COURSE 02457

Non-Linear Signal Processing: Answers to Exercise 10

Checkpoint 10.1

In this checkpoint we will get familiar with the Kernel matrix, which is defined as the inner product of the feature vectors of the data points.

$$K_{m,n} = \phi(x_m)^T \phi(x_n) = e^{-||x_m - x_n||^2/2\sigma 2} \tag{1}$$

The general idea of kernel representations for supervised learning is that similarity in input should lead to similarity in output, and the m, n entry of K tells us something about the correlation between data point x_m and x_n, or how similar they are. The standard deviation of the Kernel controls to what extent we define two inputs to be similar.

The columns of the Kernel matrix can be considered to be pseudo-features, and in a classification problem like this, we ca use the features to extract the information we are looking for; which points are similar and which ones are not. In order to get a clear picture from the Kernel matrix, it is important to choose the σ's for the kernels to be a suitable value. If it is too high, the the entries in K become larger, so the (right) picture becomes whiter, and we get less contrast. Less contrast means it is harder to classify the points. If it is too low, the image of the Kernel gets darker, and again it gets harder to distinguish the points. In general, it is a good idea to chose the σ to the mean of distances between the points.

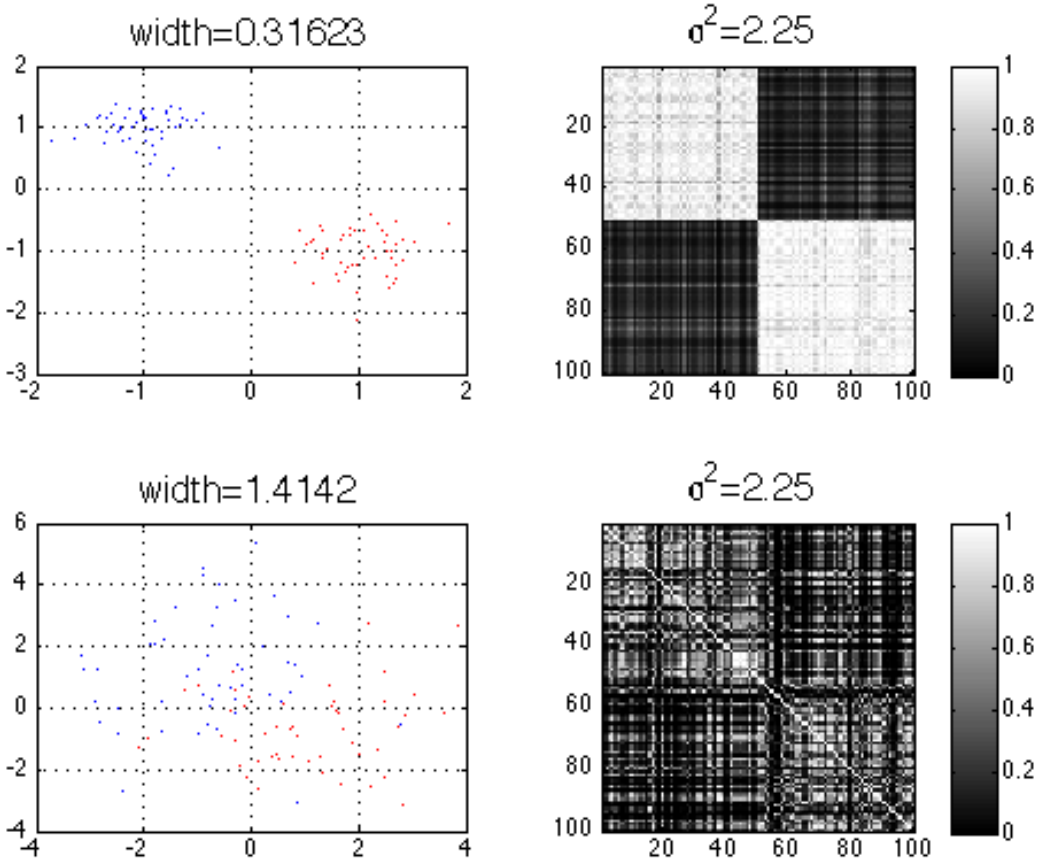


Figure 1: Data from to gaussians together with their Kernel matrix.

Next we investigate the Kernel matrices for the sun spot time series, see Fig. 2. In the rows of the kernel matrices it is easy to see the period of the sunspots, and from the columns we can see which points are at roughly in the same phase (same place in the cycle.)

