Kernel Methods for Likelihood-Free Inference

Amrit Seshadri Diggavi

UG4 Honors Project Dissertation, 2018

Introduction

- When modeling some process, we would generally try to select the model parameters that are most viable given data generated from the process. We are therefore often interested in the posterior probability density of the models parameters, given some observed data $p(\theta|x_0)$.
- Statistical inference of the posterior pdf typically requires the likelihood function

$$\underbrace{p(\theta|x)}_{\text{Posterior}} \propto \underbrace{p(x|\theta)}_{\text{Likelihood}} \underbrace{p(\theta)}_{\text{Prior}} \tag{1}$$

- For complex models, the likelihood function may not be available and numerical computation intractable.

Introduction

- When it is possible to sample from the model,

  $$x_\theta \sim M(\theta) \tag{2}$$

  we can use likelihood free inference methods to estimate the posterior that bypass the need for an analytical formula for the likelihood function
- The LFIRE technique is a recent likelihood free inference technique that approximates the posterior by estimating the ratio $r(x, \theta)$ between the data generating pdf $p(x|\theta)$ and the marginal pdf $p(x)$.
- It relies on training a logistic regression classifier to do so

  $$X_\theta \sim M(\theta) \tag{3}$$

  $$X_m \sim M(\bar{\theta}) \quad \text{where for each } \mathrm{x}_m \in X_m \quad \bar{\theta} \sim p(\theta) \tag{4}$$

Introduction

- The posterior estimated by likelihood free inference methods is heavily dependent on the choice of summary statistics used for generated data.

- Likelihood free inference methods (ABC, synthetic likelihood) typically require a careful selection of summary statistics for generated data. The LFIRE technique allows a very broad selection of summary statistics for observed data, selecting out relevant stats by regularized logistic regression.

- To capture complex information of the data generated from a model a very broad list of summary statistics for estimation of the posterior can be considered by compactly associating the list of summary statistics for *LFIRE* with a kernel function.

Introduction

- The kernel trick: If we know the dot product between data points in some complex high dimensional space (kernel function), then:
- Formulating a logistic regression classifier to only access data via the kernel functions of data points, we are able to search this complex high dimensional spaces without an explicit transformation of data. (Kernel Logistic Regression)

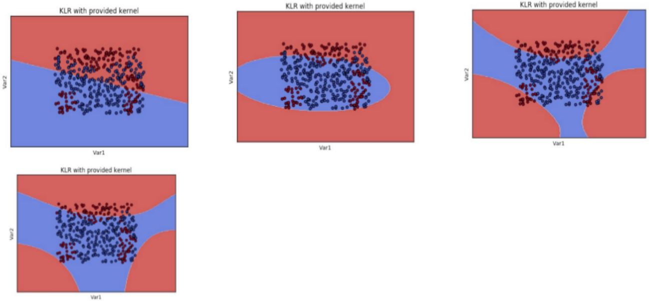


Figure: KLR - Nonlinear classification

Introduction

- Replacing the use of logistic regression with KLR for the *LFIRE* technique, the list of summary statistics considered for posterior estimation are the dimensions of the feature space associated with the kernel function.

Exploration

- How we can guide an appropriate selection of kernel function 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.
- Consider for exploration the Gaussian and ARCH(1) models, for which the true posterior can be computed so that the correctness of the estimated posterior can be evaluated.

Gaussian model

- The data generating model considered for parameter inference is a uni-variate Gaussian distribution:

  $$x_\mu \sim M(\mu) = \mathcal{N}(\mu, \sigma^2 = 9) \tag{5}$$

  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)).

Gaussian model

- Assuming a uniform prior $U(-20, 20)$ on μ, the functional form of the true posterior is

$$p(\mu|x_0) = \begin{cases} \exp\{\alpha_0(\mu) + \alpha_1(\mu)x_0 + \alpha_2(\mu)x_0^2\}, & \text{if } \mu \in (-20, 20) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

- So that summary stats $[1, x_0, x_0^2]$ are most appropriate for estimation of the posterior.

