasks defined by the LFIRE technique.

Further, it is found that for an appropriate kernel function, we can typically avoid an extensive grid search for optimal parameters that maximize prediction accuracy for tasks defined by LFIRE, and usefully reduce the costs of using KLR for LFIRE since we would try to maximize prediction accuracy on these tasks, to guide an appropriate selection of kernel hyper parameters for posterior estimation.

Acknowledgements

I thank my project supervisor, Dr. Michael U. Gutmann for his invaluable support and guidance over the course of this project.

Table of Contents

Chapter 1

Introduction

In the statistical inference of a model's parameters, the *likelihood function* (that is, the probability density function (pdf) of observed data under a particular selection of model parameters) is of great importance. It provides a measure of the viability of observed data under the parameter selection. More compactly, with Bayes Rule - given a selection of model parameters θ and observed data x for a particular model:

$$\underbrace{p(\theta|x)}_{\text{Posterior}} \propto \underbrace{p(x|\theta)}_{\text{Likelihood}} \underbrace{p(\theta)}_{\text{Prior}} \tag{1.1}$$

However, for complex models, an analyticaly formula for the likelihood function might not be available and it's numerical computation might be intractable: Very costly and possibly even infeasible. As a solution to this problem, we may consider likelihood free inference methods to approximate the posterior pdf.

Likelihood free inference methods are applicable when it is possible to sample data from the model for different parameter settings θ. That is, when the model M is a data generating simulator, from which we may generate data x_θ under different model parameter settings θ.

$$x_\theta \sim M(\theta) \tag{1.2}$$

Such methods typically approximate the posterior pdf of model parameters by approximating the likelihood function. To do this, these methods rely on running simulations of the data generating model for different settings of model parameters and bypass the need for an analytical formula of the likelihood function.

Likelihood free inference methods allow statistical inference to be considered for complex models that have no analytical likelihood formula available and are popular in domains such as systems biology (Toni et al. (2009)) and population genetics (Beaumont et al. (2002)) for the inference of parameters of models capturing complex naturally occurring processes, where obtaining an analytical solution for the likelihood function is typically not possible.

Likelihood free inference techniques typically operate upon generated data points as vectors of summary statistics and rely heavily on this choice of summary statistics.

Approximate Bayesian Computation (ABC) is a popular class of *Likelihood free* inference methods that rely on computing the distance between summary statistics for simulated data. Sunnåker et al. (2013) provides a review of ABC methods.

ABC methods typically require both a careful selection of summary statistics and a related measure of *closeness* to be effectively employed. Furthermore, the probability of generating data points that are *close* together from a simulator decreases as the dimensionality of summary statistics increases. Consequently, *ABC* tends to fail for high dimensional summary statistics. This restriction to low dimensional summary statistics in *ABC* demands a very careful selection of summary statistics and for most complex models introduces a bias in the inference of model parameters as some information of the data is typically lost.

The *Synthetic Likelihood (SL)* technique (Wood (2010)) is another popular *likelihood free* inference method that requires no measure of *closeness* and does not require a very careful selection of summary statistics. This approach assumes that the summary statistics of sampled data follows a Gaussian distribution and approximates the likelihood function to this Gaussian distribution.

The *SL* technique however still requires a fixed selection of summary statistics for generated data points that requires expert knowledge of the problem domain and the assumption of a Gaussian distribution over these summary statistics is restrictive.

The *Likelihood Free Inference by Ratio Estimation (LFIRE)* technique (Dutta et al. (2016)) is a more recent technique for *likelihood free* inference. The *LFIRE* technique approximates the posterior pdf of model parameters by estimating the ratio $r(x,\theta)$ between the data generating pdf $p(x|\theta)$ and the marginal pdf $p(x)$.

$$\underbrace{r(x,\theta)}_{\text{Ratio}} = \frac{p(x|\theta)}{p(x)} \tag{1.3}$$

$$\underbrace{\hat{p}(\theta|x)}_{\text{Approx. posterior}} = \underbrace{\hat{r}(x,\theta)}_{\text{Approx. ratio}} \underbrace{p(\theta)}_{\text{Prior}} \tag{1.4}$$

For a setting of model parameters θ, the *LFIRE* technique estimates this ratio by training logistic regression to classify between multiple data points that have been generated from the simulator for model parameters setting θ: ($x_\theta \in X_\theta$), and data points that have been generated from the simulator for random $\bar{\theta}$ values sampled from the prior distribution $p(\theta)$: ($x_m \in X_m$).

$$X_\theta \sim M(\theta) \tag{1.5}$$

$$X_m \sim M(\bar{\theta}) \quad \text{where for each } \mathrm{x}_m \in X_m \quad \bar{\theta} \sim p(\theta) \tag{1.6}$$

The *LFIRE* technique allows for the automatic selection of relevant summary statistics from a broad set by the use of regularized logistic regression. Furthermore, this

approach makes no assumptions on the distribution of summary statistics and is more flexible for likelihood pdfs that are not naturally Gaussian.

However, to capture complex information of the data generated from a model, even selecting a broad list of summary statistics for estimation of the posterior can be difficult. Rather, this selection of summary statistics can be simplified by compactly associating the list of summary statistics for *LFIRE* with a kernel function.

A *kernel function* k defines the dot product between two data points in some feature space H without necessarily performing an explicit mapping of data points from the input space X to that feature space. The *kernel trick* avoids an explicit mapping of data points to high dimensional spaces by formulating an algorithm to access data only via the dot products between data points (that is, via the their *kernel functions*).

$$\phi : X \to H \tag{1.7}$$

$$k(x,x') = < \phi(x), \phi(x') > \tag{1.8}$$

Support vector machines (SVMs) are popularly used along with the kernel trick to classify data in very complex high dimensional spaces at feasible computational costs, without necessarily requiring a vectorial representation for the input data. Scholkopf and Smola (2001) provides a comprehensive introduction to the field.

The kernel trick is most usefully employed for the classification of data in high dimensional spaces when some information is known about the structure of the data to be classified and a particular kernel function exists to compute the dot product between data points in a high dimensional space that is meaningful for the corresponding data structure. Kernel functions have been created for sequence data, graphs, text, images, as well as vectors. (Souza) provides a description of popular kernel functions along with their typical use.

Alternative to classification with SVMs, we may consider *Kernel Logistic Regression (KLR)* that uses the *kernel trick* for the classification of data points in very complex high dimensional spaces, (see Zhu and Hastie (2002)).

For estimation of the posterior of a model's parameters with the *LFIRE* technique, we can more compactly and powerfully capture complex information of the generated data by associating the list of summary statistics considered for posterior estimation with a kernel function, so that the list of summary statistics considered for posterior estimation are the dimensions of the feature space associated with the kernel function. Doing this, we can consider for summary statistics all feature spaces for which a kernel function exists.

In particular, the training of a logistic regression classifier for the estimation of the ratio $r(x,\theta)$ in the LFIRE technique can be replaced with training a *KLR* classifier, so that we can consider *likelihood free* inference of a model's parameters with complex high dimensional summary statistics for generated data.

Typically having some knowledge of the structure of the data generated from a model, we may narrow the selection of an appropriate kernel function for the estimation of

the posterior of a model's parameters. However, the exact selection of an appropriate kernel function for posterior estimation may be difficult. Further, a kernel function is typically associated with some hyper-parameters that decide it's mapped feature space. The appropriate selection of these kernel function hyper-parameters for posterior estimation is not obvious. Additionally, KLR is a computationally expensive technique, so for practical estimation of the posterior of a model's parameters using KLR for LFIRE, we would like to limit the number of times we train the KLR classifier.

This thesis applies KLR to the LFIRE technique, investigating how the cost of doing so can be minimized and how an appropriate choice of kernel function and associated hyper-parameters can be made for a given data generating model.

The work completed for this thesis includes:

- The implementation of all supporting code for the experiments conducted: Implementation of a Kernel Logistic Regression classifier, design and implementation of code for selection of appropriate kernel hyper parameters and regularization coefficients (section 6.1), the implementation of the Gaussian and ARCH(1) data generating models considered for posterior estimation along with their true posteriors, the implementation of code supporting simultaneous estimation of the posterior pdf at multiple model parameter settings for multiple observed data points, and code for evaluation of the estimated posteriors along with visualization code. All implementation was performed in Python within the SciPy framework (Jones et al. (2001–)).
- All experiments conducted with the Gaussian and ARCH(1) models for different kernel functions in exploration of the research questions of this thesis. (Detailed in section 3), along with the interpretation of the results of experiments.

From experiments conducted, it is observed that a feature space mapping (corresponding to a choice of kernel function and kernel hyper parameter) that appropriately captures the data for estimation of the posterior also appropriately captures the data to achieve 'good' classification accuracy on the classification tasks defined by the LFIRE technique for varied model parameters. Consequently, for an appropriate choice of kernel function, it seems that we are able to guide an exact selection of kernel hyper-parameters that are appropriate for posterior estimation by maximizing the prediction accuracy of classification tasks defined by the LFIRE technique.

It is further observed that the classification tasks defined by the LFIRE technique for varied settings of model parameters are of roughly similar difficulty, and exploiting this fact, we can avoid an extensive grid search for optimal parameters that maximize prediction accuracy (using the KLR classifier), so that if we are using maximum prediction accuracy of classification tasks defined by LFIRE to guide the selection of appropriate hyper parameters for posterior estimation, we can reduce the number of times we train Kernel Logistic Regression for LFIRE.

Chapter 2 provides a brief background of the techniques used, chapter 3 details the research questions that motivate the experiments conducted, chapter 4 presents the experiments conducted along with their results and interpretations for the observed results, chapter 5 presents the conclusions of the thesis and discussion of possible

further work, chapter 6 is the appendix and discusses some procedures followed in support of the experiments conducted.

Chapter 2

Background

2.1 The LFIRE technique

The Likelihood Free Inference by Ratio Estimation (LFIRE) technique is a *likelihood free* inference technique that can be used to approximate the posterior pdf of a model's parameters θ given some observed data. The LFIRE technique requires that we are able to sample data from the pdf $p(x|\theta)$ for different model parameters $\theta \in \mathbb{R}^d$ (that is, the model considered is a data generating simulator).

Given observed data $x_0 \in X$ and a model which is a data generating simulator, the LFIRE technique approximates the posterior pdf of model parameters, $p(\theta|x_0)$ by approximating the ratio $r(x,\theta)$ between the data generating pdf $p(x|\theta)$ and the marginal pdf $p(x)$. As outlined in equations (1.3), (1.4).

It is shown in (Dutta et al. (2016)), this ratio can be estimated by training logistic regression to classify between data set classes X^θ and X^m of sizes n_θ and n_m respectively, constructed such that X^θ consists of data points sampled from $p(x|\theta)$ and X^m consists of data points sampled from $p(x)$; as outlined in equations 1.5, 1.6

Specifically, the ratio can be estimated as:

$$\hat{r}(x,\theta) = e^{\hat{a}(x,\theta)} \tag{2.1}$$

where,

$$\hat{a} = \arg\min_a \{L(a,\theta)\} \tag{2.2}$$

is the minimizing function in the logistic regression setup:

Data points are classified between class X^θ and class X^m, assuming probability:

$$P(x \in X^\theta) = \frac{1}{(1+\nu e^{-a(x)})} \tag{2.3}$$

and a decision boundary characterized by a function a is learnt by minimizing the loss function:

$$L = \frac{1}{n_m + n_\theta}\{\sum_{i=1}^{n_\theta} \log(1 + \nu e^{-a(x_i^\theta)}) + \sum_{i=1}^{n_m} \log(1 + \frac{1}{\nu} e^{a(x_i^m)})\} + Cg(a) \tag{2.4}$$

Where, $\nu = \frac{n_m}{n_\theta}$ compensates for unequal class size, g is some regularizer, and C is the regularization coefficient.

In the case of linear logistic regression, we search for an optimal funciton a that is of the form:

$$a(x) = \mathbf{w}^T \mathbf{x} + w_0 \tag{2.5}$$

Where the datapoint x is compiled as a vector of summary statistics $\mathbf{x} \in \mathbb{R}^d$.
Additionally, it was shown in (Dutta et al. (2016)) that by including a regularization penalty term g in the loss function L, and choosing a regularization coefficient so that the trained classifier achieves optimal prediction accuracy on a validation set, we can select a function of relevant complexity for the classification task (that is, we can chose $\vec{w}$ to capture relevant summary statistics). This is demonstrated in (Dutta et al. (2016)) by the use of L1 regularized linear logistic regression.

2.2 The kernel trick and kernel logistic regression

A kernel function k defines the dot product between two data points in some feature space H without necessarily performing an explicit mapping of data points from the input space X to that feature space; see (1.7), (1.8).

