COURSE 02457

Non-Linear Signal Processing: Exercise 4

This exercise is based on C.M. Bishop: *Pattern Recognition and Machine Learning*, sections 1.1, 1.5.5, 3.1.1, 3.1.4, and 3.2. The objective of the exercise is to use the MATLAB software to illustrate and discuss the generalization concept for linear models.

Print and comment on the figures produced by the software main4a.m to main4c.m as outlined below at the three **Checkpoints**.

Linear Models

Let $y(\mathbf{x})$ be a function of the vector $\mathbf{x}$, where $\mathbf{x} = (x_1, \ldots, x_d)^\top$. To learn $y(\mathbf{x})$ we are given a data-set, $\mathcal{D} = \{(\mathbf{x}_n, t_n)\}$, $n = 1, \ldots, N$ of N corresponding values of input $(\mathbf{x})$ and output t.

In this exercise we assume $y(\mathbf{x})$ is a linear model

$$y(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i \quad = \quad w_0 + \mathbf{w}^\top \mathbf{x}, \tag{1}$$

where $\mathbf{w}$ is a weight vector.

The term w_0 in equation (1) can be included in the weight vector, $\mathbf{w}$, by modifying $\mathbf{x}$such that $\mathbf{x} = (1, x_1, \ldots, x_d)^\top$. This reduces equation (1) to

$$y(\mathbf{x}) = \sum_{i=0}^{d} w_i x_i \quad = \quad \mathbf{w}^\top \mathbf{x}. \tag{2}$$

The weight-vector, $\mathbf{w}$, for the given training-set can be estimated by minimizing an error function. Here we shall use the sum-of-squares error function augmented by a squared weight term corresponding to a Gaussian prior

$$\begin{aligned} E(\mathbf{w}) &= \frac{1}{2} \sum_{n=1}^{N} \{y(\mathbf{x}_n; \mathbf{w}) - t_n\}^2 + \frac{1}{2}\alpha \mathbf{w}^2 & (3) \\ &= \frac{1}{2} \sum_{n=1}^{N} \left\{\mathbf{w}^\top \mathbf{x}_n - t_n\right\}^2 + \frac{1}{2}\alpha \mathbf{w}^2. & (4) \end{aligned}$$

The parameter α is a control parameter, sometimes referred to as the "weight decay". Introducing the matrix, $\mathbf{X}$, where $\mathbf{X}^\top = (\mathbf{x}_1 \ \mathbf{x}_2 \ \ldots \mathbf{x}_N)$ and the vector, $\mathbf{t} = (t_1, t_2, \ldots, t_N)^\top$, equation (4) can be rewritten as

$$E(\mathbf{w}) = \frac{1}{2} \left(\mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w} + \mathbf{t}^\top \mathbf{t} - 2\mathbf{w}^\top \mathbf{X}^\top \mathbf{t} \right) + \frac{1}{2}\alpha \mathbf{w}^2. \tag{5}$$

Since equation (5) is quadratic in $\mathbf{w}$, the exact value of $\mathbf{w}$ minimizing $E(\mathbf{w})$ can be found analytically by equating the derivative of equation (5) to zero[1]. This gives the normal equations for the least-squares problem:

$$\left(\mathbf{X}^\top \mathbf{X} + \alpha \mathbf{1} \right) \mathbf{w} = \mathbf{X}^\top \mathbf{t}. \tag{6}$$

[1] The derivative of equation (5) can be performed using two rules $\frac{\partial \mathbf{z}^T \mathbf{B} \mathbf{z}}{\partial \mathbf{z}} = \left(\mathbf{B} + \mathbf{B}^T \right) \mathbf{z}$ and $\frac{\partial \mathbf{z}^T \mathbf{a}}{\partial \mathbf{z}} = \frac{\partial \mathbf{a}^T \mathbf{z}}{\partial \mathbf{z}} = \mathbf{a}$.

