*COURSE 02457*

# Non-Linear Signal Processing: Answers to Exercise 9

## Checkpoint 9.1

When we want to estimate the parameters in a model after receiving a data set we, we want to to find the most probable set of parameters given the data. In other words, we want to maximize $p(w|x)$. Using Baye's rule, this is equivalent to maximizing the likelihood $\prod_{n=1}^{N} p(x_n|w)$, where N is the number of data points.

Maximizing the product of the conditional densities for each data point is the same as maximizing the sum of the logs of the densities, so we can also maximize $\sum_{n=1}^{N} log(p(x_n|w))$. If we now multiply the log likelihood by -1, we can interpret the negative log likelihood as an error that we want to minimize:

$$E(w) = \sum_{n=1}^{N} - log\ p(x_n|w) \tag{1}$$

In the case of **Kernel estimation**, the likelihood is given by

$$p(x|w) = \frac{1}{N} \sum_{n=1}^{N} k(x|x_n, w) \tag{2}$$

so the error of the model would be

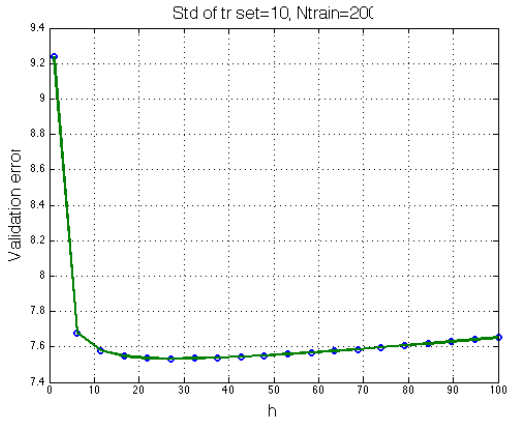
$$E(w) = \sum_{n=1}^{N} - log \left( \frac{1}{N} \sum_{n=1}^{N} k(x|x_n, w) \right) \tag{3}$$

where $k(x|x_n, w)$ is the kernel function that is usually given by the Parzen window or a gaussian centered on the point $x_n$, and there is one such kernel for each data point in the training set. Since the kernels are centered around the $x_n$'s, they can be interpreted as the parameters $\mu_n$.

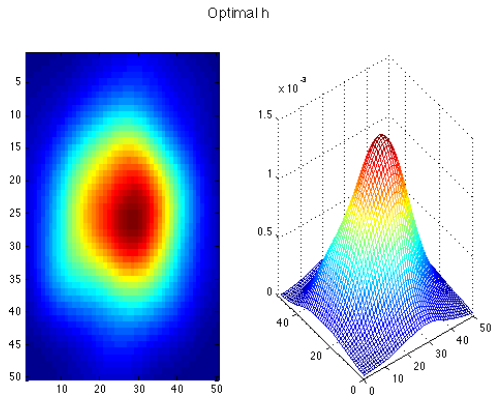
If we want to calculate the test error, the vector $x$ in the above equation is equal to $x_test$ consisting of M points, the expression for the test error becomes

$$E(w) = \sum_{m=1}^{M} - log \left( \frac{1}{N} \sum_{n=1}^{N} k(x_m|x_n, w) \right) \tag{4}$$

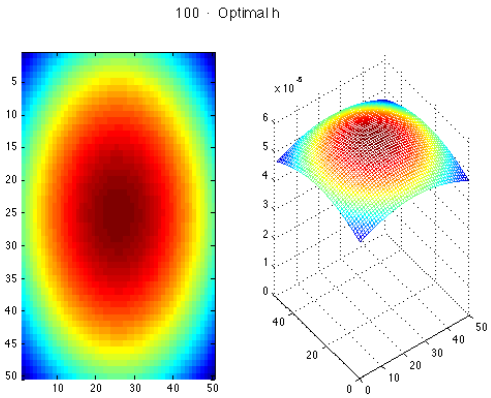
The test error can be used to find the optimal $h$.



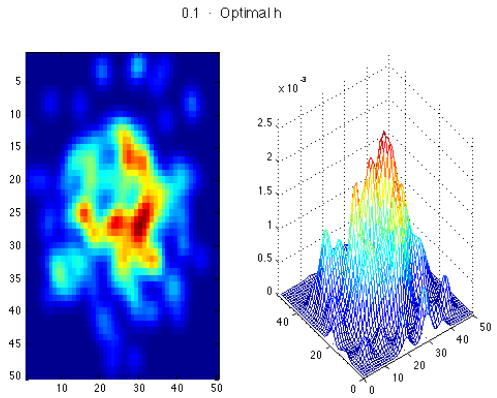

(a) The test error as a function of $h$. $h_{opt} = 27.0526$.




