Linear regression, active learning

We arrived at the logistic regression model when trying to explicitly model the uncertainty about the labels in a linear classifier. The same general modeling approach permits us to use linear predictions in various other contexts as well. The simplest of them is regression where the goal is to predict a continuous response $y_t \in \mathcal{R}$ to each example vector. Here too focusing on linear predictions won't be inherently limiting as linear predictions can be easily extended (next lecture).

So, how should we model continuous responses? The linear function of the input already produces a "mean prediction" or $\theta^T\mathbf{x} + \theta_0$. By treating this as a mean prediction more formally, we are stating that the expected value of the response variable, conditioned on $\mathbf{x}$, is $\theta^T\mathbf{x} + \theta_0$. More succinctly, we say that $E\{y|\mathbf{x}\} = \theta^T\mathbf{x} + \theta_0$. It remains to associate a distribution over the responses around such mean prediction. The simplest symmetric distribution is the normal (Gaussian) distribution. In other words, we say that the responses y follow the normal pdf

$$N(y;\, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y-\mu)^2 \right) \tag{1}$$

where $\mu = \theta^T\mathbf{x} + \theta_0$. Our response model is therefore defined as

$$P(y|\mathbf{x}, \theta, \theta_0) = N(y;\, \theta^T\mathbf{x} + \theta_0,\, \sigma^2) \tag{2}$$

So, when the input is 1-dimensional, we predict a mean response that is a line in (x, y) space, and assume that noise in y is normally distributed with zero mean and variance σ^2. Note that the noise variance σ^2 in the model does not depend on the input x. Moreover, we only model variation in the y-direction while expecting to know x with perfect precision. Taking into account the effect of potential noise in x on the responses y would tie parameters θ and θ_0 to the noise variance σ^2, potentially in an input dependent manner. The specifics of this coupling depend on the form of noise added to x. We will discuss this in a bit more detail later on.

We can also write the linear regression model in another way to explicate how exactly the additive noise appears in the responses:

$$y = \theta^T\mathbf{x} + \theta_0 + \epsilon \tag{3}$$

where $\epsilon \sim N(0, \sigma^2)$ (meaning that noise ϵ is distributed normally with mean zero and variance σ^2). Clearly for this model $E\{y|\mathbf{x}\} = \theta^T\mathbf{x} + \theta_0$ since ϵ has zero mean. Moreover, adding Gaussian noise to a deterministic prediction $\theta^T\mathbf{x} + \theta_0$ makes y normally distributed with mean $\theta^T \mathbf{x} + \theta_0$ and variance σ^2, as before. So, in particular, for the training inputs $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and outputs $y_1, \ldots, y_n$, the model relating them is

$$y_t = \theta^T \mathbf{x}_t + \theta_0 + \epsilon_t, \;\; t = 1, \ldots, n \tag{4}$$

where $e_t \sim N(0, \sigma^2)$ and e_i is independent of e_j for any $i \neq i$.

Regardless of how we choose the write the model (both forms are useful) we can find the parameter estimates by maximizing the conditional likelihood. Similarly to the logistic regression case, the conditional likelihood is written as

$$L(\theta, \theta_0, \sigma^2) = \prod_{t=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} (y_t - \theta^T \mathbf{x}_t - \theta_0)^2 \right) \tag{5}$$

Note that σ^2 is also a parameter we have to estimate. It accounts for errors not captured by the linear model. In terms of the log-likelihood, we try to maximize

$$\begin{aligned} l(\theta, \theta_0, \sigma^2) &= \sum_{t=1}^{n} \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} (y_t - \theta^T \mathbf{x}_t - \theta_0)^2 \right) \right] && (6) \\ &= \sum_{t=1}^{n} \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} \log \sigma^2 - \frac{1}{2\sigma^2} (y_t - \theta^T \mathbf{x}_t - \theta_0)^2 \right] && (7) \\ &= \text{const.} - \frac{n}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{t=1}^{n} (y_t - \theta^T \mathbf{x}_t - \theta_0)^2 && (8) \end{aligned}$$