Linear algorithms that deal with data only via dot products, may replace dot products with kernel functions to learn decision boundaries that are non-linear in the input space X. This technique is known as the kernel trick, and is especially useful when direct computation in the desired feature space H is either computationally impractical or impossible to do, as in the case of infinite dimensional spaces. Additionally, not all kernel functions require a vector representation for the input data.

2.2.1 RKHS representations for kernel functions and the Representer Theorem

The Reproducing Kernel Hilbert Space (RKHS) representation (Hastie et al. (2001)), for a positive definite kernel function k defined on input space X maps:

$$X \to H \quad where \quad H := \{f : X \to \mathbb{R}\} \tag{2.6}$$

such that:

$$x \to k(.,x) \tag{2.7}$$

Theory on RKHS representations is very well developed and allows the study of kernel functions that are associated with very complex feature spaces. Many popular kernel functions are associated with infinite dimensional spaces. However, the direct optimization

of a loss function in an infinite dimensional space is not practical. Fortunately, as a result of the Representer Theorem (see Schölkopf et al. (2001)) even when using kernels associated with infinite dimensional spaces, the optimal solution to the optimization of a point wise loss function lies within the finite dimensional span of the training examples mapped into the feature space associated with the kernel function.

Specifically, by the Representer Theorem, for a kernel function k associated with the feature space mapping H, a point-wise loss function L defined over training examples X, target values y and a function $f \in H$, any f^* satisfying:

$$f^* = \arg\min_{f \in H} \{L(X, y, f) + \|f\|_H^2\} \tag{2.8}$$

has the form

$$f^* = \sum_{i=1}^{n} \alpha_i k(., x_i) \tag{2.9}$$

A positive definite kernel is associated with the reproducing property in RKHS. By which,

$$< f, k(., x_i) >= f(x) \tag{2.10}$$

and in particular,

$$< k(., x), k(., x^{'}) >= k(x, x^{'}) \tag{2.11}$$

Where K is the kernel gram matrix of training examples X so that, K_{ij} =$k(x_i, x_j)$, along with the reproducing property of a kernel function in RKHS, the Representer Theorem implies that:

$$< f^*, k(., x_i) >= \sum_{j=1}^{n} \alpha_j k(x_i, x_j) = w^T K_i \tag{2.12}$$

and

$$\|f^*\|_H^2 =< f^*, f^* >= w^T K w \tag{2.13}$$

for some $w \in \mathbb{R}^N$, where N is the number of training examples used.

2.2.2 Kernel Logistic Regression

A standard regularized logistic regression classifier classifies data points $x \in X$ between two classes as outlined in 2.4. L2 regularized linear logistic regression would learn a decision boundary characterized by a weight vector $w \in \mathbb{R}^d$ and bias term w_0 by using the activation function

$$a(x_i) = w_0 + < w, x_i > \tag{2.14}$$

and regularizer

$$g(a) = \|w\|^2 \tag{2.15}$$

For data points $x \in X$, we can train a kernel logistic regression classifier in the non-linear RKHS feature space H associated with a kernel function k by using the activation function

$$a(x_i) = w_0 + < f^*, k(., x_i) > \tag{2.16}$$

and regularizer

$$g(a) = \|f^*\|_H^2 \tag{2.17}$$

From 2.12, 2.13, we can equivalently train a kernel logistic regression classifier in the non-linear RKHS by using the activation function

$$a(x_i) = w_0 + w^T K_i \tag{2.18}$$

and regularizer

$$g(a) = w^T K w \tag{2.19}$$

So that practically we can train a KLR classifier by searching for a vector $w \in \mathbb{R}^N$ that minimizes the logistic regression cost function 2.4, where N is the number of training examples used.

KLR is an expensive technique. The computational cost of training KLR with N training samples is $O(N^3)$ and space cost of training KLR is $O(N^2)$.

Chapter 3

Research Questions

The *likelihood function* is of great importance in the inference of the posterior pdf of a model's parameters. However, for complex models, an analytical formula for the likelihood function might not be available and it's numerical computation might be intractable. As a solution to this problem, when the model used is a data generating model (that is, it is possible to sample data from the model for different parameter settings θ), we may consider likelihood free inference methods to approximate the posterior pdf of the model's parameters.

These methods rely on running simulations of the data generating model for different settings of model parameters and bypass the need for an analytical formula of the likelihood function. Typically such methods operate upon generated data points as vectors of summary statistics and rely heavily on this choice of summary statistics.

The *Likelihood Free Inference by Ratio Estimation (LFIRE)* technique is a relatively recent technique for *likelihood free* inference that approximates the posterior pdf of model parameters by estimating the ratio $r(x,\theta)$ between the data generating pdf $p(x|\theta)$ and the marginal pdf $p(x)$.

The *LFIRE* technique allows for the automatic selection of relevant summary statistics from a broad set by the use of regularized logistic regression, making no assumptions on the distribution of the summary statistics. However, even selecting a broad list of summary statistics that appropriately captures complex information from the generated data for estimation of the posterior can be difficult.

The training of a logistic regression classifier for the estimation of the ratio $r(x,\theta)$ in the LFIRE technique can be replaced with training a *Kernel Logistic Regression (KLR)* classifier, so that so that the list of summary statistics considered for posterior estimation are the dimensions of the feature space associated with the chosen kernel function, allowing us to compactly and powerfully capture complex information of the generated data.

Specifically, for model parameters θ and observed data x_0 we can estimate

$$\hat{p}(\theta|x_0) = p(\theta)\hat{r}(x_0,\theta) = p(\theta)e^{\hat{a}(x_0,\theta)} \tag{3.1}$$

where,

$$\hat{a} = \arg\min_{a}\{L(a,\theta)\} \tag{3.2}$$

minimzes the loss function:

$$L = \frac{1}{n^{\theta}+n^{m}}\{\sum_{i=1}^{n_{\theta}}\log(1+\nu e^{-a(x_i^{\theta})}) + \sum_{i=1}^{n_m}\log(1+\frac{1}{\nu}e^{a(x_i^{m})})\} + Cw^{T}Kw \tag{3.3}$$

where $\nu = \frac{n_m}{n_\theta}$ compensates for unequal class size, C is the regularization coefficient, $K_{ij} = k(x_i, x_j)$, for a positive definite kernel k defined on X and $a(x_i) = w_0 + w^T K_i$ for some $w \in \mathbb{R}^N$ for N training samples made up of data sets X_θ and X_m, generated as outlined in equations (1.3), (1.4).

We typically have some knowledge of the structure of the data generated from a model, that can help narrow the selection of an appropriate kernel function for the estimation of the posterior of the model's parameters. However, the exact selection of an appropriate kernel function for posterior estimation may be difficult. Further, a kernel function is typically associated with some hyper-parameters that decide it's mapped feature space and the appropriate selection of these hyper-parameters for estimation of the model's posterior is not obvious. Additionally, as KLR is a computationally expensive technique, for practical estimation of the posterior of a model's parameters using KLR for LFIRE, we would like to limit the number of times we train the KLR classifier. The research questions explored in this thesis are therefore:

- How we can guide an appropriate selection of kernel function and kernel hyper parameters for the posterior estimation of a given model's parameters using the LFIRE technique.
- How we can minimize the costs of using the expensive Kernel Logistic Regression technique in its application to the LFIRE technique.

To guide exploration, the following objectives were considered:

- Implement a Kernel Logistic Regression classifier.
- Using the implemented KLR classifier, for a Gaussian data generating model, analyze the classification tasks defined by the LFIRE technique for estimation of the posterior of the model's parameter. First using a linear kernel with a basis expansion for generated data, such that the summary statistics required for estimation of the posterior pdf correspond exactly to the dimensions of the feature space mapped by the kernel function.
- Using the LFIRE technique with KLR, approximate the posterior pdf of the Gaussian model's parameter using the linear kernel and a basis expansion such that the kernel's mapped feature space corresponds exactly to the summary statistics appropriate for posterior estimation. By doing so, validate the application of KLR to the LFIRE technique.
- Analyze the classification tasks defined by the LFIRE technique for the Gaussian model, using polynomial kernels of varying degree for KLR.

- Using the LFIRE technique with KLR, approximate the posterior pdf of the Gaussian model's parameter using polynomial kernels of varying degree. From observed results, consider how an appropriate choice of kernel function hyper parameter can be made for posterior estimation.
- Analyze the classification tasks defined by the LFIRE technique for the Gaussian model, using RBF kernels of varying kernel hyper parameter for KLR.
- Using the LFIRE technique with KLR, consider how an appropriate choice of hyper parameter can be made for the RBF kernel for posterior estimation. Approximate the posterior pdf of the Gaussian model's parameter, using this choice if possible.
- Consider a lag-one auto-regressive model with conditional heteroskedasticity (ARCH(1)) for posterior inference of model parameters. Analyze the classification tasks defined by the LFIRE technique for the ARCH(1) model, using RBF kernels of varying kernel hyper parameter for KLR.
- Using the LFIRE technique with KLR, consider how an appropriate choice of hyper parameter can be made for the RBF kernel for posterior estimation. Approximate the posterior pdf of the ARCH(1) model's parameters, using this choice if possible.
- In all experiments using KLR with LFIRE to estimate the posterior pdf of a model's parameters, attempt to minimize computational costs by reducing the number of times the KLR classifier is trained.

Chapter 4

Experiments

4.1 KLR classifier

A KLR classifier was implemented in Python, minimizing the objective function (3.3). This implementation is largely based on a Matlab implementation of kernel logistic regression (Schmidt (2008)).

Modifications include the addition of a bias term to the activation function, definition of the gradient of the objective function for the minimizer and scaling of the kernel gram matrix by its mean absolute value before computation (that is, scaling the matrix by the mean computed over absolute values of it's entries).

A cutoff of 100 iterations is used in minimizing the objective function. As the training set size used for experiments is typically 2000 samples, 100 iterations is sufficient for the minimizer to reach convergence.

The magnitudes of the regularizer and the point-wise loss of the objective function (3.3) are dependent on the magnitude of values in the kernel gram matrix. By scaling the kernel gram matrix by it's mean absolute value before computation, we are able to select optimal regularization coefficients from a common scale, invariant of the average magnitude of the kernel function used. It is further observed that optimal weight vector values may be very small when the magnitude of the kernel gram matrix values are very large and visa versa. Scaling of the kernel gram matrix avoids this and adds stability to the minimization problem, making it more robust as small changes in the weight vector will not not result in large changes in loss and visa versa.

4.2 Experiments with the Gaussian model

The data generating model considered for parameter inference is a uni-variate Gaussian distribution with standard deviation $\sigma_0 = 3$. The mean of the distribution μ is the model parameter to be inferred. The analytical formula for the true posterior of the model is known (Dutta et al. (2016)). Assuming a uniform prior on the unknown mean

$U(-20,20)$, the analytical formula for the true posterior of the mean given an observed data point x_0 is:

$$p(\mu|x_0) = \begin{cases} \exp\{\alpha_0(\mu) + \alpha_1(\mu)x_0 + \alpha_2(\mu)x_0^2\}, & \text{if } \mu \in (-20,20) \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

For coefficients:

$$\alpha_0(\mu) = \frac{-\mu^2}{2\sigma^2} - \log(\sqrt{2\pi\sigma^2})) - \log\left(\phi\left(\frac{20-x_0}{\sigma_0}\right) - \phi\left(\frac{-20-x_0}{\sigma_0}\right)\right) \tag{4.2}$$

$$\alpha_1(\mu) = \frac{\mu}{\sigma_0^2} \tag{4.3}$$

$$\alpha_2(\mu) = \frac{-1}{2\sigma_0^2} \tag{4.4}$$

Using the LFIRE technique to estimate of posterior of the model for points $\mu \in (-20,20)$, we define classification tasks:

Classifying between:

- N data points generated from the Gaussian distribution for mean μ: (X_μ)
- N data points generated from the Gaussian distribution for random $\bar{\mu}$ values sampled from the prior distribution $\bar{\mu} \sim U(-20,20)$: (X_m).

For each $\mu \in (-20,20)$ value we wish to estimate the posterior for. In the following experiments, for each classification task, we generate N =1000 data points for each set X_μ, X_m.

This data generating model was previously considered for parameter inference by (Dutta et al. (2016)).

4.2.1 Linear kernel: Classification Tasks

To estimate the posterior of the Gaussian model parameter μ using KLR, we first consider using the linear kernel function along with a basis expansion $[1,x,x^2]$ for generated data points x.