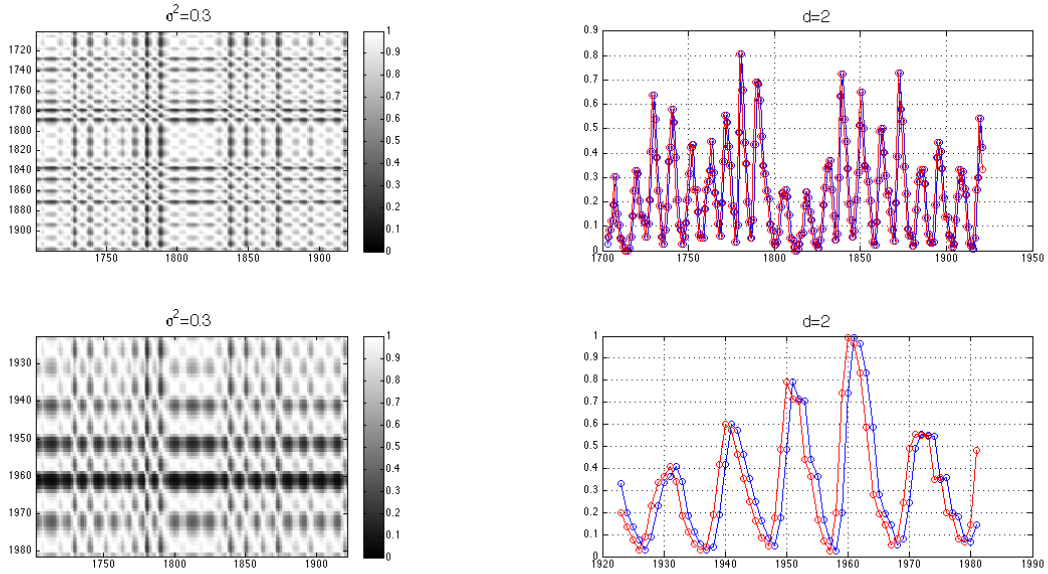


Figure 2: Kernel matrices and data together with the predictions.

Checkpoint 10.2

In this checkpoint we apply a simple Gaussian process (GP) model to the prediction of sunspots. Consider the training data set where we have the data and their labels x_{train}, t_{train}, and assume that the system is described by the target model $y_n = y(x_n)$. The targets of our data x_n are corrupted by zero mean independent random noise ϵ_n with precision β^{-1}:

$$t_n = y_n + \epsilon_n \tag{2}$$

The joint distribution of all the target values t conditioned on y is therefore $p(t|y) = \mathcal{N}(t|y, \beta^{-1}, I_N)$, where the marginal distribution of y is $p(y) = \mathcal{N}(y|0, K)$ with $K_{m,n} = e^{-||x_m - x_n||^2/2\sigma^2}$

The marginal distribution of t can then

$$p(t) = \int p(t|y)\ p(y)\ dy = \mathcal{N}(t|0, C) \tag{3}$$

where $C = K + \beta^{-1} I$. Now we want to predict the target variables t_{test}, so we need to evaluate the predictive distribution $p(t_{test}|t_{train})$. Following the results from Section 2.3.1 and 6.4.2, we do this by first finding the joint distribution

$$p(t_{test}, t_{train}) = \mathcal{N}(t_{test}, t_{train}|0, C_{test,train}) \tag{4}$$

where

$$C_{test,train} = \begin{pmatrix} C_{test} & k \\ k^T & C_{train} \end{pmatrix} \tag{5}$$

where $k = \mathbf{k}(x_{train}, x_{test})$ and $C_{test} = \mathbf{k}(x_{test}, x_{test}) + \beta^{-1}$.

By using Eqs. (2.81)-(2.82) from the book stating

$$\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(x_b - \mu_b) \tag{6}$$
$$\sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ab} \tag{7}$$

we can get the parameters for the predictive distribution from the parameters for the joint distribution:

$$\mu_{test|train} = k^T C_{train}^{-1} t_{train} \tag{8}$$
$$\sigma_{test|train} = \Sigma_{test} - k^T \Sigma_{train}^{-1} k \tag{9}$$

Since the optimal prediction is the expected conditional mean, we have that

$$\hat{t}_{test} = \mu_{test|train} \tag{10}$$

But first we need to find the parameters β and σ, and these are estimated by minimizing the error, both for the negative log likelihood and the least squared error. This is simply done by looping over a range of the parameters and choosing the best ones, see Fig. 3.

	σ	β
LL	0.4289	211.1053
LS	0.5337	289.8947

We see that the least squared error approach estimates higher values for both parameters.

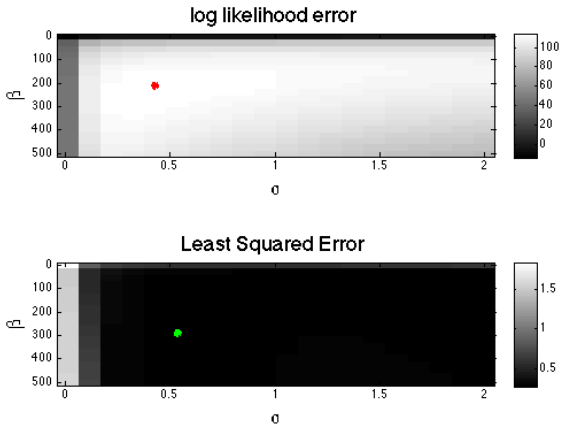


Figure 3: The optimal set of parameters for the log likelihood error and the squared prediction error.