where 'const.' absorbs terms that do not depend on the parameters. Now, the problem of estimating the parameters θ and θ_0 is nicely decoupled from estimating σ^2. In other words, we can find the *maximizing* $\hat{\theta}$ and $\hat{\theta}_0$ by simply *minimizing* the mean squared error

$$\sum_{t=1}^{n} (y_t - \theta^T \mathbf{x}_t - \theta_0)^2 \tag{9}$$

It is perhaps easiest to write the solution based on a bit of matrix calculation. Let $\mathbf{X}$ be a matrix whose rows, indexed by training examples, are given by $[\mathbf{x}_t^T, 1]$ ($\mathbf{x}_t$ turned into a row vector and 1 added at the end). In terms of this matrix, the minimization problem becomes

$$\begin{aligned}
\sum_{t=1}^{n}\left(y_t-\left[\begin{array}{c}\theta\\ \theta_0\end{array}\right]^T\left[\begin{array}{c}\mathbf{x}_t\\ 1\end{array}\right]\right)^2 &= \sum_{t=1}^{n}\left(y_t-[\mathbf{x}_t^T,1]\left[\begin{array}{c}\theta\\ \theta_0\end{array}\right]\right)^2 && (10)\\
&= \left\|\left[\begin{array}{c}y_1\\ \cdots\\ y_n\end{array}\right]-\left[\begin{array}{c}\mathbf{x}_1^T,1\\ \cdots\\ \mathbf{x}_n^T,1\end{array}\right]\left[\begin{array}{c}\theta\\ \theta_0\end{array}\right]\right\|^2 && (11)\\
&= \left\|\mathbf{y}-\mathbf{X}\left[\begin{array}{c}\theta\\ \theta_0\end{array}\right]\right\|^2 && (12)\\
&= \mathbf{y}^T\mathbf{y}-2\left[\begin{array}{c}\theta\\ \theta_0\end{array}\right]^T\mathbf{X}^T\mathbf{y}+\left[\begin{array}{c}\theta\\ \theta_0\end{array}\right]^T\mathbf{X}^T\mathbf{X}\left[\begin{array}{c}\theta\\ \theta_0\end{array}\right] && (13)
\end{aligned}$$

where $\mathbf{y}=[y_1,\ldots,y_n]^T$ is a vector of training responses. Solving it yields

$$\left[\begin{array}{c}\hat{\theta}\\ \hat{\theta}_0\end{array}\right]=(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \qquad (14)$$

Note that the optimal parameter values are linear functions of the observed responses $\mathbf{y}$. We will make use of this property later on. The dependence on the training inputs $\mathbf{x}_1,\ldots,\mathbf{x}_n$ (or the matrix $\mathbf{X}$) is non-linear, however.

The noise variance can be subsequently set to account for the remaining prediction errors. Indeed, the the maximizing value of σ^2 is given by

$$\hat{\sigma}^2=\frac{1}{n}\sum_{t=1}^{n}(y_t-\hat{\theta}^T\mathbf{x}_t-\hat{\theta}_0)^2 \qquad (15)$$

which is the average squared prediction error. Note that we cannot compute $\hat{\sigma}^2$ before knowing how well the linear model explains the responses.

Bias and variance of the parameter estimates

We can make use of the closed form parameter estimates in Eq.(14) to analyze how good these estimates are. For this purpose let's make the strong assumption that the actual relation between $\mathbf{x}$ and y follows a linear model of the same type that we are estimating (we just don't know the correct parameter values θ^*, θ_0^*, and σ^{*2}). We can therefore describe the observed responses y_t as

$$y_t=\theta^{*T}\mathbf{x}_t+\theta_0^*+\epsilon_t,\;\; t=1,\ldots,n \qquad (16)$$

where $\epsilon_t \sim N(0, \sigma^{*2})$. In a matrix form

$$\mathbf{y} = \mathbf{X} \begin{bmatrix} \theta^* \\ \theta_0^* \end{bmatrix} + \mathbf{e} \tag{17}$$

where $\mathbf{e} = [\epsilon_1, \ldots, \epsilon_n]^T$, $E\{\mathbf{e}\} = 0$ and $E\{\mathbf{e}\mathbf{e}^T\} = \sigma^{*2}\,\mathbf{I}$. The noise vector $\mathbf{e}$ is also independent of the inputs or $\mathbf{X}$. Plugging this form of responses into Eq.(14) we get