Where $\mathbf{1}$ is a unit matrix. Solving for $\mathbf{w}$ gives the optimal $\mathbf{w}$. Since $\mathbf{X}$ is an $N \times (d+1)$ matrix, $\mathbf{X}^\top\mathbf{X}$ is a $(d+1) \times (d+1)$ square matrix. Thus the solution to equation (6) is given by

$$\mathbf{w} = \left(\mathbf{X}^\top\mathbf{X} + \alpha\mathbf{1}\right)^{-1}\mathbf{X}^\top\mathbf{t}. \tag{7}$$

The generalization error is defined as the expectation

$$\begin{aligned} E_G(\mathbf{w}) &= \frac{1}{2}\int\int\{y(\mathbf{x};\mathbf{w}) - t\}^2 p(t|\mathbf{x})p(\mathbf{x})dtd\mathbf{x} & (8) \\ &\approx \frac{1}{2M}\sum_{m=1}^{M}\left\{\mathbf{w}^\top\mathbf{x}_m - t_m\right\}^2 & (9) \end{aligned}$$

approximated by the mean value over a large *test set* consisting of M examples drawn independently from the N examples in the training set.

Checkpoint 4.1:

Use the program main4a.m to create a training-set with a 2-dimensional input variable and a 1-dimensional output variable. Evaluate the training and test errors on independent sets generated by the same true weight vector and the same noise variance, for a model with one and two input variables respectively. In this checkpoint the weight decay is set to zero. Compare the training and test errors *per example* as function of the size of the training set. Compare the value of the training and test errors for large training sets with the value of the noise variance.

Time Series Prediction

An example where the linear model can be used is in time series prediction. To illustrate this, consider the example of the sunspot measurements. The number of sunspots oscillates almost periodically over a period of some years. The average number of sunspots has been measured yearly since 1700. Imagine we want to predict the average number of sunspots next year. The linear model can be used for this.

Let the number of sunspots in year n be x_n. Let's assume that the number of sunspots in year n only depends on the number of sunspots in the previous d years. This is reasonable since there must be a limit as to how far back one can expect a correlation. This can be expressed as

$$x_n = f(x_{n-1}, x_{n-2}, \dots x_{n-d}). \tag{10}$$

Approximating the function f with a linear model gives

$$x_n = w_0 + \sum_{j=1}^{d} w_j x_{n-j}. \tag{11}$$

This corresponds to equation (1), and hence is the same problem given by equations (2) to (7), where the training set is given by

$$\left.\begin{aligned} \mathbf{x}_n &= (1, x_{n-d}, \dots, x_{n-1})^\top \\ t_n &= x_n \end{aligned}\right\} \quad n = 1, \dots, N-d-1. \tag{12}$$

The weights can be found using equation (7), and the predicted value, x_{n+1}, can be found from

$$x_n = y(\mathbf{x}_n) = \mathbf{w}^\top \mathbf{x}_n. \tag{13}$$

In the context of sunspot time series prediction, the data set from 1700-1920 is used for training while the data from 1921-1979 is used to test performance.

Checkpoint 4.2:

Use the program main4b.m to perform a time series prediction of the number of sunspots with the data from 1700-1920 as training set. Evaluate the test error on the set 1921-1979. Normalize the test error per example by the total variance of the sunspot series. Study the test error as function of the number of weights, d, (hence years) included in the model. Which value of d do you recommend?

Bias-variance trade-off

The training set averages generalization error in the point $\mathbf{x}$ can be rewritten,

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[(y(\mathbf{x})) - \mathbb{E}_t[t|\mathbf{x}])^2\right] &= \mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x})]\}^2\right] \\ &+ \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x})] - \mathbb{E}_t[t|\mathbf{x}]\}^2.\end{aligned}$$

Where $\mathbb{E}_{\mathcal{D}}$ is the expectation with respect to training sets. Note $y(\mathbf{x}) = y(\mathbf{x}; \mathbf{w}(\mathcal{D}))$.

Hence the average error is split into a *variance* part, quantifying the variation among solutions for different training sets and a *bias* part quantifying the performance of the average model with respect to best possible model $\mathbb{E}_t[t|\mathbf{x}] = \int tp(t|\mathbf{x})dt$ (the conditional mean of the output given the input).

Checkpoint 4.3:

Use the program main4c.m to measure the relative amount of variance and bias for a linear model as in checkpoint 4.1 with two inputs and controlled by weight decay. Plot the average generalization error, the bias error, and the variance error for a large range of weight decay values. Comment on the two regimes where the generalization error stems from variance and bias respectively. What is the role of the weight decay in these two regimes. Which weight decay value would you recommend?

Challenge:

Consider a linear model and simulated data sets of various size N from a given (high dimensional) weight vector, say $d = 300$, with random input vectors and additive normal noise. Plot the learning curve (test error as function of sample size) for different values of the weight decay α.

DTU, September 2009,

Lars Kai Hansen, Karam Sidaros, and Carsten Stahlhut