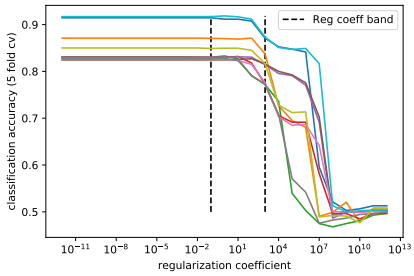
The linear kernel function is simply the dot product of data points in the input feature space.

$$k(\mathbf{x},\mathbf{x}') = <\mathbf{x},\mathbf{x}'> \tag{4.5}$$

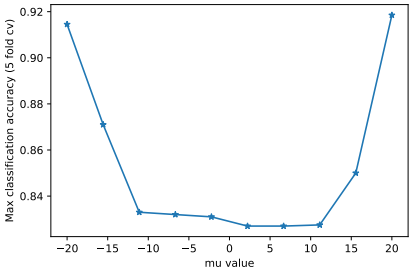
The summary statistics considered for the estimation of the posterior are the dimensions of the feature space mapped by the kernel function. That is, $[1,x,x^2]$. In particular, these summary statistics are exactly appropriate to approximate the true posterior of the model's parameter as they capture the functional form of the true posterior exactly (see (4.1)).

For 10 classification tasks defined by evenly spaced $\mu \in (-20, 20)$, generated data points were transformed to have a basis $[1, x, x^2]$ and the KLR classifier was trained, using the linear kernel function.

Optimal regularization coefficients were selected separately for each task from a scale $(10^{-12}, 10^{12})$ to maximize 5 fold cross validated prediction accuracy.



(a) Prediction accuracy vs regularization coefficient.



(b) Max prediction accuracy for optimal choice of regularization coefficient.

Figure 4.1: Plots for multiple classification tasks defined by estimating the posterior of model parameter μ for 10 points $\mu \in (-20, 20)$ using the Gaussian model, linear kernel and basis expansion $[1, x, x^2]$ for generated data. In Figure 4.1a, each curve plotted correspond to a different classification task defined by a single $\mu \in (-20, 20)$. The reg coeff band captures the optimal regularization coefficients for all classification tasks considered.

4.2.1.1 Result

It is observed (Figure 4.1a) that for all classification tasks defined by estimating the posterior for $\mu \in (-20, 20)$ regularization coefficients of similar magnitude optimize prediction accuracy.

It is also observed (Figure 4.1b) that maximized prediction accuracy (by appropriate choice of regularization coefficeint) is worse for tasks defined by estimating the posterior close to the median of the prior distribution (that is close to $\mu = 0$).

4.2.1.2 Interpretation

The selection of similar regularization coefficients to optimize prediction accuracy for all classification tasks defined by estimating the posterior for $\mu \in (-20, 20)$ indicates that these classification tasks are of roughly similar difficulty in the feature space defined by the kernel function used (namely $[1, x, x^2]$).

X_m remains unchanged across classification tasks, and similar difficulty for classification tasks indicates a similar separation of data-sets X_μ from X_m for different $\mu \in (-20, 20)$ in the feature space associated with the kernel function.

Further, worse classification accuracy for tasks defined by μ closer to $\mu = 0$ indicates that the datasets X_m and X_μ are worst separated close to $\mu = 0$.

A possible explanation for this observation is as follows: For a feature space that captures information of the generated data points meaningfully for the data generating model, we would expect that model parameters that are close to each other generate data points that are close to each other in the mapped feature space (That is, we would expect continuity for a function that maps model parameters to generated data points in the feature space). In case of the Gaussian model, we would expect that μ values that are close to each other generate similar data-points from the data generating model in a feature space that meaningfully captures information of the generated data. On average, the closest model parameter to values in the set model parameters, $\bar{\mu} \sim U(-20, 20)$ is the median of the prior distribution $U(-20, 20)$, that is $\mu = 0$. Consequently, in a meaningful feature space mapping, the data-set X_m that is generated from the set of model parameters $\bar{\mu} \sim U(-20, 20)$ would be most similar to data-set X_μ when μ is close to the median of the prior distribution, that is $\mu = 0$. If the data-sets X_m and X_μ are most similar for μ close to 0, then $\mu = 0$ defines the *hardest* classification task and for this task, we would expect the poorest classification accuracy, in a feature space that meaningfully captures information about the generated data points with respect to the data generating model.

4.2.1.3 Consequence

During training, for N training instances, the computational cost of the KLR is $O(N^3)$ and space cost of KLR is $O(N^2)$. KLR is typically too costly to be applied to very large data-sets, and when data-set size does allow the application of KLR, to reduce costs, we would prefer to train the classifier as few times as possible.

Selection of an optimal regularization coefficient for a single classification task is a computationally expensive process as it requires training the classifier once for each possible choice of coefficient. In the case of estimating posterior probability using the LFIRE technique, where we wish to train KLR for numerous classification tasks, multiple such searches for optimal regularization coefficients would be very computationally expensive.

In the case of the linear kernel using a basis expansion $[1, x, x^2]$, on account of the similar choices of regularization coefficient made, we are able to select a band of

values in the neighborhood of the maximizing regularization coefficient for a single classification task, so that for tasks defined by all other μ values $\in (-20, 20)$, we can narrow the search for optimal regularization coefficients and reduce the number of times we train KLR, significantly reducing computational costs for posterior estimation using KLR.

The process for selection of a band of regularization coefficient values in the neighborhood of the maximizing regularization coefficient for a single classification task is detailed in the Appendix chapter, section 6.1.

By considering the classification task defined by $\mu = 0$, a band of regularization coefficients was selected for all classification tasks. Shown in Figure 4.1a

4.2.2 Linear kernel: Posterior estimation

As mentioned earlier, using a linear kernel with basis expansion $[1, x, x^2]$ for generated data, it should be possible to capture the analytical function for the true posterior exactly. We would therefore expect a good estimation of the posterior by using the linear kernel function along with a basis expansion $[1, x, x^2]$ for generated data points x and the success of these experiments in estimating the posterior of model parameter μ serves to validate the use of KLR for the LFIRE technique.

Using the chosen common narrow regularization band for all tasks (selected for classification task $\mu = 0$ as described in 6.1) and choosing a regularization coefficient to maximize prediction accuracy on a common generated validation set for each task, the posterior was estimated for the Gaussian model parameter μ for observed data $x_0 = 0$ at 10 points $\mu \in (-6, 6)$, as the majority of the probability mass lies within this region. Each classification task was run 10 times with different random seeds (so that data sets generated from the model simulator were different each run). Percentiles for the estimated posteriors at each point $\mu \in (-6, 6)$ are plotted in Figure 4.2 along with the true posterior value. Estimated posterior values approximately match the values of the true posterior.

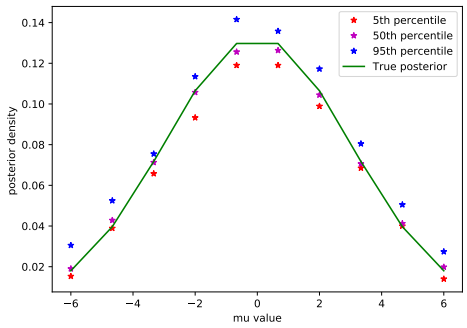


Figure 4.2: Multiple experiments run for estimation of the posterior probability density of model parameter μ at 10 points $\mu \in (-6, 6)$ for observed data $x_0 = 0$, using the Gaussian model with a linear kernel and basis expansion $[1, x, x^2]$ for generated data.

4.2.3 Polynomial kernel: Classification Tasks

The polynomial kernel is commonly used in SVMs and has associated feature space mappings that correspond to logical conjunctions of the input features.

The polynomial kernel function of degree d has the formula:

$$k_d(\mathbf{x}, \mathbf{x}') = (a < \mathbf{x}, \mathbf{x}' > +c)^d \tag{4.6}$$

For adjustable slope a and constant term c.

For the Gaussian model, where generated data points are one dimensional, polynomial kernel functions of degree d with slope $a = 1$ and constant term $c = 1$ are considered, so that the feature space mapping associated with a polynomial kernel function of degree d is:

$$\phi_d(x) = [x^d, \alpha_{d-1}x^{d-1}, ..., \alpha_1 x, 1] \tag{4.7}$$

For some coefficients α_i.

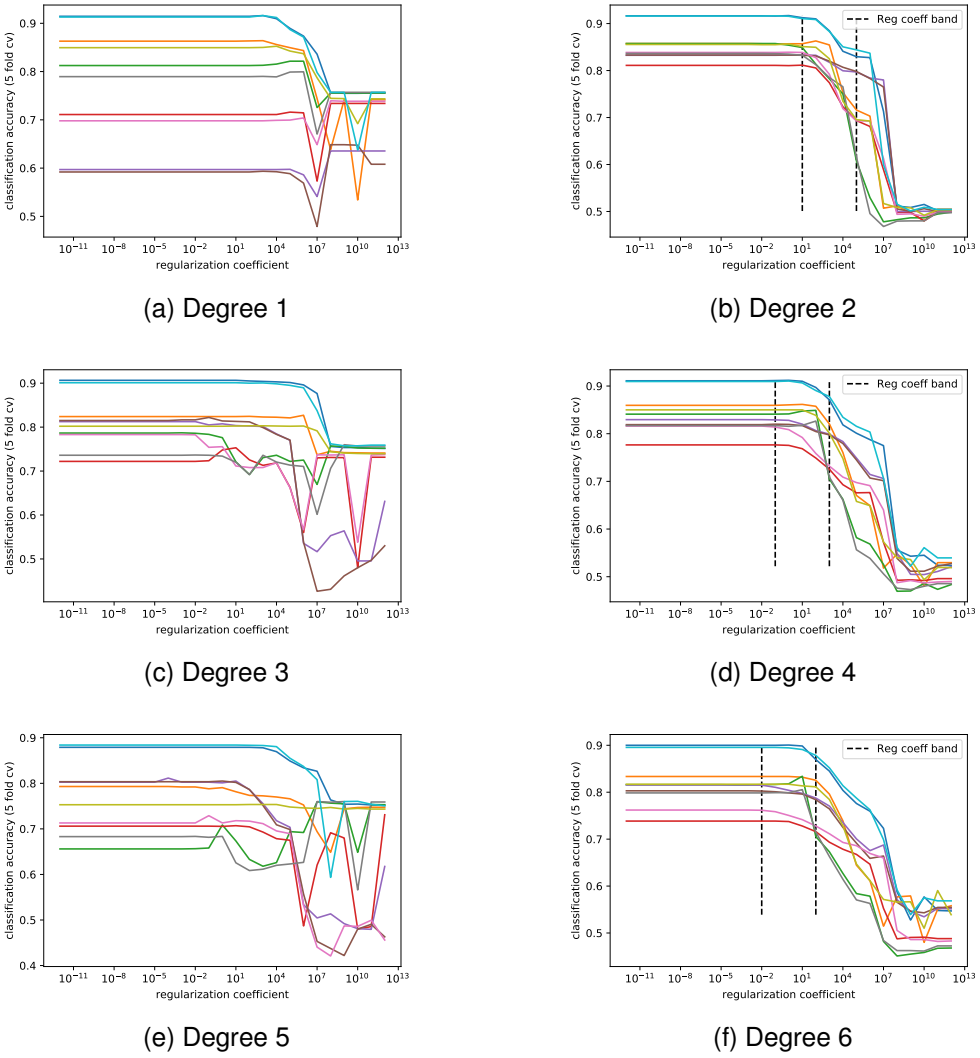
Once again, classification tasks defined by the estimation of the posterior of the Gaussian model at 10 evenly spaced points $\mu \in (-20, 20)$ were considered. The KLR classifier was trained using each polynomial kernel function of degree 1-6 on common generated data-sets.

Optimal regularization coefficients were selected separately for each task, for each choice of degree, from a scale $(10^{-12}, 10^{12})$ to maximize 5 fold cross validated prediction accuracy.

4.2.3.1 Result

Figure 4.3 shows the variation of cross validated classification accuracy for the multiple classification tasks defined by $\mu \in (-20, 20)$. Interestingly, for polynomials kernels of even degree, there is again a similar choice of regularization coefficients that maximize prediction accuracy, so that it is again possible to make a single choice of a narrow regularization coefficient band based on a single task, to allow cheaper costs for posterior estimation. Optimal choices of regularization coefficients for polynomial kernels of odd degree are far more erratic.

For the multiple classification tasks defined by $\mu \in (-20, 20)$, Figure 4.4a shows the maximum achievable prediction accuracies by making optimal choices of regularization coefficient (maximizing 5-fold cross validated accuracies on common data-sets) for polynomial kernel functions of degrees 1-6. Figure 4.4b shows the averaged maximum achievable prediction accuracies by making optimal choices of regularization coefficient for the 10 classification tasks defined by $\mu \in (-20, 20)$. It is observed that the polynomial kernel of degree 2 typically achieves best classification accuracy for all classification tasks.