Gaussian model: Linear kernel

- 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.

Gaussian model: Linear kernel: Classification tasks

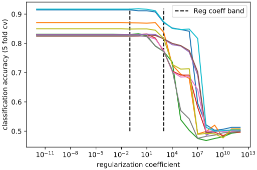


Figure: Prediction accuracy vs regularization coefficient.

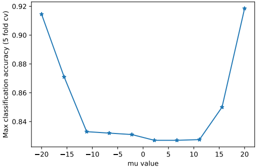


Figure: Max prediction accuracy for optimal choice of regularization coefficient.

Gaussian model: Linear kernel: Posterior estimation

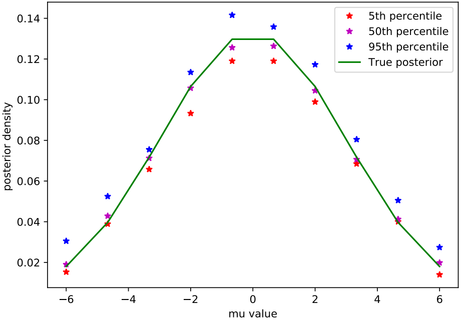


Figure: Multiple experiments run for estimation of the posterior probability density of model parameter μ at 10 points $\mu \in (-6, 6)$ for observed data $x_0 = 0$, using the Gaussian model with a linear kernel and basis expansion $[1, x, x^2]$ for generated data.

Gaussian model:Polynomial kernel

- The polynomial kernel function of degree d used is:

$$k_d(\mathbf{x}, \mathbf{x}') = (< \mathbf{x}, \mathbf{x}' > +1)^d \tag{7}$$

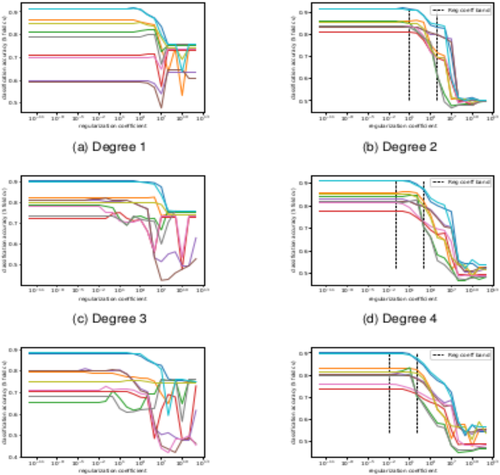
- So that the feature space mapping associated with a polynomial kernel function of degree d is:

$$\phi_d(x) = [x^d, \alpha_{d-1}x^{d-1}, ..., \alpha_1 x, 1] \tag{8}$$

  For some coefficients α_i .

- Not obvious what choice of degree d corresponds to the most appropriate choice of summary statistics for estimation of the posterior.

Gaussian model: Polynomial kernel: Classification tasks



(a) Degree 1

(b) Degree 2

(c) Degree 3

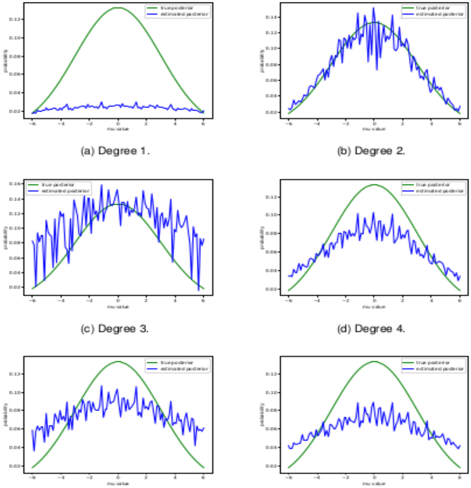
(d) Degree 4

(e) Degree 5

(f) Degree 6

Figure: Prediction accuracy vs Regularization coefficient

Gaussian model: Polynomial kernel: Posterior estimation



(a) Degree 1.

(b) Degree 2.