(b) The density estimate with the optimal band width, $h_{opt}$.




(c) The density estimate using $h = 100\ h_{opt}$.




(d) The density estimate using $h = 0.1\ h_{opt}$.

Figure 1: In all figures $N_{train} = 100$ and $\sigma^2_{train} = 10$.

We see that the width of the kernel $h$ plays a significant role. This is better illustrated using the figures from the book.

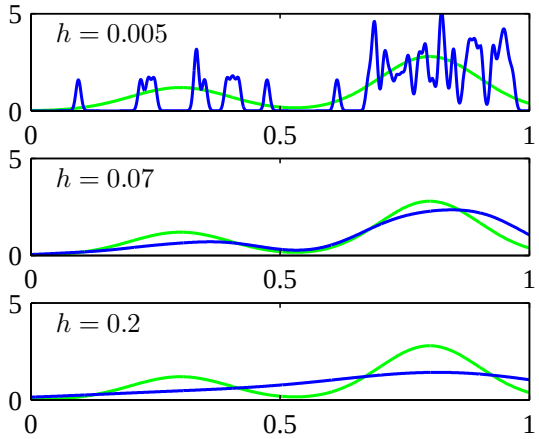



Figure 2: Example of the result of a density estimation with different $h$. he green line is the true density, the blue line is the estimated.

In Fig. 2 we see that if we have a too small $h$, the kernels centered around the training points are very narrow, and if we get a test point outside any of the kernels, the negative log likelihood (test error) of that test point will be very large.

In the other extreme case, the estimated likelihood is too low at the maxima of the true density, and since we are likely to get many test points at the maxima, we will get many points with a higher negative log likelihood than they should, and this will also create a large test error.

The optimal $h$ is somewhere in between the two extremes, and should be chosen to balance the variance and bias in two extreme cases.

Of the two extreme cases, the first one will be the most severe with the highest test error.

## Checkpoint 9.2

One of the main problems with the kernel estimation method described above is that all the kernels have the same width $h$, which can be problematic in practice when handling real data sets.

A way to overcome this is by instead of choosing a kernel that has a fixed width that may contain few or many points, we fix the number of neighboring points and let the volume of the kernel vary. This is known as the nearest neighbor method.

We are now going to use this method to classify the pima indian women. For each data point $x_m$ in the test set, we are going to pick $K$ of it's nearest neighbors in the training set and assign $x_m$ to class that the majority of it's training neighbors belong to.

$K$ is chosen as:

```
for k=Kmin:Kmax
    for all points x in training set:
      find k nearest neighbors, and classify
      x as the same as the majority the neighbors.
    end
    compute misclass-error
Kopt=k(index(min(misclass-error)))
end
```

The misclassification error can also be interpreted as a **leave-one-out-error**, because we for each training point, we select a subset of the data point around it and use this subset excluding the point to be classified to estimate the label. Then the estimate is checked against the true label in cross-validation procedure, as in Fig. 3.