(a) Degree 1

(b) Degree 2

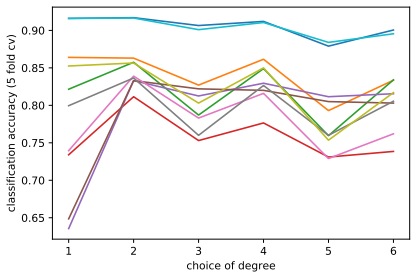
(c) Degree 3

(d) Degree 4

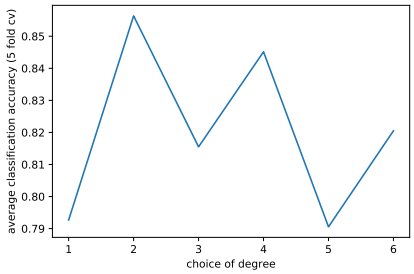
(e) Degree 5

(f) Degree 6

Figure 4.3: Plots of prediction accuracies vs regularization coefficients for multiple classification tasks defined by estimating the posterior of model parameter μ for 10 points $\mu \in (-20, 20)$, using the Gaussian model with polynomial kernels of different degrees. Each curve plotted in a figure corresponds to a different classification task defined by a single $\mu \in (-20, 20)$. For polynomial kernels of even degree, the reg coeff band captures the optimal regularization coefficients for all classification tasks considered.



(a) Max prediction accuracies for multiple classification tasks.



(b) Averaged max prediction accuracies for multiple classification tasks.

Figure 4.4: Plots of max prediction accuracies for optimal choices of regularization coefficient for multiple classification tasks defined by estimating posterior of model parameter $\mu \in (-20, 20)$, using the Gaussian model with polynomial kernels of different degrees. In Figure 4.4a, each curve plotted correspond to a different classification task defined by a single $\mu \in (-20, 20)$.

4.2.3.2 Interpretation

From the similar and dissimilar choices of optimal regularization coefficient observed, it seems that in feature spaces corresponding to polynomial kernel functions with even degrees, the data-sets X_m and X_μ are again similarly separated for different $\mu \in (-20, 20)$ and in feature spaces corresponding to polynomial kernel functions with odd degrees, the data-sets X_m and X_μ are not similarly separated for different $\mu \in (-20, 20)$.

Further note that using a kernel function with feature space mapping where the data-sets X_m and X_μ are best separated, we would expect the best classification accuracy from the KLR classifier (that attempts to linearly seperate the two data-sets in the mapped feature space). Conversely, we would expect that the kernel function that achieves the best classification accuracy with the KLR classifier corresponds to a feature space in which data-sets X_m and X_μ are best separated. Based on observed accuracies (Figure 4.4a), in the Gaussian model case, the polynomial kernel of degree 2 corresponds to a feature space where these data-sets are best separated.

4.2.4 Polynomial kernel: Posterior estimation

The summary statistics considered for the estimation of the posterior are the dimensions of the feature space mapped by the kernel function. Apriori, we would expect that the polynomial kernel of degree 1 provides a poor estimation of posterior probability density of μ for the Gaussian model, as it's mapped feature space $[1, x]$ certainly cannot capture the true analytical formula of the Gaussian model's posterior. For polynomial kernels of degrees 2-6, it is not obvious what choice of degree will provide the best estimation of posterior probability density.

For polynomial kernels of even degree: Using a common narrow regularization band for all tasks, (selected for the classification task defined by $\mu = 0$ as described in 6.1), and choosing regularization coefficient to maximize prediction accuracy on a common generated validation set for each task, the posterior probability density of μ was estimated for the Gaussian model at 100 points $\mu \in (-6, 6)$ for observed data $x_0 = 0$, as well as for 10 random observed data points $x_0 \in (-2, 2)$, so that the majority of the probability mass lies within the region $\mu \in (-6, 6)$.

For polynomial kernels of odd degree: The same posterior density estimation process is repeated, except using a regularization scale of 40 points $\in (10^{-2}, 10^{12})$.

4.2.4.1 Result

Figure 4.5 shows the posterior probabilities estimated for polynomial kernels of degrees 1-6 for observed data $x_0 = 0$ for a single run. The estimate closest to the true posterior of the model is achieved using a polynomial kernel of degree 2.

To measure the correctness of an estimated posterior density for a single observed data point, we may consider the symmetrised Kullback-Leibler divergence (sKL) between the estimated posterior and true posterior, where sKL divergence between two continuous distributions with densities p and q is defined as:

$$sKL(p||q) = \frac{1}{2}\int p(x)\log\frac{p(x)}{q(x)}dx + \frac{1}{2}\int q(x)\log\frac{q(x)}{p(x)}dx \tag{4.8}$$

A lower value of sKL divergence between the estimated posterior and the true posterior of a model's parameters corresponds to a better estimation. So, we would desire a larger negative value of sKL divergence for better estimation of the posterior of a model's parameters. Figure 4.6a shows a plot of negative sKL divergence for 10 random observed datapoints $x_0 \in (-2, 2)$ for different choices of polynomial kernel degree, while 4.6b shows a plot of the averaged negative sKL divergence for 100 random observed datapoints $x_0 \in (-2, 2)$ for different choices of polynomial kernel degree.

sKL divergence does show some variation depending on the observed datapoint. Typically, it is found that the largest negative values of sKL divergence (that corresponds to the best estimation of posterior probability density) is achieved by using a polynomial kernel of degree 2.

Interestingly, the trends observed in negative sKL divergence by varying polynomial kernel function degree are similar to the trends observed in cross validated prediction accuracy by varying polynomial kernel function degree in the classification between data-sets X_m and X_μ, as defined by the LFIRE method of posterior estimation. (Figure 4.4)

4.2.4.2 Interpretation

There seems to be a relationship between the most appropriate choice of feature space (corresponding to an appropriate choice of kernel function and kernel hyper parameter) that maximizes prediction accuracy classifying between data-sets X_m and X_μ and the most appropriate choice of feature space that minimizes sKL divergence between the posterior estimated using the LFIRE technique and the models true posterior. In particular, the summary statistics for generated data that are most appropriate for estimation of the posterior (corresponding to the dimensions of the feature space associated with choice of kernel function and kernel hyper parameter), also seem to appropriately capture information of the generated data for 'good' prediction accuracy, classifying between the data-sets X_m and X_μ, as defined by the LFIRE method of posterior estimation.

4.2.4.3 Consequence

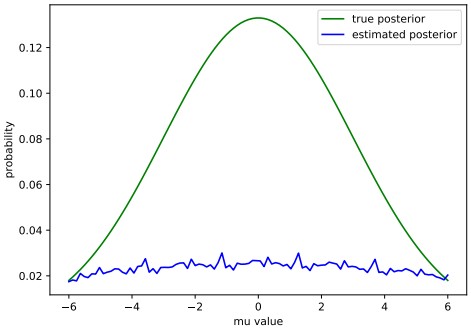
It can be difficult to know apriori what kernel function corresponds to a feature space that appropriately captures information of the generated data for posterior estimation. It might be possible to make a guess at the appropriate type of kernel function to use based on the structure of generated data. Then too, a kernel function is typically associated with some hyper-parameters that determine it's mapped feature space. (Such as the hyper-parameter d for polynomial kernels).

For data generating models for which the true posterior is available (typically artificial toy models), we can verify the choice of kernel function and associated hyper-parameter by evaluating the sKL divergence between the estimated posterior and true posterior.

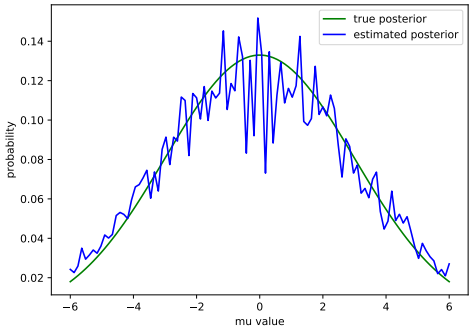
For data generating models for which the true posterior is not available (most models in real use, for which we would be interested in considering likelihood free inference of parameters), it can be very difficult to make an appropriate choice of kernel function, as the correctness of the estimated posterior becomes difficult to evaluate.

The observation that appropriate summary statistics (corresponding to an appropriate choice of kernel function and kernel hyper parameters) for posterior estimation using the LFIRE technique also allows better prediction accuracy classifying between the data-sets X_m and X_μ is potentially very useful. Given a selection of kernel functions or selection of hyper-parameters for a single kernel function, we would expect that the choices that achieve higher prediction accuracy are more likely to correspond to good choices for estimation of the posterior.

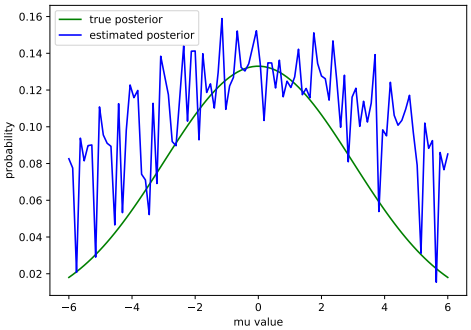
Potentially, given a choice of kernel functions, we can use prediction accuracy on tasks classifying between data-sets X_m and X_θ to roughly guide the selection of kernel



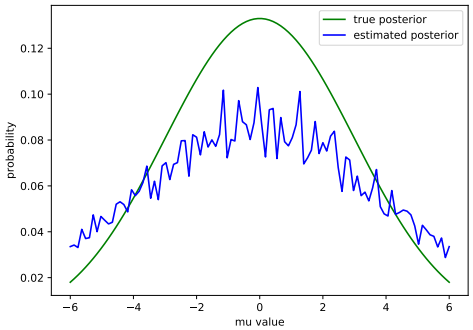
(a) Degree 1.



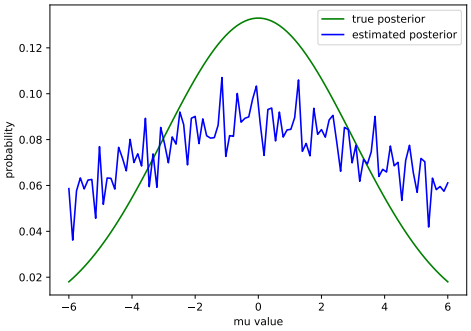
(b) Degree 2.



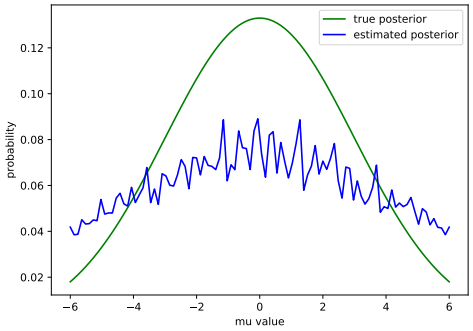
(c) Degree 3.



(d) Degree 4.

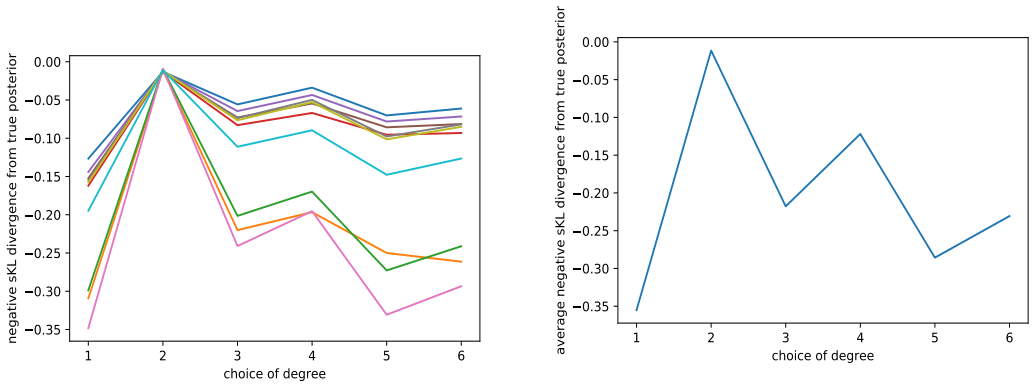


(e) Degree 5.



(f) Degree 6.

Figure 4.5: Estimated posterior probability density of model parameter μ at 100 points $\mu \in (-6, 6)$ for observed data $x_0 = 0$, using the Gaussian model with polynomial kernels of different degrees (Single run).