(c) Degree 3.

(d) Degree 4.

(e) Degree 5.

(f) Degree 6.

Figure: Posterior estimation

Gaussian model: Prediction Accuracy and sKL divergence

- sKL divergence between estimated and true posterior ditributions is used to measure of the correctness of the estimated posterior. A smaller sKL divergence indicates a better estimation.

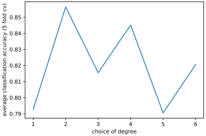


Figure: Averaged max prediction accuracies for multiple classification tasks.

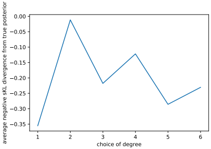


Figure: -sKL div averaged for 100 observed data points $x_0 \in (-2, 2)$

Gaussian model: RBF kernel

- 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.

  $$k(\mathbf{x}, \mathbf{x}') = e^{-\gamma \|\mathbf{x} - \mathbf{x}'\|^2} \tag{9}$$

  The complexity of the mapped space may be considered to increases as the magnitude of the hyper parameter γ increases.

Gaussian model: RBF kernel: Classification tasks

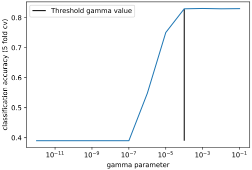


Figure: Max prediction accuracies for optimal choices of regularization coefficient vs choice of γ.

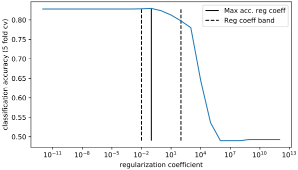


Figure: Prediction Accuracies varying regularization coefficient, using chosen threshold γ.

Gaussian model: RBF kernel: Classification tasks

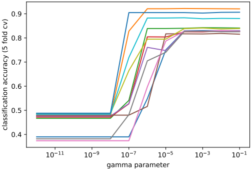


Figure: Max prediction accuracies for optimal choices of regularization coefficient vs choice of γ.

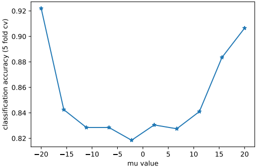


Figure: Max prediction accuracies for optimal choices of regularization coefficient and γ.

Gaussian model: RBF kernel: Classification tasks

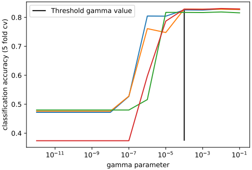


Figure: Max prediction accuracies for optimal choices of regularization coefficient vs choice of γ.

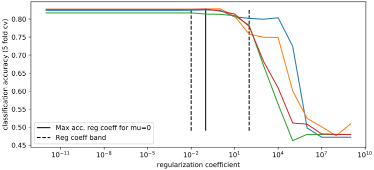


Figure: Prediction Accuracies varying regularization coefficient, using chosen threshold γ.

Gaussian model: Posterior estimation

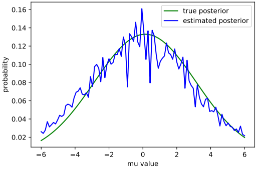


Figure: Polynomial kernel

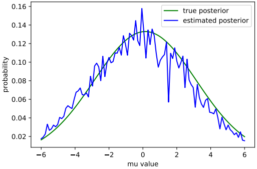


Figure: RBF kernel

Gaussian model: Posterior estimaiton

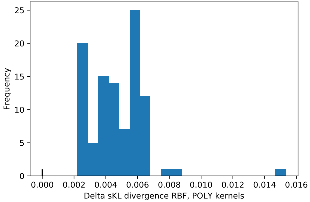


Figure: (δsKL divergence: Polynomial kernel - RBF kernel) for 100 observed data points $x_0 \in (-2, 2)$, where the posterior of model parameter μ is estimated at 100 points $\mu \in (-6, 6)$, using the Gaussian model.