$$\begin{aligned} \begin{bmatrix} \hat{\theta} \\ \hat{\theta}_0 \end{bmatrix} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\begin{bmatrix} \theta^* \\ \theta_0^* \end{bmatrix} + \mathbf{e}) && (18) \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\begin{bmatrix} \theta^* \\ \theta_0^* \end{bmatrix} + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{e} && (19) \\ &= \begin{bmatrix} \theta^* \\ \theta_0^* \end{bmatrix} + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{e} && (20) \end{aligned}$$

In other words, our parameter estimates can be decomposed into the sum of correct underlying parameters and estimates based on noise alone (i.e., based on $\mathbf{e}$). Thus, on average with fixed inputs

$$E\{\begin{bmatrix} \hat{\theta} \\ \hat{\theta}_0 \end{bmatrix}|\mathbf{X}\} = \begin{bmatrix} \theta^* \\ \theta_0^* \end{bmatrix} + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T E\{\mathbf{e}|\mathbf{X}\} = \begin{bmatrix} \theta^* \\ \theta_0^* \end{bmatrix} \tag{21}$$

Our parameter estimates are therefore *unbiased* or correct on average when averaging is over possible training sets we could generate. The averaging here is conditioned on the specific training inputs.

Using Eq.(20) and Eq.(21) we can also evaluate the conditional co-variance of the parameter estimates where the expectation is again over the noise in the outputs:

$$\begin{aligned} Cov\{\begin{bmatrix} \hat{\theta} \\ \hat{\theta}_0 \end{bmatrix}|\mathbf{X}\} &= E\left\{\left(\begin{bmatrix} \hat{\theta} \\ \hat{\theta}_0 \end{bmatrix} - \begin{bmatrix} \theta^* \\ \theta_0^* \end{bmatrix}\right)\left(\begin{bmatrix} \hat{\theta} \\ \hat{\theta}_0 \end{bmatrix} - \begin{bmatrix} \theta^* \\ \theta_0^* \end{bmatrix}\right)^T |\mathbf{X}\right\} && (22) \\ &= E\left\{\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{e}\right)\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{e}\right)^T |\mathbf{X}\right\} && (23) \\ &= E\left\{(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{e}\mathbf{e}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}|\mathbf{X}\right\} && (24) \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\,E\{\mathbf{e}\mathbf{e}^T|\mathbf{X}\}\,\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} && (25) \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\,(\sigma^{*2}\,\mathbf{I})\,\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} && (26) \\ &= \sigma^{*2}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} && (27) \\ &= \sigma^{*2}(\mathbf{X}^T\mathbf{X})^{-1} && (28) \end{aligned}$$

So the way in which the parameters vary in response to noise in the outputs is a function of the inputs or $\mathbf{X}$. We will use this property in the next section to select inputs so as to improve the quality of the parameter estimates or to reduce the variance of predictions.

Based on the bias and variance calculations we can evaluate the mean squared error of the parameter estimates. To this end, we use the fact that the expectation of the squared norm of any vector valued random variable can be decomposed into a bias and variance components as follows:

$$\begin{aligned}
E\left\{\|\mathbf{z}-\mathbf{z}^*\|^2\right\} &= E\left\{\|\mathbf{z}-E\{\mathbf{z}\}+E\{\mathbf{z}\}-\mathbf{z}^*\|^2\right\} && (29)\\
&= E\left\{\|\mathbf{z}-E\{\mathbf{z}\}\|^2+2(\mathbf{z}-E\{\mathbf{z}\})^T(E\{\mathbf{z}\}-\mathbf{z}^*)+\|E\{\mathbf{z}\}-\mathbf{z}^*\|^2\right\} && (30)\\
&= E\left\{\|\mathbf{z}-E\{\mathbf{z}\}\|^2\right\}+2E\left\{(\mathbf{z}-E\{\mathbf{z}\})^T\right\}(E\{\mathbf{z}\}-\mathbf{z}^*)+\|E\{\mathbf{z}\}-\mathbf{z}^*\|^2 && \\
&= \overbrace{E\left\{\|\mathbf{z}-E\{\mathbf{z}\}\|^2\right\}}^{\text{variance}}+\overbrace{\|E\{\mathbf{z}\}-\mathbf{z}^*\|^2}^{\text{bias}^2} && (31)
\end{aligned}$$