(a) 10 observed data points $x_0 \in (-2,2)$

(b) Averaged for 100 observed data points $x_0 \in (-2,2)$

Figure 4.6: Negative sKL divergence between estimated and true posterior densities of model parameter μ at 100 points $\mu \in (-6,6)$, using the Gaussian model with polynomial kernels of different degrees. In Figure 4.6a, each curve plotted correspond to the negative sKL divergence computed using a different observed data point $x_0 \in (-2,2)$.

functions and associated hyper-parameters (at the least eliminating choices that achieve poor prediction accuracies) to make an appropriate choice for estimation of the posterior, independent of a measure of correctness of the estimation.

4.2.5 RBF kernel: Classification Tasks

The RBF kernel function is commonly used by algorithms employing the kernel trick, and relies on computation of the Euclidean distance between data points in the input space.

$$k(\mathbf{x},\mathbf{x}') = e^{-\gamma\|\mathbf{x}-\mathbf{x}'\|^2} \tag{4.9}$$

The RBF kernel may be interpreted as a similarity measure that exponentially decreases with distance between vectors in the input space, the hyper parameter γ controlling the speed of this exponential decrease. The mapped feature space of the RBF kernel has an infinite number of dimensions and the complexity of the mapped space may be considered to increases as the magnitude of the hyper parameter γ increases.

Again, classification tasks defined by the estimation of the posterior of the Gaussian model's parameter μ are considered, at 10 evenly spaced points $\mu \in (-20,20)$. The KLR classifier was trained using RBF kernel functions with γ parameters $\in (10^{-12}, 10^{-1})$ on common generated data-sets.

Optimal regularization coefficients were selected separately for each task, for each choice of γ, from a scale $(10^{-12}, 10^{12})$ to maximize 5 fold cross validated prediction accuracy.

4.2.5.1 Result

Figure 4.7a shows the maximum achievable prediction accuracies, using RBF kernel functions with different hyper parameters $\gamma \in (10^{-12}, 10^{-1})$ and making optimal choices of regularization coefficient to maximize 5-fold cross validated prediction accuracy for the classification task defined by $\mu = 0$.

It is observed that above a threshold γ value, a similar classification accuracy can be achieved for any choice of γ. It is possible to automatically select this threshold given the achieved accuracies (described in the Appendix chapter section 6.1).

Figure 4.7b shows the variation in prediction accuracy for the classification task defined by $\mu = 0$, using an RBF kernel with the selected threshold γ value. Variation in classification accuracy with regularization coefficient is fairly smooth and again, it is possible to select a regularization band in the neighborhood of the maximizing regularization coefficient.

For the multiple classification tasks defined by $\mu \in (-20, 20)$, Figure 4.8a shows the maximum achievable classification accuracies by making optimal choices of regularization coefficient (to maximize 5-fold cross validated accuracy on common data-sets) for RBF kernel functions with hyper parameters $\gamma \in (10^{-12}, 10^{-1})$.

There is some variation in the threshold value γ as μ (that defines the classification task) varies. It is however observed that tasks defined by μ closer to 0 tend to have similar threshold γ values.

To highlight this, Figure 4.9a shows the maximum achievable classification accuracies for classification tasks defined by $\mu \in (-6, 6)$, along with the threshold γ value selected for the classification task defined by $\mu = 0$.

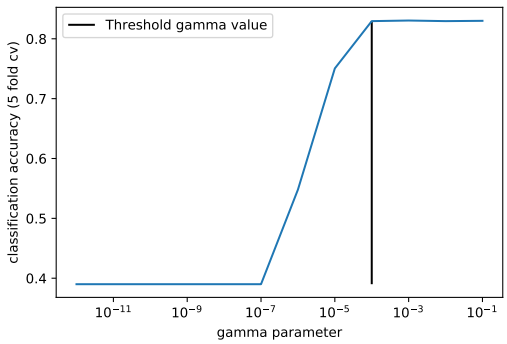
For the classification tasks defined by $\mu \in (-6, 6)$, using an RBF kernel with γ parameter equal to the threshold γ for the classification task defined by $\mu = 0$, again it is observed that optimal regularization coefficients for tasks defined by multiple $\mu \in (-6, 6)$ lie in the neighborhood of each other. Shown in Figure 4.9b along with the regularization coefficient band selected for the task defined by $\mu = 0$.

Additionally, as shown in Figure 4.8b, selecting optimal γ and regularization coefficients to maximize cross validated prediction accuracy, it is observed that maximum achievable prediction accuracy is typically lower for tasks defined by μ closer to the median of the prior distribution $\mu = 0$.

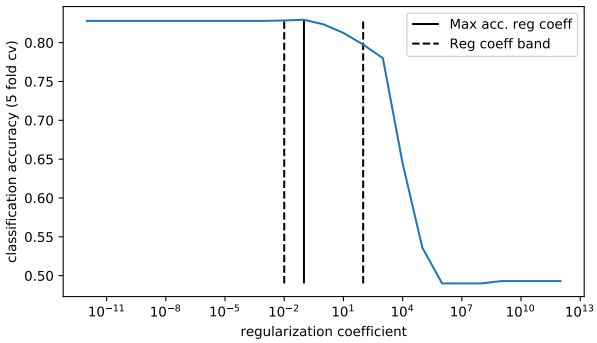
4.2.5.2 Interpretation

It seems that for a single classification task, all feature spaces mapped by an RBF kernel above a threshold complexity (corresponding to the threshold γ value for the classification task) are able to appropriately capture information about data points - so that separation of data-sets X_m and X_μ is possible to similar accuracies.

Summary statistics considered for the estimation of the posterior are the dimensions



(a) Max prediction accuracies for optimal choices of regularization coefficient vs choice of γ.

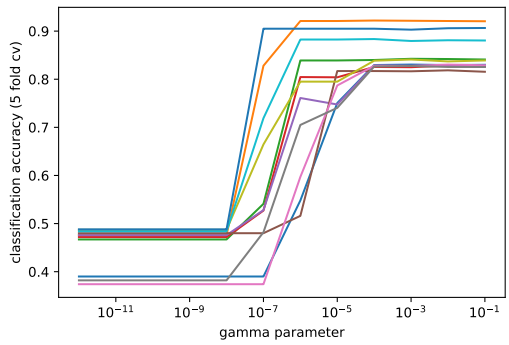


(b) Prediction Accuracies varying regularization coefficient, using chosen threshold γ.

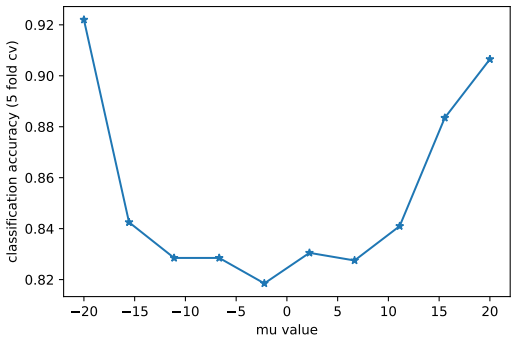
Figure 4.7: Prediction accuracies varying kernel hyper-parameter and regularization coefficient, classification task defined by estimating the posterior of the Gaussian model parameter μ at $\mu = 0$, using RBF kernels with different choices of γ. The reg coeff band captures the optimal regularization coefficients for all classification tasks considered. Above the chosen threshold γ, similar classification accuracies are achievable for optimal choice of reg coefficient.

of the feature space mapped by the kernel function. Then, following the observation that a feature space mapping that appropriately captures the data for estimation of the posterior also appropriately captures the data for 'good' classification accuracy in separation of data sets X_m and X_μ, the observation that the threshold γ value varies according to μ (that defines the classification tasks) and that tasks defined by μ closer to the median select higher threshold γ suggests that summary statistics that are sufficient for estimation of the posterior $p(\mu|x)$ for μ close to the fringe of the assumed prior (U(-20,20)) are not necessarily sufficient for estimation of the posterior closer to the median of the assumed prior distribution.

It is simple to explain the variation in threshold γ in light of the classification tasks defined. A larger γ corresponds to a more complex feature space. We expect tasks defined by μ closer to the median of the assumed prior to be more difficult, requiring



(a) Max prediction accuracies for optimal choices of regularization coefficient vs choice of γ.

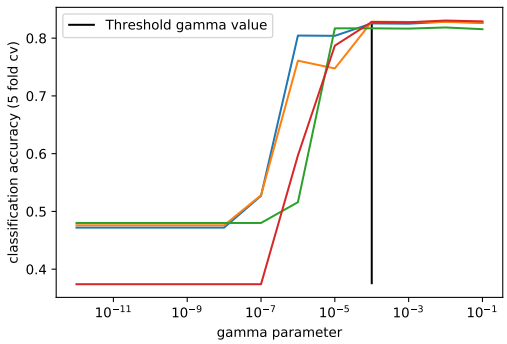


(b) Max prediction accuracies for optimal choices of regularization coefficient and γ.

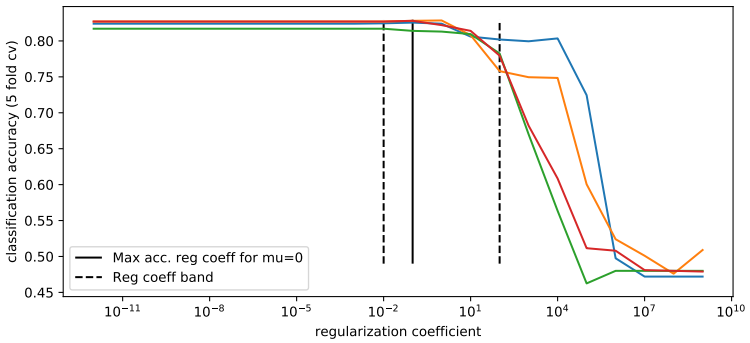
Figure 4.8: Prediction accuracies for classification tasks defined by estimating the posterior of the Gaussian model parameter μ at $\mu \in (-20, 20)$, using the RBF kernel. In Figure 4.8a, each curve plotted correspond to a different classification task defined by a single $\mu \in (-20, 20)$.

more complex feature space mappings for optimal separation - The observation that tasks defined by μ closer to the median of the prior distribution are 'harder' was previously made using a linear kernel with basis expansion $[1, x, x^2]$ for generated data-points (Figure 4.1b). Figure 4.13b shows the maximum achievable prediction accuracy for each classification task defined by $\mu \in (-20, 20)$ for optimal choice of γ and regularization coefficient. Again, it is observed that the tasks defined by μ closer to the median of the prior distribution achieves typically lower maximum prediction accuracy and consequently these choices of μ must define harder classification tasks.

Similar optimal regularization coefficients for tasks defined $\mu \in (-6, 6)$ suggests that the tasks are of similar difficulty close to the median of the prior distribution in the mapped feature space.



(a) Max prediction accuracies for optimal choices of regularization coefficient vs choice of γ.



(b) Prediction Accuracies varying regularization coefficient, using chosen threshold γ.

Figure 4.9: Prediction accuracies for classification tasks defined by estimating posterior of model parameter μ at $\mu \in (-6,6)$, using the Gaussian model with the RBF kernel. Each curve plotted correspond to a different classification task defined by a single $\mu \in (-6,6)$. The reg coeff band captures the optimal regularization coefficients for all classification tasks considered. The threshold γ is chosen for the classification task defined by $\mu = 0$.

4.2.5.3 Consequence

Following the observation made for previous experiments, that a feature space mapping that appropriately captures the data for estimation of the posterior also appropriately captures the data for 'good' classification accuracy in separation of data sets X_m and X_μ, conversely we would like to try and maximize prediction accuracy on classification tasks defined by $\mu \in (-6,6)$, to try to find a feature space mapping that is appropriate to estimate the posterior at these points.

By making a choice of threshold γ value and selecting a band of regularization coefficients from a single classification task, we are able to select an appropriate kernel function hyper parameter and define a narrow band of regularization coefficients that come close to satisfying the optimal choices for all tasks defined by $\mu \in (-6,6)$, again allowing us

to reduce computational costs for posterior estimation.

4.2.6 RBF kernel: Posterior estimation

The summary statistics considered for the estimation of the posterior are the dimensions of the feature space mapped by the RBF kernel function for a chosen γ value. Following the observation made for previous experiments, that a feature space mapping that appropriately captures the data for estimation of the posterior also appropriately captures the data for 'good' classification accuracy in separation of data sets X_m and X_μ, conversely we can now attempt to select an appropriate hyper-parameter γ for the RBF kernel function for posterior estimation at $\mu \in (-6,6)$ by maximizing prediction accuracy on classification tasks defined by these points.

Using the threshold γ value and common narrow regularization band selected for the classification task defined by $\mu = 0$ (as described in 6.1) for all tasks, and choosing regularization coefficient to maximize prediction accuracy on a common generated validation set for each task, the posterior probability density of μ was estimated for the Gaussian model at 100 points $\mu \in (-6,6)$ for observed data $x_0 = 0$, as well as for 100 random observed data points $x_0 \in (-2,2)$, so that the majority of the probability mass lies within the region $\mu \in (-6,6)$.