The prediction made with the parameters from the two different methods can be seen in Fig. 4. We see that this method has higher error than the ones obtained by NN and RBF.

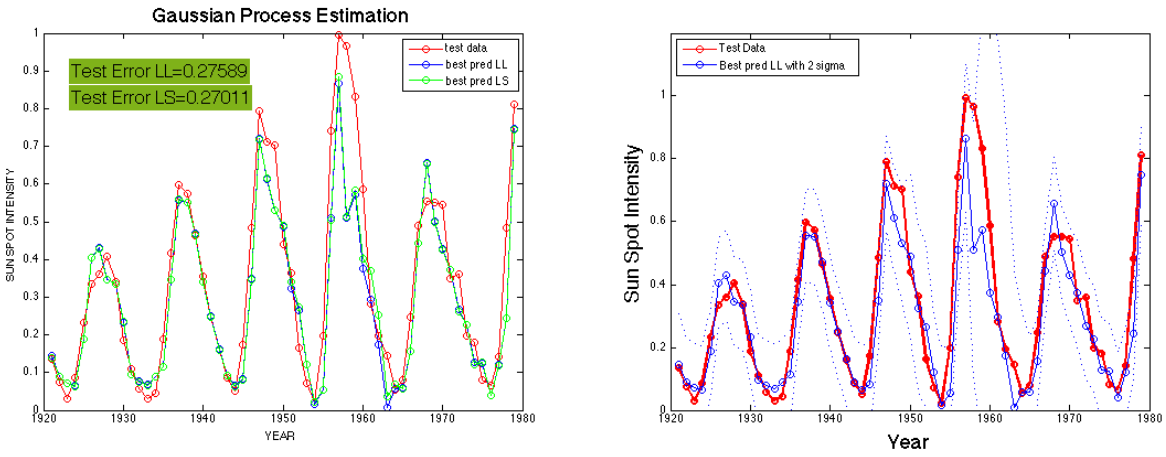


Figure 4

Checkpoint 10.3

In the kernel methods considered so far, the kernel function $k(x_n, x_m)$ had to be evaluated for all possible pairs x_n and x_m of training points, which can be computationally infeasible during training and can lead to excessive computation times when making predictions for new data points.

In this checkpoint we shall look at kernel-based algorithms that have sparse solutions, so that predictions for new inputs depend only on the kernel function evaluated at a subset of the training data points.

We are going to use the SVM to solve classification problems, first on synthetic data and then on the pima set. In both cases we want to predict the binary target label of new training points.

In Support Vector Machine Classification, we want to construct a decision boundary for classifying data so that the distance from the decision boundary to the closest point is maximized, as illustrated in Fig. 5. The location of the boundary is determined by a subset of the data points, and these are called the support vectors. The decision boundary $y(x)$ can be found by minimizing the Lagrangian

$$\tilde{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m) \tag{11}$$

$$s.t. \quad 0 \leq a_n \leq C \tag{12}$$

$$\sum_{n=1}^{N} a_n t_n = 0 \tag{13}$$

which gives us the necessary parameters C, σ and a_n to compute the boundary,

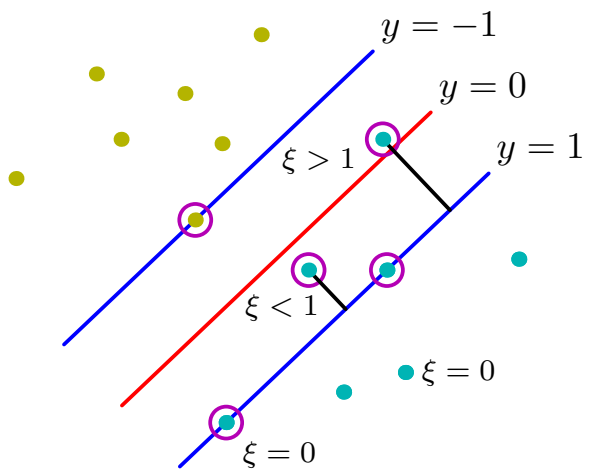


Figure 5: Illustration of the decision boundary, the margins and the slack variables. We want to maximize the margin while softly penalizing the points lying on the wrong side of the boundary.

$$y(x) = \sum_{n01}^{N} a_n t_n k(x, x_n) + b \tag{14}$$

and the target label of the new data points as the sign of $y(x)$.