where we have assumed that $\mathbf{z}^*$ is fixed. Make sure you understand how this decomposition is derived. We will further elaborate the variance part to better use the result in our context:

$$\begin{aligned}
E\left\{\|\mathbf{z}-E\{\mathbf{z}\}\|^2\right\} &= E\left\{(\mathbf{z}-E\{\mathbf{z}\})^T(\mathbf{z}-E\{\mathbf{z}\})\right\} && (32)\\
&= E\left\{Tr\left[(\mathbf{z}-E\{\mathbf{z}\})^T(\mathbf{z}-E\{\mathbf{z}\})\right]\right\} && (33)\\
&= E\left\{Tr\left[(\mathbf{z}-E\{\mathbf{z}\})(\mathbf{z}-E\{\mathbf{z}\})^T\right]\right\} && (34)\\
&= Tr\left[E\left\{(\mathbf{z}-E\{\mathbf{z}\})(\mathbf{z}-E\{\mathbf{z}\})^T\right\}\right] && (35)\\
&= Tr\left[Cov\{\mathbf{z}\}\right] && (36)
\end{aligned}$$

where $Tr[\cdot]$ is the matrix *trace*, the sum of its diagonal components, and therefore a linear operation (exchangeable with the expectation). We have also used the fact that $Tr[\mathbf{a}^T\mathbf{b}] = Tr[\mathbf{a}\mathbf{b}^T]$ for any vectors $\mathbf{a}$ and $\mathbf{b}$.

Now, adapting the result to our setting, we get

$$\begin{aligned}
E\left\{\left\|\left[\begin{array}{c}\hat{\theta}\\ \hat{\theta}_0\end{array}\right]-\left[\begin{array}{c}\theta^*\\ \theta_0^*\end{array}\right]\right\|^2 |\mathbf{X}\right\} &= \overbrace{Tr\left[Cov\left\{\left[\begin{array}{c}\hat{\theta}\\ \hat{\theta}_0\end{array}\right]|\mathbf{X}\right\}\right]}^{\text{variance}} + \overbrace{\left\|E\left\{\left[\begin{array}{c}\hat{\theta}\\ \hat{\theta}_0\end{array}\right]|\mathbf{X}\right\}-\left[\begin{array}{c}\theta^*\\ \theta_0^*\end{array}\right]\right\|^2}^{\text{bias}^2=0} \\
&= \sigma^{*2}\, Tr\left[(\mathbf{X}^T\mathbf{X})^{-1}\right] \qquad (37)
\end{aligned}$$

Let's understand this result a bit further. How does it depend on n, the number of training examples? In other words, how quickly does the mean squared error decrease as the number of training examples increases, assuming the input examples $\mathbf{x}$ are sampled independently from some underlying distribution $P(\mathbf{x})$? To answer this let's start by analyzing what happens to the matrix $\mathbf{X}^T\mathbf{X}$:

$$\begin{aligned}
\mathbf{X}^T\mathbf{X} &= \sum_{t=1}^{n}\left[\begin{array}{c}\mathbf{x}_t\\ 1\end{array}\right][\mathbf{x}_t^T, 1] \qquad (38)\\
&= n\cdot\frac{1}{n}\sum_{t=1}^{n}\left[\begin{array}{c}\mathbf{x}_t\\ 1\end{array}\right][\mathbf{x}_t^T, 1] \qquad (39)\\
&\approx n\cdot E_{\mathbf{x}\sim P}\left\{\left[\begin{array}{c}\mathbf{x}\\ 1\end{array}\right][\mathbf{x}^T, 1]\right\} = n\cdot\mathbf{C} \qquad (40)
\end{aligned}$$

where for large n the average will be close to the corresponding expected value. For large n the mean squared error of the parameter estimates will therefore be close to

$$\frac{\sigma^{*2}}{n}\cdot Tr[\mathbf{C}^{-1}] \qquad (41)$$

The variance of simply averaging the (noise in the) outputs would behave as σ^{*2}/n. Since we are estimating $d+1$ parameters where d is the input dimension, this dependence would have to be in $Tr[\mathbf{C}^{-1}]$. Indeed it is. This term, a trace of a $(d+1)\times(d+1)$ matrix $\mathbf{C}^{-1}$, is directly proportional to $d+1$.