For comparison, the posterior probability density was estimated at the same points, for the same observed data, using the same generated data sets, using a polynomial kernel of degree 2 and the common narrow regularization band selected for the classification task defined by $\mu = 0$ for a polynomial kernel of degree 2 (chosen as described in 6.1).

4.2.6.1 Result

Figure 4.10 shows the classification accuracies achieved by using the polynomial kernel of degree 2 and the RBF kernel with the selected threshold value on the generated validation sets for tasks defined by $\mu \in (-6,6)$.

The classification accuracies achieved using the two kernel functions are very similar across the different classification tasks.

For a single run, Figure 4.11 shows the estimated posterior for the observed data point 0, using both the polynomial kernel of degree 2 and the RBF kernel at chosen threshold γ.

Both kernel functions are able to achieve reasonable estimates of the true posterior.

To compare the effectiveness of estimated posteriors p and q, by using different kernel functions, for a single observed data point, we can consider the difference in sKL divergence from the true posterior, using each of the two kernel functions for posterior estimation. That is,

$$\delta sKL(p||q) = sKL(p||r) - sKL(q||r) \tag{4.10}$$

Where r is the true posterior probability density.

For 100 observed data points $x_0 \in (-2,2)$, Figure 4.12 shows the δsKL between estimated distributions using the polynomial kernel of degree 2 and the RBF kernel with the chosen threshold γ.

A positive value of δsKL indicates a better estimate using the RBF kernel than using the Polynomial kernel.

For observed data points $x_0 \in (-2,2)$, the estimate using the RBF kernel at chosen threshold γ is always better than the estimate using the polynomial kernel of degree 2. It is also noted that:

- The difference in KL divergence is very small (of the order 10^{-3}), so that the estimated posteriors must be very nearly the same, such that both estimated posteriors are very similar to the true posterior, as visualized for observed data-point 0 (Figure 4.11)
- The choice of γ and regularization scale for each classification task was close to the optimal selection, but were not selected to exactly maximize prediction accuracy (done to reduce computational costs) and it is possible that a better estimate can be achieved by using the RBF kernel with exact choices of γ and regularization coefficient to maximize prediction accuracy for each task.

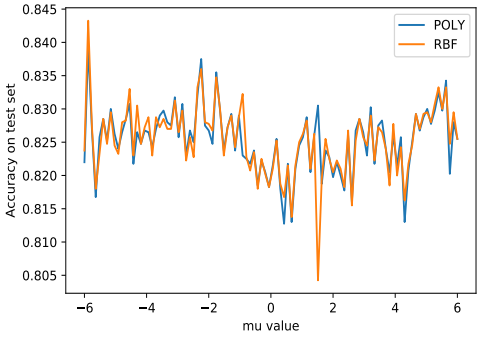
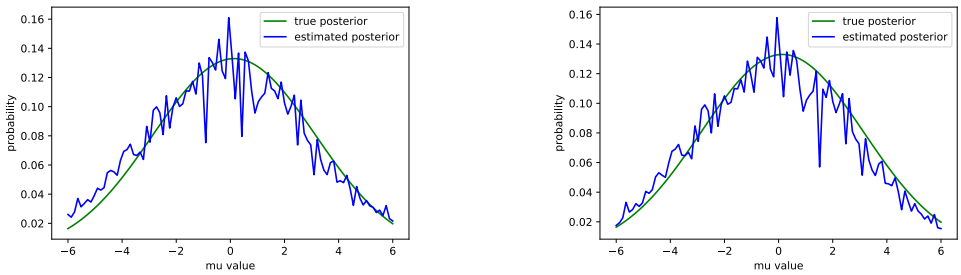


Figure 4.10: Using the Gaussian model, with the RBF kernel with selected threshold γ and the Polynomial kernel of degree 2: Prediction accuracies on a generated validation set, for classification tasks defined by model parameter μ at 100 points $\mu \in (-6,6)$.

4.2.6.2 Interpretation

The reasonable estimate of the posterior density that is achieved at data points $\mu \in (-6,6)$ by selecting the RBF kernel hyper-parameter γ that achieves close to optimal prediction accuracies separating X_m and X_μ for $\mu \in (-6,6)$, shows that at least for selection of the hyper-parameter of an appropriate kernel function, maximizing the prediction accuracy on the tasks separating X_m and X_θ (as defined by the LFIRE technique) can guide the selection of a feature space (defined by the kernel function and kernel hyper-parameters used) that appropriately captures information about the generated data for posterior estimation.



(a) Polynomial kernel (b) RBF kernel

Figure 4.11: Estimated posterior probability density of model parameter μ at 100 points $\mu \in (-6, 6)$ for observed data $x_0 = 0$, using the Gaussian model with polynomial kernels degree 2, and RBF kernel with the chosen threshold γ.

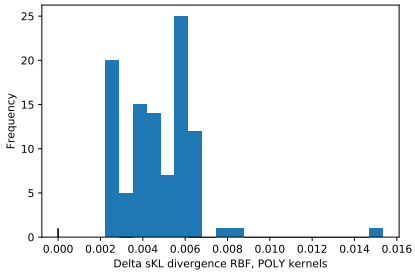


Figure 4.12: (δsKL divergence: Polynomial kernel - RBF kernel) for 100 observed data points $x_0 \in (-2, 2)$, where the posterior of model parameter μ is estimated at 100 points $\mu \in (-6, 6)$, using the Gaussian model.

4.2.6.3 Consequence

The selection of an appropriate hyper-parameter for the RBF kernel (and consequently appropriate feature space) for the estimation of the posterior of the Gaussian model serves to demonstrate that given an appropriate kernel function (associated with a family of feature space mappings), we can guide the selection of an appropriate feature space (that is appropriate choice of kernel hyper-parameter), for posterior estimation by maximizing the average prediction accuracy for tasks defined by the LFIRE technique (for different choice of model parameters) without necessarily requiring a measure of the 'correctness' for the estimated posterior.

The very similar prediction accuracies achieved by using both the RBF kernel and Polynomial kernel on classification tasks defined by $\mu \in (-6, 6)$ (Figure 4.10) along with the very similar divergence so the estimated posteriors from the true posterior suggests that it might be possible to associate with the estimation of the posterior of a single setting of model parameters θ an optimal classification accuracy that corresponds to use of the summary statistics best suited for posterior estimation (where summary statistics correspond to the dimensions of the feature space associated with a choice of kernel function and kernel hyper parameter).

4.3 Experiments with the ARCH(1) model

The data generating model considered for parameter inference is a lag-one auto-regressive model with conditional heteroskedasticity (ARCH(1)). The observed data x_0 generated from the model is a time series of a specified number of time steps T, such that:

$$x_0 = (y^{(t)}, t = 1, ..., T) \tag{4.11}$$

For time steps $y^{(t)}$,

$$y^{(t)} = \theta_1 y^{(t-1)} + e^{(t)}, \quad e^{(t)} = \xi^{(t)} \sqrt{0.2 + \theta_2 (e^{(t-1)})^2}, \quad t = 1, ..., T, \quad y^{(0)} = 0 \tag{4.12}$$

Where $\xi^{(t)}$ and $e^{(0)}$ are independent standard normal random variables and the parameters in the model θ_1 and θ_2 correspond to the mean and variance process coefficients. Uniform priors $U(-1,1)$ and $U(0,1)$ are assumed for the model parameters θ_1 and θ_2 respectively.

The true posterior distribution of $\theta = (\theta_1, \theta_2)$ for the ARCH(1) model can be numerically computed (see Gutmann et al. (2018, Appendix 1.2)).

ARCH models are popularly used for financial modelling (see Xekalaki and Degiannakis (2010)) and the ARCH(1) model was previously considered for posterior inference using the LFIRE technique by (Dutta et al. (2016)).

In the experiments that follow, we consider 2 types of vector representations for data generated from the model:

- The first representation is the default setting wherein for T time steps, the observed data considered for posterior inference is the vector $x_0 = (y^{(t)}, t = 1, ..., T)$
- In the second representation, we only consider data generated from the model every 10th time step. That is $x_0 = (y^{(t)}, \ t = 1, ..., T$ and $(t \mod 10) = 0)$. This preprocessing step was previously considered by (Rüping (2001)) for SVM classification of data generated from a lag-one auto-regressive model, where it improved prediction accuracy.

Using the LFIRE technique to estimate the posterior of the model's parameters θ for points in a mesh-grid $\theta \in [-1,1] \times [0,1]$, we define classification tasks:

Classifying between:

- N data points generated from the ARCH(1) model for parameters θ: (X_θ)
- N data points generated from the ARCH(1) model for random $\bar{\theta}$ values sampled from the prior distribution $\bar{\theta_1} \sim U(-1,1)$, $\bar{\theta_2} \sim U(0,1)$: (X_m).

For each $\theta \in [-1,1] \times [0,1]$ value we wish to estimate the posterior for. In the following experiments, for each classification task we generate N =1000 data points for each set X_θ, X_m.

4.3.1 RBF kernel: Classification Tasks

As the RBF kernel relies on computation of the Euclidean distance between data points in the input space and the Euclidean distance between time series vectors in the input space captures the cumulative difference between their corresponding time steps, the RBF kernel function is a interesting choice for the classification of time series data.

Classification tasks defined by the estimation of the posterior of the ARCH(1) model at 25 evenly spaced points $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$ are considered under both vector representations for data. That is, data is generated from the ARCH(1) model for T = 1000 time steps and we consider:

- Classification tasks using the default vector representation of generated data, representing data points by their first 100 time steps.
- Classification tasks using the second representation of generated data, representing data points as vectors of every 10th time step of the generated data, so that vectors used are of length 100.

For classification tasks under both vector representations, the KLR classifier was trained using RBF kernel functions with γ parameters $\in (10^{-12}, 10^{-1})$ on common generated data-sets.

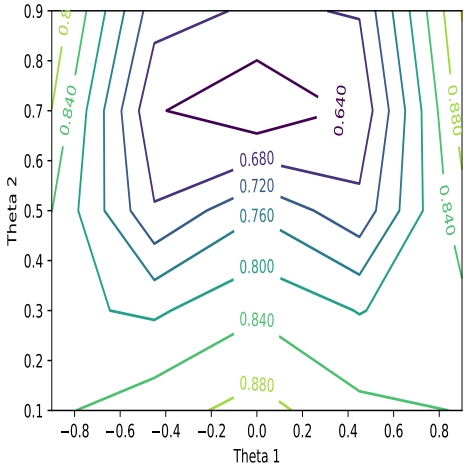
Optimal regularization coefficients were selected separately for each task, for each choice of γ, from a scale $(10^{-12}, 10^{12})$ to maximize 5 fold cross validated prediction accuracy.

Additionally, as the magnitude of the RBF kernel function depends on the Euclidian distance computed between vectors generated from the model, as an additional preprocessing step, vectors representing data points are scaled by the square root of their length, to avoid large variances in the kernel gram matrix (that typically makes it harder to minimize the objective function (3.3)).

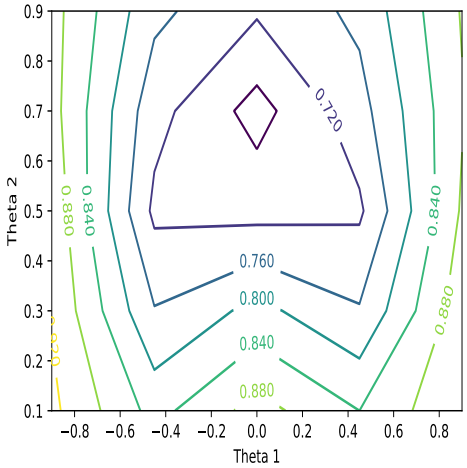
4.3.1.1 Result

For both vector representations of generated data, Figure 4.13 shows the maximum achievable classification accuracies for classification tasks defined by $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$. In both cases, similar to observations with the Gaussian model, it is found that selecting optimal γ and regularization coefficients to maximize cross validated prediction accuracy, the maximum achievable prediction accuracy is typically lower for tasks defined by θ with θ_1 closer to the median of the prior distribution $\theta_1 = 0$. This is not the case for θ_2, as maximum achievable prediction accuracy is lowest for tasks defined by θ with θ_2 close to 0.7.

Also similar to the observations for the Gaussian model, as shown in Figures 4.14a and 4.15a, for both choices of representation for generated data, it is found that most classification tasks defined by $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$, have similar threshold γ values, with which prediction accuracy close the maximum can be achieved.



(a) Classification tasks using vectors of 100 time steps for generated data.