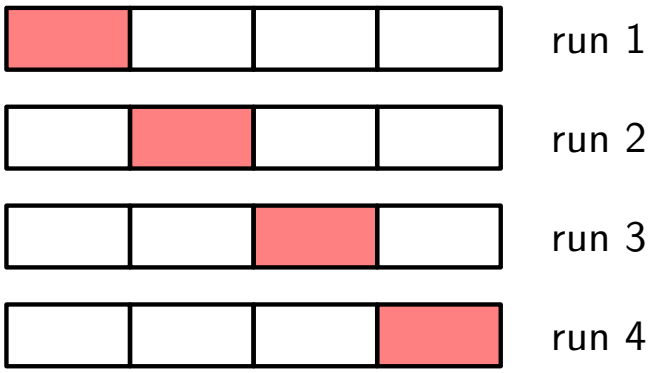
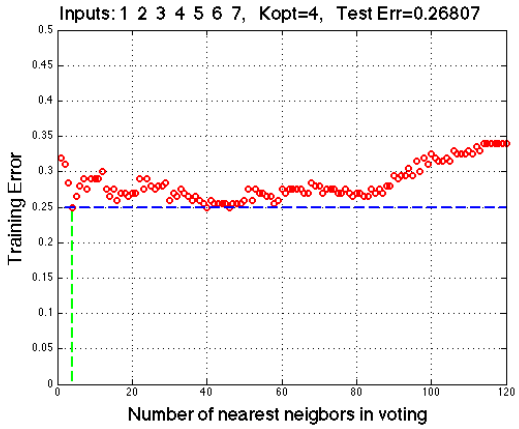
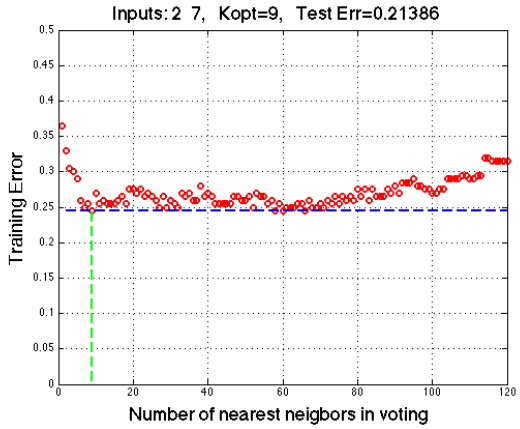



Figure 3: Leave-one-out cross-validation.

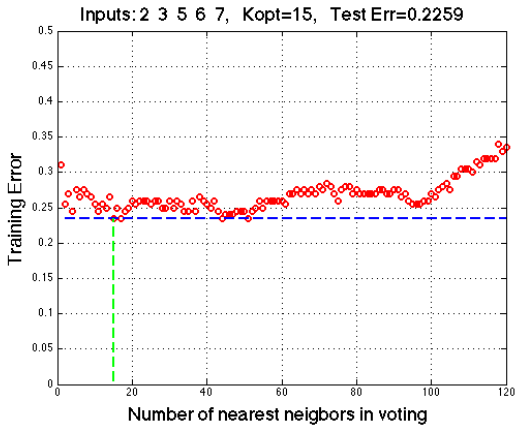
The pima women were classified using the entire input set as training data and different subsets, see Fig. 4. The set giving the lowest test error is the the subset with inputs 2 and 7 as expected, resulting in a $Kopt = 9$ and a misclassification error of $Etest = 0.2139$. In comparison, the lowest misclassification error found using neural networks was $Etest = 0.1988$ using all inputs.




(a) The training error as a function of neighbors for the pima set with all inputs.




(b) The training error as a function of neighbors for the pima set with inputs 2 and 7.




(c) The training error as a function of neighbors for the pima set with inputs 2, 3, 5, 6 and 7.

Figure 4: Finding the optimal number of neighbors for different dimensionality of the input data.

## Checkpoint 9.3

In this check point we are going to model the sunspot data applying a linear model with regularization to each data point based on the K-nearest neighbors in the training set. The regularization is necessary because the when the regression is based on few data points, we risk over fitting.

The parameters $K$,$d$ are both found by minimizing the prediction error, and the the optimal $d$ is 5 for $\alpha = 0.001$, see Fig. 5.

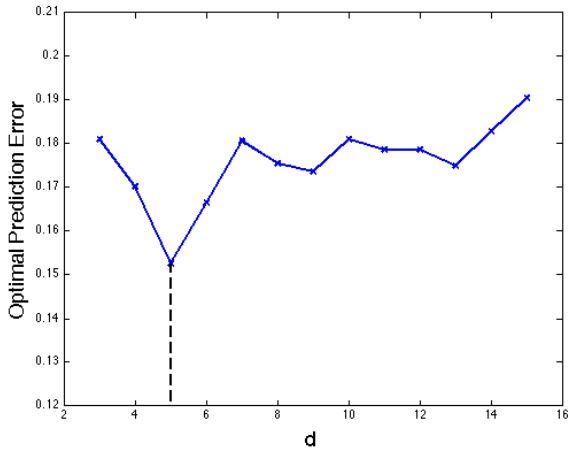



Figure 5: The optimal $d$ is found by minimizing the squared prediction error.