Penalized log-likelihood and Ridge regression

When the number of training examples is small, i.e., not too much larger than the number of parameters (dimension of the inputs), it is often beneficial to *regularize* the parameter estimates. We will derive the form of regularization here by assigning a prior distribution over the parameters $P(\theta, \theta_0)$. The purpose of the prior is to prefer small parameter values (predict values close to zero) in the absence of data. Specifically, we will look at simple zero mean Gaussian distributions

$$P(\theta, \theta_0; \sigma'^2) = N(\begin{bmatrix} \theta \\ \theta_0 \end{bmatrix}; \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \sigma'^2 \mathbf{I}) = N(\theta_0; 0, \sigma'^2) \prod_{j=1}^{d} N(\theta_j; 0, \sigma'^2) \tag{42}$$

where the variance parameter σ'^2 in the prior distribution specifies how strongly we wish to bias the parameters towards zero.

By combining the log-likelihood criterion with the prior we obtain a *penalized log-likelihood* function (penalized by the prior):

$$\begin{aligned} l'(\theta, \theta_0, \sigma^2) &= \sum_{t=1}^{n} \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} (y_t - \theta^T \mathbf{x}_t - \theta_0)^2 \right) \right] + \log P(\theta, \theta_0; \sigma'^2) && (43) \\ &= \text{const.} - \frac{n}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{t=1}^{n} (y_t - \theta^T \mathbf{x}_t - \theta_0)^2 \\ &\quad - \frac{1}{2\sigma'^2} (\theta_0^2 + \sum_{j=1}^{d} \theta_j^2) - \frac{d+1}{2} \log \sigma'^2 && (44) \end{aligned}$$

It is convenient to tie the prior variance σ'^2 to the noise variance σ^2 according to $\sigma'^2 = \sigma^2/\lambda$. This has the effect that if the noise variance σ^2 is large, we penalize the parameters very little (permit large deviations from zero by assuming a large σ'^2). On the other hand, if the noise variance is small, we could be *over-fitting* the linear model. This happens, for example, when the number of training examples is small. In this case most of the responses can be explained directly by the linear model making the noise variance very small. In such cases our penalty for the parameters will be larger as well (prior variance is smaller).

Incorporating this parameter tie into the penalized log-likelihood function gives

$$\begin{aligned} l'(\theta, \theta_0, \sigma^2) &= \text{const.} - \frac{n}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{t=1}^{n} (y_t - \theta^T \mathbf{x}_t - \theta_0)^2 \\ &\quad - \frac{\lambda}{2\sigma^2} (\theta_0^2 + \sum_{j=1}^{d} \theta_j^2) - \frac{d+1}{2} \log(\sigma^2/\lambda) && (45) \\ &= \text{const.} - \frac{n+d+1}{2} \log \sigma^2 + \frac{d+1}{2} \log \lambda && (46) \\ &\quad - \frac{1}{2\sigma^2} \left[\sum_{t=1}^{n} (y_t - \theta^T \mathbf{x}_t - \theta_0)^2 + \lambda (\theta_0^2 + \sum_{j=1}^{d} \theta_j^2) \right] && (47) \end{aligned}$$

where again the estimation of θ and θ_0 separates from setting the noise variance σ^2. Note that this separation is achieved because we tied the prior and noise variance parameters. The above regularized problem of finding the parameter estimates $\hat{\theta}$ and $\hat{\theta}_0$ is known as *Ridge regression*.