ARCH(1) model

- The data generating model 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:

$$x_0 = (y^{(t)}, t = 1, ..., T) \tag{10}$$

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

$$y^{(t)} = \theta_1 y^{(t-1)} + e^{(t)}, \quad e^{(t)} = \xi^{(t)} \sqrt{0.2 + \theta_2 (e^{(t-1)})^2}, \quad y^{(0)} = 0 \tag{11}$$

ARCH(1) model

- The parameters θ_1 and θ_2 are the model parameters for which we would like to approximate the posterior.
- Uniform priors $U(-1, 1)$ and $U(0, 1)$ are assumed for the model parameters θ_1 and θ_2 respectively.
  The true posterior pdf of $\theta = (\theta_1, \theta_2)$ for the ARCH(1) model, $p(\theta|x_0)$, can be numerically computed (Gutmann et al. (2018, Appendix 1.2)).

ARCH(1) model: RBF kernel

- 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.

ARCH(1) model: RBF kernel

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 $x_0 = (y^{(t)}, t = 1, ..., T \text{ and } (t \mod 10) = 0)$. 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.

ARCH(1) model: RBF kernel: Classification tasks

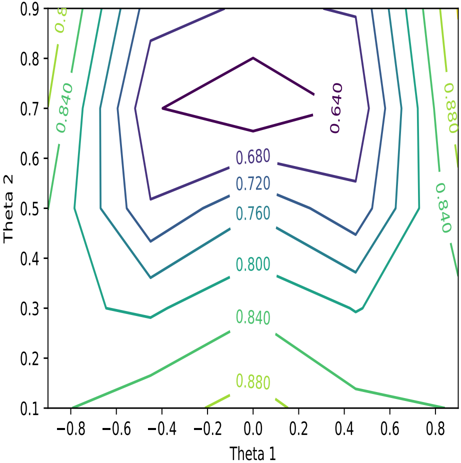


Figure: Classification tasks using vectors of 100 time steps for generated data.

ARCH(1) model: RBF kernel: Classification tasks

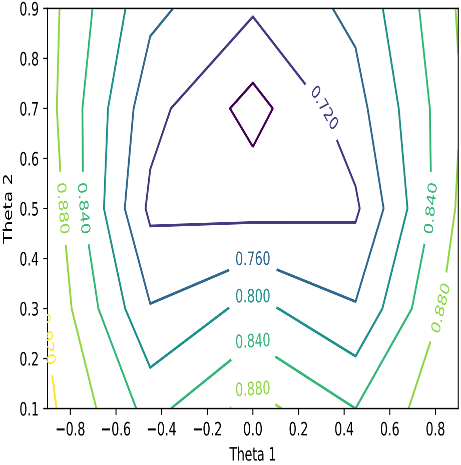


Figure: Classification tasks using vectors of every 10th time step for 1000 time steps for generated data.

ARCH(1) model: RBF kernel: Classification tasks

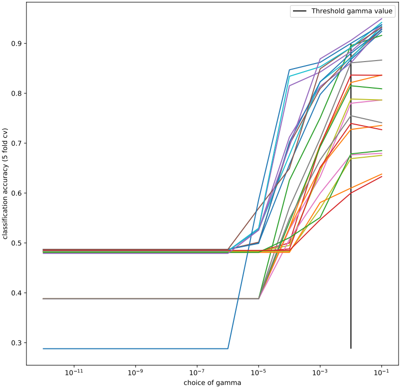


Figure: Representation 1: Max prediction accuracies for optimal choices of regularization coefficient vs choice of γ.

ARCH(1) model: RBF kernel: Classification tasks

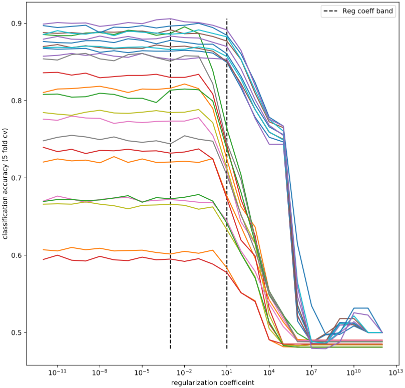


Figure: Representation 1: Prediction Accuracies varying regularization coefficient, using chosen threshold γ.