The effect of the regularization parameter $\alpha$ can be seen on the predictions below:

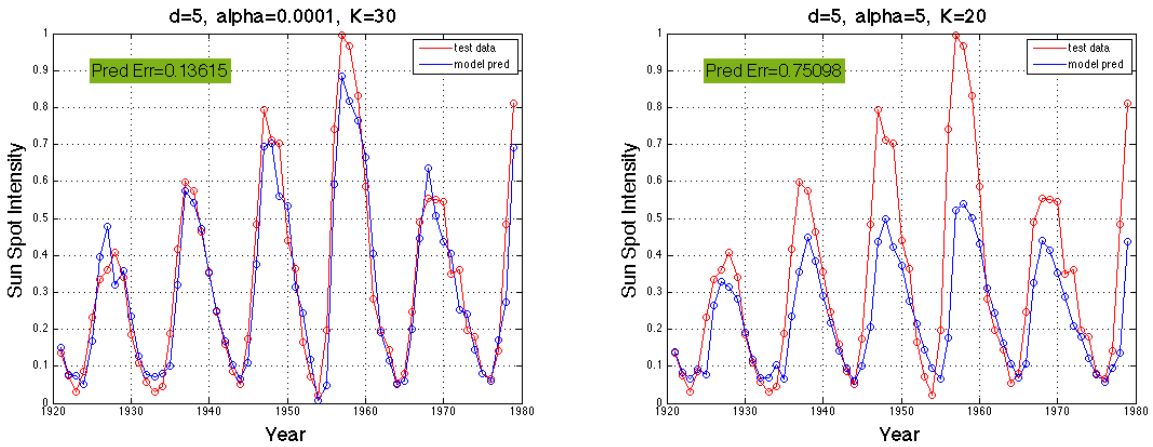
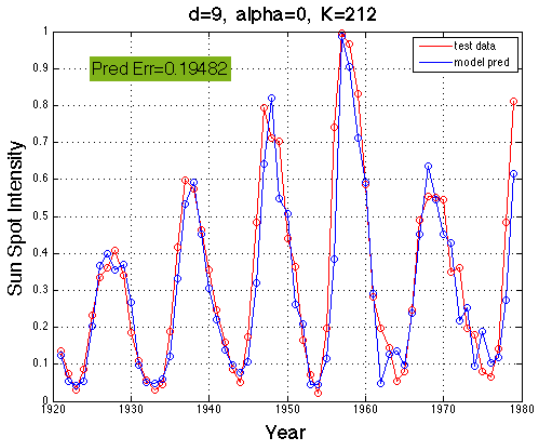
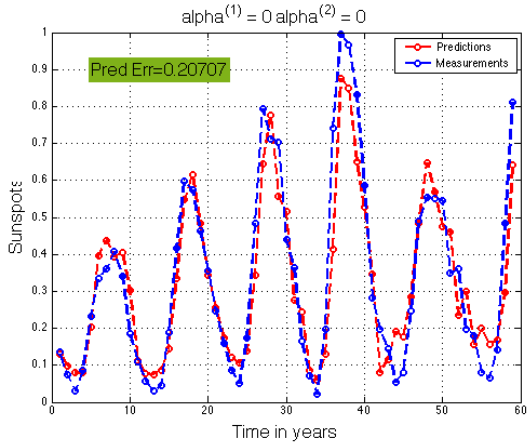



Figure 6: A large $\alpha$ reduces the models ability to make an accurate prediction at the peaks. In the fitting-a-plane analogy from previous exercises, this corresponds to restricting the slopes of the plane, forcing the predicted values to become smaller.

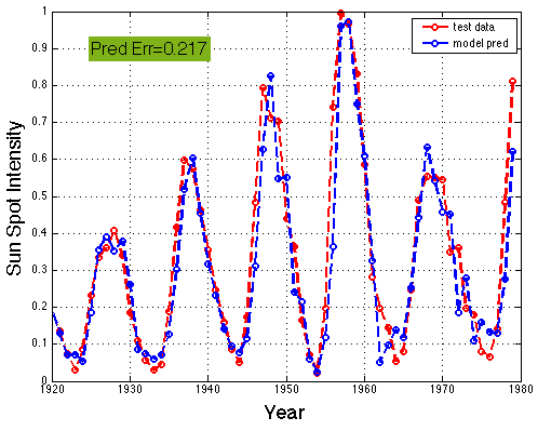
When $K$ is set to be equal to the size of the training data set, it corresponds to fitting one linear model to all the test data, as we did in exercise 4. This produces almost the exact same prediction as we did in **exercise 4b** with neural networks in **exercise 5b**.




(a) Linear regression from Exercise 4b.




(b) Neural Network solution.




(c) K-nearest-neighbors.

Figure 7