In the script **main10d**, we apply this method on synthetic data, and the result can be seen in Fig. 6.

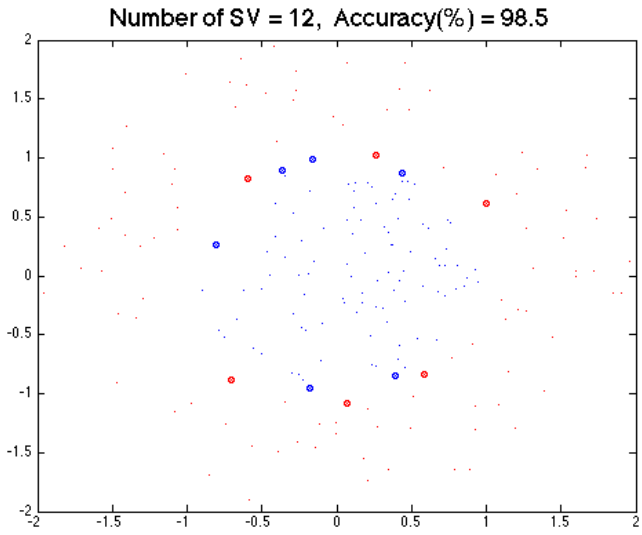
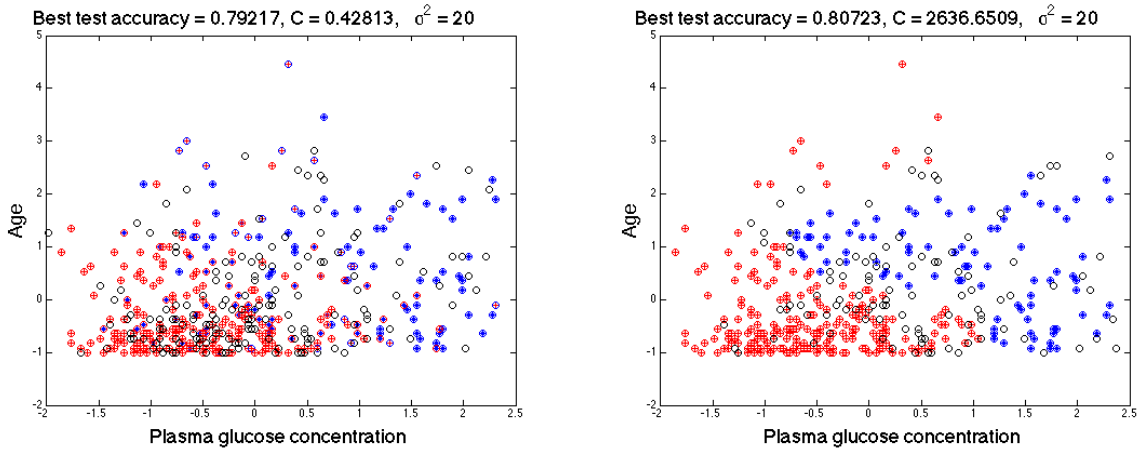


Figure 6

Next we try to classify the pima indian set.



(a) Input data comprises all 7 inputs.

(b) Input data comprises only input 2,7.

Figure 7: Data and their predicted labels. The color of the cross is the true label, and the color of the circle is the predicted label.

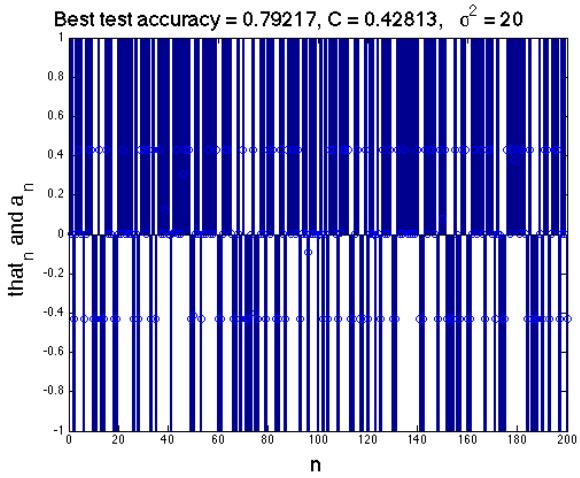


Figure 8: The bars indicate the true labels of the training set, and the circles are $a_n y$ for the SVM solution using all inputs.

Problems with SVM

The support vector machines suffer from some limitations:

The outputs of an SVM represent decisions rather than posterior probabilities. Also, the SVM was originally formulated for two classes, and the extension to $K > 2$ classes is problematic. There is a complexity parameter C, or ν (as well as a parameter ϵ in the case of regression), that must be found using a hold-out method such as cross-validation. Finally, predictions are expressed as linear combinations of kernel functions that are centered on training data points and that are required to be positive definite.