ARCH(1) model: RBF kernel: Classification tasks

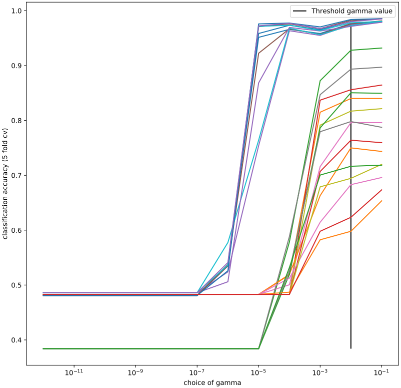


Figure: Representation 2: Max prediction accuracies for optimal choices of regularization coefficient vs choice of γ.

ARCH(1) model: RBF kernel: Classification tasks

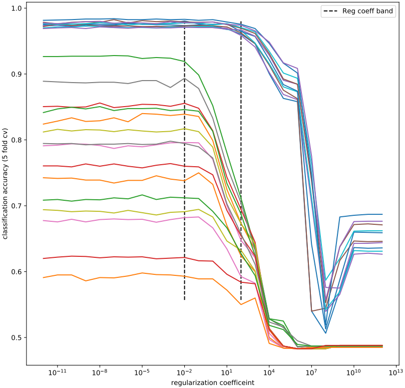


Figure: Representation 2: Prediction Accuracies varying regularization coefficient, using chosen threshold γ.

ARCH(1) model: RBF kernel: Posterior estimation

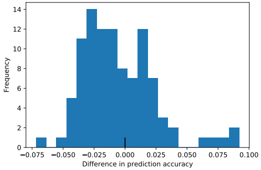


Figure: Difference in prediction accuracies on common validation sets for tasks defined by 100 points $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$.

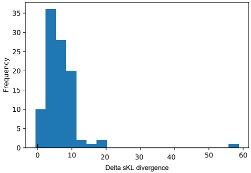


Figure: δsKL divergence for 100 observed data points $\theta_0 \in [-0.6, 0.6] \times [0.3, 0.7]$, where the posterior of model parameter μ is estimated at 100 points $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$.

ARCH(1) model: RBF kernel: Posterior estimation

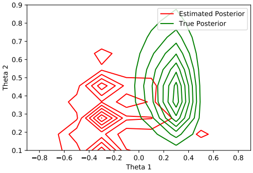


Figure: Contour plot of the estimated and true posteriors for a typical observed data point. sKL divergence = 2.08.

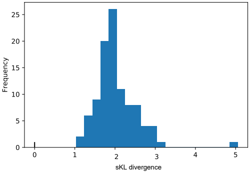


Figure: Histogram of sKL divergence computed between estimated and true posteriors for 100 observed data points $\theta_0 \in [-0.6, 0.6] \times [0.3, 0.7]$, where the posterior of model parameter μ is estimated at 100 points $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$.

ARCH(1) model: RBF kernel: Posterior estimation

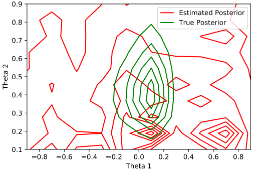


Figure: Contour plot of the estimated and true posteriors for a typical observed data point. sKL divergence = 6.27.

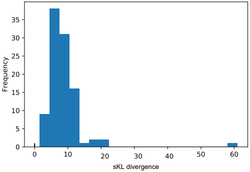


Figure: Histogram of sKL divergence computed between estimated and true posteriors for 100 observed data points $\theta_0 \in [-0.6, 0.6] \times [0.3, 0.7]$, where the posterior of model parameter μ is estimated at 100 points $\theta \in [-0.75, 0.75] \times [0.1, 0.9]$.

Conclusions

- 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.

Conclusions

- 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.

Conclusions

- 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.

Further work

- An 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.
- 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.

Further work

- 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).

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

- 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.

Bibliography

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.

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