(b) Classification tasks using vectors of every 10th time step for 1000 time steps for generated data.

Figure 4.13: Contour plots of max prediction accuracies for optimal choices of regularization coefficient and γ, classification tasks defined by estimating the posterior of the ARCH(1) model's parameters $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$, using the RBF kernel.

Further, as shown in Figure 4.14b, and 4.15b, for both vector representations of generated data, using a RBF kernel with γ parameter equal to the majority threshold γ for classification tasks, it is observed that optimal regularization coefficients for tasks defined by multiple $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$ lie in the neighborhood of each other.

4.3.1.2 Interpretation

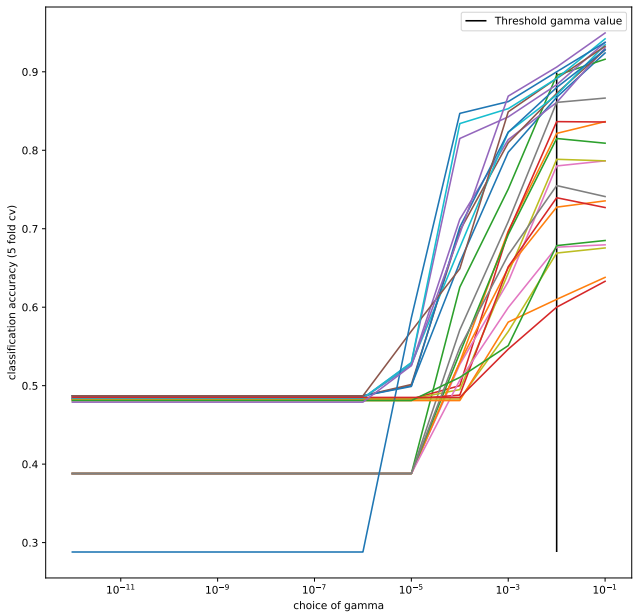
Using a feature space mapping that meaningfully captures some information about the generated data, we expect tasks defined by θ closer to the median of the prior to be more difficult (as discussed in section 4.2.1.2). This was found to be the case in previous experiments with the linear and RBF kernels for the Gaussian model (Figures 4.1b. 4.13b). Again, this observation holds for the parameter θ_1, as it is found that the tasks defined by θ with θ_1 closer to the median of the prior distribution achieve typically lower maximum prediction accuracies and consequently these choices of θ_1 must define harder classification tasks. However, this is not the case for the parameter θ_2, as maximum prediction accuracies are found to be lowest for tasks defined by θ with θ_2 close to 0.7. Possibly, we can interpret that the feature space mapped by the RBF kernel for optimal choice of γ is not able to very meaningfully capture information relevant to the parameter θ_2 from the generated data. This would not be surprising, considering that the data generated by the ARCH(1) model is more directly dependent on θ_1 than θ_2 (see (4.12)).

The similar optimal selection of γ and regularization coefficients for multiple tasks defined by $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$ suggests that these tasks are of similar difficulty in the mapped feature space.

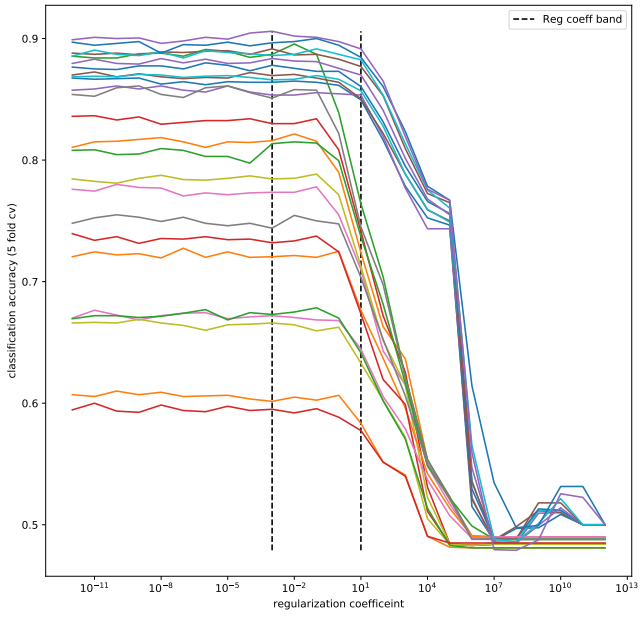
4.3.1.3 Consequence

Following the observation for previous experiments with the Gaussian model, that a feature space mapping that appropriately captures the generated data for estimation of the posterior also appropriately captures the generated data for 'good' classification accuracy in separation of data sets X_m and X_μ, we would like to try to select an appropriate hyper-parameter γ for the RBF kernel function for posterior estimation at $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$ by maximizing prediction accuracy on classification tasks defined by these points.

For both vector representations of generated data, by making the majority choice of threshold γ value and selecting a band of regularization coefficients from a single classification task, we are able to select an appropriate kernel function hyper-parameter and define a narrow band of regularization coefficients that come close to satisfying the optimal choices for the majority of tasks defined by $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$, again allowing us to reduce computational costs for posterior estimation.

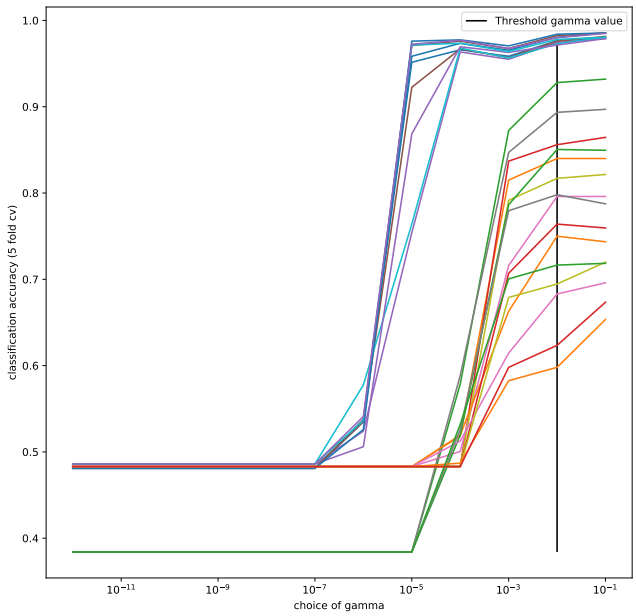


(a) Max prediction accuracies for optimal choices of regularization coefficient vs choice of γ.

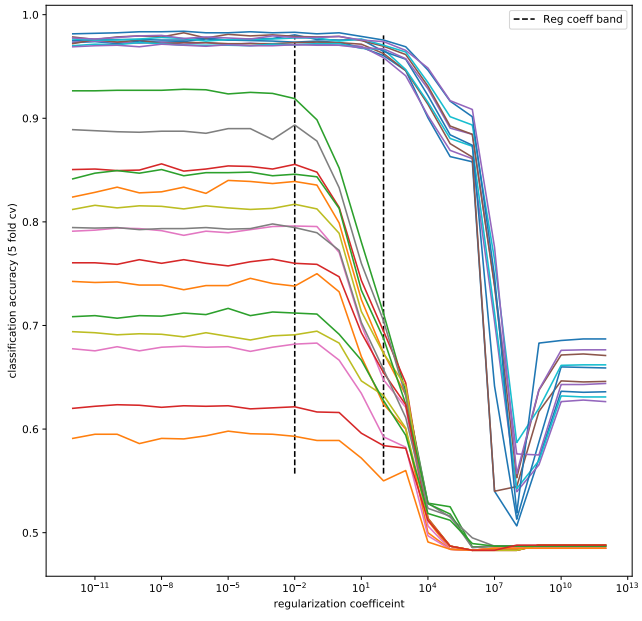


(b) Prediction Accuracies varying regularization coefficient, using chosen threshold γ.

Figure 4.14: Prediction accuracies for classification tasks using vectors of 100 time steps for generated data, defined by estimating the posterior of the ARCH(1) model's parameters θ at $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$, using the RBF kernel. The reg coeff band captures the optimal regularization coefficients for all classification tasks considered. The threshold γ is the majority threshold chosen by all classification tasks.



(a) Max prediction accuracies for optimal choices of regularization coefficient vs choice of γ.



(b) Prediction Accuracies varying regularization coefficient, using chosen threshold γ.

Figure 4.15: Prediction accuracies for classification tasks using vectors of every 10th time step for 1000 timesteps for generated data, defined by estimating the posterior of the ARCH(1) model's parameters θ at $\theta \in [-0,75, 0,75] \times [0.1, 0.9]$, using the RBF kernel. The reg coeff band captures the optimal regularization coefficients for all classification tasks considered. The threshold γ is the majority threshold chosen by all classification tasks.

4.3.2 RBF kernel: Posterior Estimation

Following the observation made for previous experiments using the Gaussian model, that a feature space mapping that appropriately captures the data for estimation of the posterior also appropriately captures the data to achieve 'good' classification accuracy in separation of data sets X_m and X_μ, and the demonstration made for the Gaussian model, that we can use this fact to guide selection of an appropriate kernel hyper parameter for the RBF kernel, we would like to attempt to select appropriate hyper-parameter γ for the RBF kernel function for posterior estimation at $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$ by maximizing prediction accuracy on classification tasks defined by these points.

For both vector representations of generated data, using the majority choice of threshold γ value and a common narrow regularization band for all tasks, and choosing regularization coefficient to maximize prediction accuracy on a common generated validation set for each task, the posterior probability density of θ was estimated for the ARCH(1) model at 100 points in the meshgrid $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$ for 100 random observed data points $\theta_0 \in [-0.6, 0.6] \times [0.3, 0.7]$, so that the majority of the probability mass lies within the estimated region.

4.3.2.1 Result

Figure 4.16a shows the difference in prediction accuracies for the two vector representations of generated data, achieved by using the RBF kernel with their chosen threshold γ values and optimal regularization coefficients selected on a common generated validation set for each task defined by $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$.

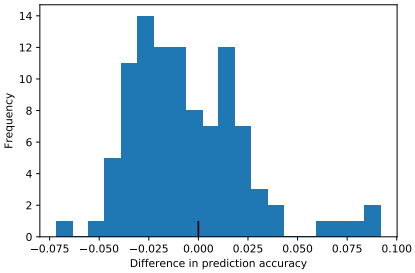
As shown in Figure 4.16a, neither vector representation achieves a clearly better classification accuracy on all defined classification tasks. The average difference in classification accuracies achieved is close to zero.

In contrast, the δsKL computed between these two estimated distributions is very large. Shown in Figure 4.16b. That is, the estimated posterior using a RBF kernel with the vector representation of every 10th time step for 1000 time steps is significantly better than the estimated posterior using a RBF kernel with the vector representation of the first 100 time steps.

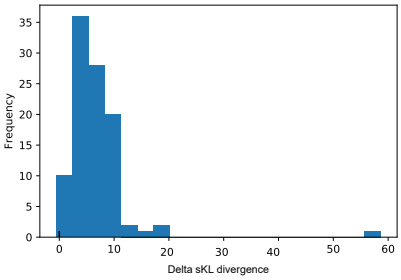
The posterior estimated using a RBF kernel with both vector representations is quite poor as shown in Figures 4.17 and 4.18.

4.3.2.2 Interpretation

As the data generating model is a lag-one model (that is, value of a time step $y^{(t)}$ is dependent on the value of the previous time step $y^{(t-1)}$) we would expect that the vector representation only considering every 10th time step ignores some information that is important for the estimation of the posterior. It is therefore not very surprising that the posterior probability estimated by using a RBF kernel with this vector representation is poor.

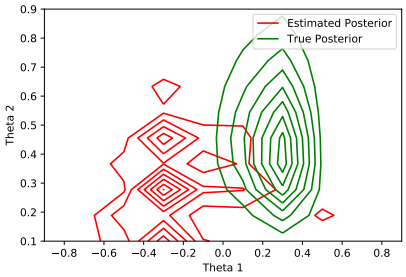


(a) Difference in prediction accuracies on common validation sets for tasks defined by 100 points $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$.

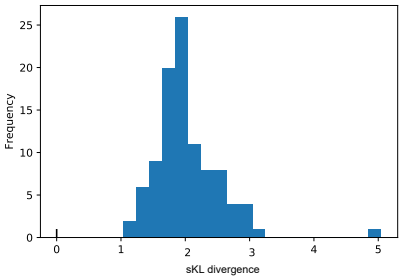


(b) δsKL divergence for 100 observed data points $\theta_0 \in [-0.6, 0.6] \times [0.3, 0.7]$, where the posterior of model parameter μ is estimated at 100 points $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$.

Figure 4.16: δsKL divergence and difference in prediction accuracies on a common validation set: Using the RBF kernel with first 100 time steps for generated data - Using the RBF kernel with every 10th time step for 1000 time steps for generated data, for the ARCH(1) model.