As before, we can get closed form estimates for the parameters (we omit the analogous derivation):

$$\left[\begin{array}{c} \hat{\theta} \\ \hat{\theta}_0 \end{array} \right] = (\lambda\mathbf{I} + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{48}$$

It is now useful to understand how the properties of these parameter estimates depend on λ. For example, are the parameter estimates *unbiased*? No, they are not:

$$\begin{aligned} E\left\{ \left[\begin{array}{c} \hat{\theta} \\ \hat{\theta}_0 \end{array} \right] |\mathbf{X} \right\} &= (\lambda\mathbf{I} + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X} \left[\begin{array}{c} \theta^* \\ \theta_0^* \end{array} \right] && (49) \\ &= (\lambda\mathbf{I} + \mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I} - \lambda\mathbf{I}) \left[\begin{array}{c} \theta^* \\ \theta_0^* \end{array} \right] && (50) \\ &= \left[\begin{array}{c} \theta^* \\ \hat{\theta}_0^* \end{array} \right] \overbrace{-\lambda(\lambda\mathbf{I} + \mathbf{X}^T\mathbf{X})^{-1} \left[\begin{array}{c} \theta^* \\ \theta_0^* \end{array} \right]}^{\text{bias}} && (51) \\ &= \left(\mathbf{I} - \lambda(\lambda\mathbf{I} + \mathbf{X}^T\mathbf{X})^{-1}\right) \left[\begin{array}{c} \theta^* \\ \theta_0^* \end{array} \right] && (52) \end{aligned}$$

It is straightforward to check that $\left(\mathbf{I} - \lambda(\lambda\mathbf{I} + \mathbf{X}^T\mathbf{X})^{-1}\right)$ is a positive definite matrix with eigenvalues all less than one. The parameter estimates are therefore shrunk towards zero and more so the larger the value of λ. This is what we would expect since we explicitly favored small parameter values with the prior penalty. What do we gain from such *biased* parameter estimates? Let's evaluate the mean squared error, starting with the covariance:

$$\begin{aligned} Cov\left\{ \left[\begin{array}{c} \hat{\theta} \\ \hat{\theta}_0 \end{array} \right] |\mathbf{X} \right\} &= \sigma^{*2}(\lambda\mathbf{I} + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}(\lambda\mathbf{I} + \mathbf{X}^T\mathbf{X})^{-1} && (53) \\ &= \sigma^{*2}(\lambda\mathbf{I} + \mathbf{X}^T\mathbf{X})^{-1}(\lambda\mathbf{I} + \mathbf{X}^T\mathbf{X} - \lambda\mathbf{I})(\lambda\mathbf{I} + \mathbf{X}^T\mathbf{X})^{-1} && (54) \\ &= \sigma^{*2}(\lambda\mathbf{I} + \mathbf{X}^T\mathbf{X})^{-1} - \lambda\sigma^{*2}(\lambda\mathbf{I} + \mathbf{X}^T\mathbf{X})^{-2} && (55) \end{aligned}$$

The mean squared error in the parameters is therefore given by (we again omit the derivation that can be obtained similarly to previous expressions):

$$E\left\{\left\|\begin{bmatrix}\hat{\theta}\\ \hat{\theta}_0\end{bmatrix}-\begin{bmatrix}\theta^*\\ \theta^*_0\end{bmatrix}\right\|^2|\mathbf{X}\right\} = \sigma^{*2}\cdot Tr\left[(\lambda\mathbf{I}+\mathbf{X}^T\mathbf{X})^{-1}-\lambda(\lambda\mathbf{I}+\mathbf{X}^T\mathbf{X})^{-2}\right] +\lambda^2\begin{bmatrix}\theta^*\\ \theta^*_0\end{bmatrix}^T(\lambda\mathbf{I}+\mathbf{X}^T\mathbf{X})^{-2}\begin{bmatrix}\theta^*\\ \theta^*_0\end{bmatrix} \tag{56}$$

Can this be smaller than the mean squared error corresponding to the unregularized estimates $\sigma^{*2}\cdot Tr\left[(\mathbf{X}^T\mathbf{X})^{-1}\right]$? Yes, it can. This is indeed the benefit from regularization: we can reduce large variance at the cost of introducing a bit of bias. We will get back to this trade-off in the context of *model selection*.