(a) Contour plot of the estimated and true posteriors for a typical observed data point. sKL divergence = 2.08.



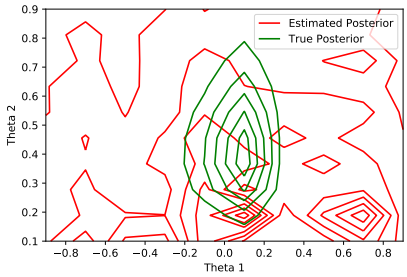
(b) Histogram of sKL divergence computed between estimated and true posteriors for 100 observed data points $\theta_0 \in [-0.6, 0.6] \times [0.3, 0.7]$, where the posterior of model parameter μ is estimated at 100 points $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$.

Figure 4.17: Results using the RBF kernel with vectors of every 10th time step for 1000 time steps for generated data to estimate the posterior of model parameters for the ARCH(1) model.

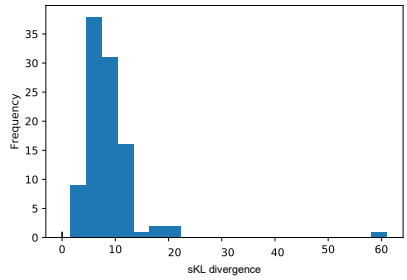
The estimated posterior using a RBF kernel with vector representation of the first 100 time steps is very poor. Possibly, this is because by using vectors of successive time steps, the Euclidean distance computed between generated data points captures too much random noise.

Interestingly, while the vector representation using only every 10th time step is far more appropriate for estimation of the posterior of the model's parameters (see Figure 4.16b) neither representation achieves clearly better classification accuracies on all tasks defined by $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$. That is, a feature space mapping that is appropriate for estimation of the posterior and a feature space mapping that is not appropriate for estimation of the posterior may both achieve similar prediction accuracies on the classification tasks defined by estimation of the posterior of a model's parameters by the LFIRE technique.

While this observation does not contradict the observation previously made, that a feature space mapping that appropriately captures the generated data for estimation of the posterior also appropriately captures the generated data to achieve 'good' classification



(a) Contour plot of the estimated and true posteriors for a typical observed data point. sKL divergence = 6.27.



(b) Histogram of sKL divergence computed between estimated and true posteriors for 100 observed data points $\theta_0 \in [-0.6, 0.6] \times [0.3, 0.7]$, where the posterior of model parameter μ is estimated at 100 points $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$.

Figure 4.18: Results using the RBF kernel with vectors of the first 100 time steps of generated data to estimate the posterior of model parameters for the ARCH(1) model.

accuracy in separation of data sets X_m and X_θ, is cautions that we can alternatively capture information about the generated data that is appropriate for classification between the data sets but not appropriate for posterior estimation of the model's parameters.

4.3.2.3 Consequence

This experiment demonstrates that it is possible to select a feature space that is appropriate for the classification tasks defined by LFIRE but not appropriate for posterior estimation. Consequently, in the case of feature space mappings that capture different types of information about of the generated data, and achieve similar average classification accuracies on common validation sets, we would not expect that classification accuracy can guide the selection of an appropriate feature space for posterior estimation.

It is however noted that for most kernel functions, we would not expect the type of information captured by the mapped feature space to vary drastically as the kernel hyper-parameter varies. Consequently, feature spaces that achieve similar classification accuracies on the tasks defined by the LFIRE technique but do not capture similar type

of information of the generated data are unlikely within the family of feature spaces defined by a single kernel function.

Therefore, following the observation that a feature space mapping that appropriately captures the data for estimation of the posterior also appropriately captures the data to achieve 'good' classification accuracy on the LFIRE classification tasks, for a kernel function that is appropriate for posterior estimation, we would still expect that we can use classification accuracy to guide an exact selection of kernel hyper-parameters that are appropriate for posterior estimation.

Chapter 5

Conclusions

5.1 Discussion

The main research questions explored in this thesis are:

- How we can guide an appropriate selection of kernel function and kernel hyper parameters for the posterior estimation of a given model's parameters using the LFIRE technique.
- How we can minimize the costs of using the expensive Kernel Logistic Regression technique in its application to the LFIRE technique.

In relation to the first research question, it was observed that a feature space mapping (corresponding to a choice of kernel function and kernel hyper parameter) that appropriately captures the generated data for estimation of the posterior also appropriately captures the generated data to achieve 'good' classification accuracy on the classification tasks defined by the LFIRE technique.

Using a ARCH(1) data generating model and the RBF kernel, it was demonstrated that it is also possible to select a feature space that is appropriate for the classification tasks defined by LFIRE technique but not appropriate for posterior estimation of a model's parameters.

However, for most kernel functions, we would not expect the type of information captured by the mapped feature space to vary drastically as the kernel hyper-parameter varies. Consequently, feature spaces that achieve similar classification accuracies on the tasks defined by the LFIRE technique but do not capture similar type of information of the generated data are unlikely within the family of feature spaces defined by a single kernel function. For an appropriate choice of kernel function, it is therefore meaningful to consider maximization of the average prediction accuracy of tasks defined by the LFIRE technique to guide the exact selection of the kernel function's hyper parameters that are appropriate for estimation of the posterior of a model's parameters.

Using a Gaussian data generating model and the RBF kernel, it was demonstrated that maximization of the average prediction accuracy of tasks defined by the LFIRE

technique can guide the exact selection of a kernel function's hyper parameters that are appropriate for estimation of the posterior of a model's parameters.

In relation to the second research question, for both the Gaussian data generating model and the ARCH(1) data generating model, it was observed that the classification tasks defined by the LFIRE technique for varied settings of model parameters are roughly of similar difficulty, so that a single choice of optimal kernel function hyper parameters can be made, and a narrow band of regularization coefficients can be selected for this choice of hyper parameters, to achieve near optimal prediction accuracies for all such tasks. This is useful when we attempt to guide the selection of a kernel function's hyper parameters by maximizing the average prediction accuracy of classification tasks defined by the LFIRE technique, as by using this single choice of hyper parameters and narrow regularization band, rather than performing an extensive grid search for optimal parameters, we can reduce the number of times we train Kernel Logistic Regression for posterior estimation by the LFIRE technique.

5.2 Further work

There exist a very rich selection of kernel functions to capture information of generated data in complex high dimensional spaces. It would be interesting to see how results compared with current state of the art benchmarks for estimation of the posterior of model parameters for more complex models. We could apply KLR to LFIRE for more complex data generating models and kernel functions, by using prediction accuracy to guide the selection of kernel hyper parameters, and making a single choice of appropriate kernel hyper parameters and selecting a narrow regularization coefficient band for all tasks defined by LFIRE to reduce computational costs.

Another interesting avenue to explore is kernel combination for LFIRE. Selecting an appropriate kernel function for estimation of the posterior requires some knowledge of the structure of generated data. It would be interesting to try to eliminate this necessity. A single kernel function may or may not be appropriate for estimation of the posterior. If it is appropriate for estimation of the posterior, then maximizing prediction accuracy should guide selection of appropriate kernel function hyper parameters. We could do this for a large number of varied kernel functions regardless of whether or not they are appropriate for estimation of the true posterior and consider a combination of the feature spaces selected (mapped by each kernel function with choice of hyper parameter). The theory of kernel functions allows us to do this most simply by addition of kernel functions (using the individually selected kernel hyper parameters for each kernel function in the combination).

$$k_{12}(x,x^{'}) = k_1(x,x^{'}) + k_2(x,x^{'}) \tag{5.1}$$

As the LFIRE technique automatically selects relevant summary statistics for posterior estimation from the feature space used, it should also automatically discern appropriate information associated with any appropriate kernel function that is included in the combination.

Chapter 6

Appendix

6.1 Selection of a threshold γ hyper parameter and regularization coefficient band for a kernel function

This section details the procedure followed for selection of the threshold γ parameter of a kernel function, for a single classification task, using which it is possible to achieve similar classification accuracy as with the optimal choice of γ, while favouring less complex feature spaces.

This section also details the procedure followed for the selection of a regularization coefficient band in the neighborhood of the optimal regularization coefficient for a single classification task.

Algorithm 1 concisely describes the procedure used to make these selections.

In Algorithm 1, $options$ specifies the number of options for the chosen regularization band, $width$ specifies how broad the chosen regularization band is and $percentage1$ and $percentage2$ specify how close the chosen parameters are to the parameter selection that yields the highest cross validated accuracy. In all experiments conducted in this thesis, unless otherwise mentioned, a regularization scale of ($options = 20$) points is selected of ($width = 4$), using ($percentage1 = 0.95, \quad percentage2 = 0.90$).

Specifically, in the case of the RBF kernel, using Algorithm 1, we select for the threshold γ value the smallest γ that, for optimal selection of regularization coefficient, achieves a prediction accuracy close to the maximum achievable prediction accuracy (for optimal choice of γ). We define 'close' as covering the difference between maximum accuracy (for optimal choice of γ) and maximum accuracy using the smallest γ value, up to a percentage $percentage1$.

For a fixed γ value, to select a regularization coefficient band, first we select an upper limit of the scale, by choosing the largest regularization coefficient that covers the difference between maximum achievable prediction accuracy for that choice of γ and accuracy corresponding to the largest choice of regularization coefficient, up to a percentage $percentage2$. For a lower limit of the band, select 10^{-width} times the upper

limit. The granularity of the band chosen is defined by the specified number of *options* desired.

Note, by making selections with respect to a percentage of the difference between accuracies at different parameter/regularization choices, the selections are chosen invariant of the magnitude of variance of prediction accuracies for the classification task.

Figure 4.7 shows the selections of threshold γ and a regularization coefficient band using the RBF kernel for a single classification task.

Algorithm 1 Selection of kernel hyper-parameters and regularization band

scaleReg is a broad list of options for the regularization coefficient, in decreasing order of magnitude.

scaleHyper is a list of options for the kernel hyper-parameter, in increasing order of complexity of the mapped feature space.

scores is a matrix such that scores[i,j] $\leftarrow$ 5 fold cross validated prediction accuracy using scaleReg[i] regularization coefficient and scaleHyper[j] kernel hyper parameter to train the KLR classifier for the classification task.

1: **procedure** ($options, width, percentage1, percentage2$)

2: γ_baseScore = max(scores[0]) + percentage1*(max(scores)- max(scores[0]))
3: $\gamma \leftarrow$ smallest value i for which max(scores[i]) $>=$ γ_baseScore
4: γ_*choice* $\leftarrow$ scaleHyper[γ]
5:
6: *C_baseScore* = scores[γ,0] + percentage2*(max(scores[γ])- scores[γ,0])
7: $C \leftarrow$ smallest value j for which scores[γ,j] $>=$ C_baseScore
8: $C\_upper \leftarrow$ scaleReg[C]
9:
10: band $\leftarrow$ linspace($C\_upper * 10^{-width}$, C_upper, $options$)
11: **return** (γ_*choice*, band)

Bibliography

Mark A Beaumont, Wenyang Zhang, and David J Balding. Approximate Bayesian Computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

Ritabrata Dutta, Jukka Corander, Samuel Kaski, and Michael U Gutmann. Likelihood-free inference by ratio estimation. *arXiv preprint arXiv:1611.10242*, 2016.

Michael U Gutmann, Ritabrata Dutta, Samuel Kaski, and Jukka Corander. Likelihood-free inference via classification. *Statistics and Computing*, 28(2): 411–425, 2018.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning, section 12.3. 2001.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python. `http://www.scipy.org/`, 2001–. [Online; accessed April-2018].

Stefan Rüping. Svm kernels for time series analysis. Technical report, Technical Report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund, 2001.

Mark Schmidt. Kernel logistic regression. `https://www.cs.ubc.ca/~schmidtm/Software/minFunc/examples.html`, 2008. [Online; accessed September-2017].

Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized Representer Theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.

Csar Souza. Popular kernel functions. `http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications`. [Online; accessed April-2018].

Mikael Sunnåker, Alberto Giovanni Busetto, Elina Numminen, Jukka Corander, Matthieu Foll, and Christophe Dessimoz. Approximate Bayesian Computation. *PLoS computational biology*, 9(1):e1002803, 2013.

Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael PH Stumpf. Approximate Bayesian Computation scheme for parameter inference and model

selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31): 187–202, 2009.

Simon N Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102, 2010.

Evdokia Xekalaki and Stavros Degiannakis. *ARCH models for financial applications*. John Wiley & Sons, 2010.

Ji Zhu and Trevor Hastie. Kernel logistic regression and the import vector machine. In *Advances in neural information processing systems*, pages 1081–1088, 2002.