Let's exemplify the effect of λ on the mean squared error in a context of a very simple 1-dimensional example. Suppose, we have observed responses for only two points, $x=-1$ and $x=1$. In this case,

$$\mathbf{X}=\begin{bmatrix}-1 & 1\\ 1 & 1\end{bmatrix},\ \mathbf{X}^T\mathbf{X}=\begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix},\ (\lambda\mathbf{I}+\mathbf{X}^T\mathbf{X})^{-1}=\begin{bmatrix}1/(2+\lambda) & 0\\ 0 & 1/(2+\lambda)\end{bmatrix} \tag{57}$$

The expression for the mean squared error therefore becomes

$$\begin{aligned}E\left\{\left\|\begin{bmatrix}\hat{\theta}\\ \hat{\theta}_0\end{bmatrix}-\begin{bmatrix}\theta^*\\ \theta^*_0\end{bmatrix}\right\|^2|\mathbf{X}\right\} &= \sigma^{*2}\left(\frac{2}{(2+\lambda)}-\frac{2\lambda}{(2+\lambda)^2}\right)+\frac{\lambda^2}{(2+\lambda)^2}(\theta^{*2}+\theta^{*2}_0)\\ &= \frac{4\sigma^{*2}}{(2+\lambda)^2}+\frac{\lambda^2}{(2+\lambda)^2}(\theta^{*2}+\theta^{*2}_0)\end{aligned} \tag{58}$$

We should compare this to $\sigma^{*2}Tr\left[(\mathbf{X}^T\mathbf{X})^{-1}\right]=\sigma^{*2}$ obtained without regularization (corresponds to setting $\lambda=0$). In the noisy case $\sigma^{*2}>\theta^{*2}+\theta^{*2}_0$ we can set $\lambda=2$ and obtain

$$E\left\{\left\|\begin{bmatrix}\hat{\theta}\\ \hat{\theta}_0\end{bmatrix}-\begin{bmatrix}\theta^*\\ \theta^*_0\end{bmatrix}\right\|^2|\mathbf{X}\right\}=\frac{4\sigma^{*2}}{16}+\frac{4}{16}(\theta^{*2}+\theta^{*2}_0)<\frac{8\sigma^{*2}}{16}=\frac{1}{2}\sigma^{*2} \tag{59}$$

The mean squared error of the parameters is therefore clearly smaller than without regularization.

Active learning

We can use the expressions for the mean squared error to actively select input points $\mathbf{x}_1,\ldots,\mathbf{x}_n$, when possible, so as to reduce the resulting estimation error. This is an *active* *learning* (*experiment design*) problem. The goal is to get by with as few responses as possible without sacrificing the estimation accuracy. For example, in training a classifier we would like to minimize the number of training images we would have to label in order obtain a certain classification accuracy. In the regression context, we may be selecting among possible experiments to carry out (e.g., choosing different operating points for a factory or gauging market/customer responses to different strategies available to us). By letting the method guide the selection of the training examples (inputs), we will generally need far fewer examples in comparison to selecting them at random from some underlying distribution, database, or trying available experiments at random. Fewer responses means less human effort, less cost, or both.

To develop this further let's go back to the unregularized case where

$$E\left\{ \left\| \begin{bmatrix} \hat{\theta} \\ \hat{\theta}_0 \end{bmatrix} - \begin{bmatrix} \theta^* \\ \theta_0^* \end{bmatrix} \right\|^2 |\mathbf{X} \right\} = \sigma^{*2} Tr\left[(\mathbf{X}^T\mathbf{X})^{-1} \right] \tag{60}$$

We do not know the noise variance σ^{*2} for the correct model but it only appears as a multiplicative constant in the above expression and therefore won't affect how we should choose the inputs. When the choice of inputs is indeed up to us (e.g., which experiments to carry out) we can select them so as to minimize $Tr\left[(\mathbf{X}^T\mathbf{X})^{-1} \right]$. One caveat of this approach is that it relies on the underlying relationship between the inputs and the responses to be linear. When this is no longer the case we may end up with clearly suboptimal selections.

We will discuss next time how we can find say n input examples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ that minimize the criterion.