Lecture topics:

- Active learning
- Non-linear predictions, kernels

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. 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.

To develop this further, recall that we continue to assume that the responses y come from some linear model $y = \theta^{*T}\mathbf{x} + \theta_0^* + \epsilon$ where $\epsilon \sim N(0, \sigma^{*2})$. Nothing is assumed about the distribution of $\mathbf{x}$ as the choice of the inputs is in our control. For any given set of inputs, $\mathbf{x}_1, \ldots, \mathbf{x}_n$, we derived last time an expression for the mean squared error of the maximum likelihood parameter estimates $\hat{\theta}$ and $\hat{\theta}_0$:

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

where the expectation is relative to the responses generated from the underlying linear model, i.e., over different training sets generated from the linear model. 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.

Given the selection criterion, how should we find say n input examples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ that minimize it? One simple approach is to select them one after the other, merely optimizing the selection of the next one in light of what we already have. Let's assume then that we already have $\mathbf{X}$ and thus have $\mathbf{A} = (\mathbf{X}^T\mathbf{X})^{-1}$ (assuming it is already invertible). We are trying to select another input example $\mathbf{x}$ that adds a row $[\mathbf{x}^T, 1]$ to $\mathbf{X}$. In an applied context we are typically constrained by what $\mathbf{x}$ can be (e.g., due to the experimental setup). We will discuss simple constraints below. Let's now evaluate the effect of adding a new (valid) row:

$$\left[\begin{array}{c} \mathbf{X} \\ \mathbf{x}^T\ 1 \end{array}\right]^T \left[\begin{array}{c} \mathbf{X} \\ \mathbf{x}^T\ 1 \end{array}\right] = (\mathbf{X}^T\mathbf{X}) + \left[\begin{array}{c} \mathbf{x} \\ 1 \end{array}\right]\left[\begin{array}{c} \mathbf{x} \\ 1 \end{array}\right]^T = \mathbf{A}^{-1} + \mathbf{v}\mathbf{v}^T \tag{2}$$

where $\mathbf{v} = [\mathbf{x}^T, 1]^T$. We would like to find a valid $\mathbf{v}$ that minimizes

$$Tr\left[(\mathbf{A}^{-1} + \mathbf{v}\mathbf{v}^T)^{-1}\right] \tag{3}$$

The matrix inverse can actually be carried out in closed form (easy enough to check)

$$(\mathbf{A}^{-1} + \mathbf{v}\mathbf{v}^T)^{-1} = \mathbf{A} - \frac{1}{(1 + \mathbf{v}^T\mathbf{A}\mathbf{v})}\mathbf{A}\mathbf{v}\mathbf{v}^T\mathbf{A} \tag{4}$$

so that the trace becomes

$$\begin{aligned} Tr\left[(\mathbf{A}^{-1} + \mathbf{v}\mathbf{v}^T)^{-1}\right] &= Tr\left[\mathbf{A}\right] - \frac{1}{(1 + \mathbf{v}^T\mathbf{A}\mathbf{v})}Tr\left[\mathbf{A}\mathbf{v}\mathbf{v}^T\mathbf{A}\right] && (5) \\ &= Tr\left[\mathbf{A}\right] - \frac{1}{(1 + \mathbf{v}^T\mathbf{A}\mathbf{v})}Tr\left[\mathbf{v}^T\mathbf{A}\mathbf{A}\mathbf{v}\right] && (6) \\ &= Tr\left[\mathbf{A}\right] - \frac{\mathbf{v}^T\mathbf{A}\mathbf{A}\mathbf{v}}{(1 + \mathbf{v}^T\mathbf{A}\mathbf{v})} && (7) \end{aligned}$$

Note that since $Tr[\mathbf{A}] = Tr[(\mathbf{X}^T\mathbf{X})^{-1}]$ is the mean squared error before adding the new example, any choice of $\mathbf{v}$, i.e., any additional example $\mathbf{x}$ will reduce the mean squared error. We are interested in finding the one that reduces the error the most. This is the example that maximizes

$$\frac{\mathbf{v}^T\mathbf{A}\mathbf{A}\mathbf{v}}{(1 + \mathbf{v}^T\mathbf{A}\mathbf{v})} \tag{8}$$

How much can we possibly reduce the squared error? The above term is bounded by the largest eigenvalue of $\mathbf{A}$. In other words, with each new example we can at most remove one degree of freedom from the parameter space. If we assume no constraints on the choice of $\mathbf{v}$, the maximizing vector would be of infinite length and proportional to the eigenvector of $\mathbf{A}$ with the largest eigenvalue (all the eigenvalues of $\mathbf{A}$ are positive as it is an inverse of a positive definite matrix $\mathbf{X}^T\mathbf{X}$). It is indeed advantageous in linear regression to have the input points as far from each other as possible (see Figure 1). If we constrain $\|\mathbf{v}\| \leq c$, then the maximizing $\mathbf{v}$ is the normalized eigenvector of $\mathbf{A}$ with the largest eigenvalue,

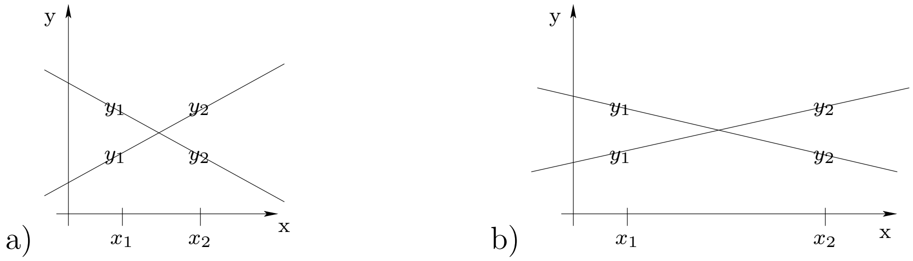

Figure 1: a) The effect of noise in the responses has a large effect on the parameters of the linear model when the corresponding inputs are close to each other; b) the effect is smaller when the inputs are further away.

normalized such that $\|\mathbf{v}\| = c$. Note that it may not be possible to select this eigenvector since $\mathbf{v} = [\mathbf{x}^T, 1]^T$. Other constraints on $\mathbf{x}$ will further restrict $\mathbf{v}$.

Let's take a simple example to illustrate the criterion. Suppose we have a 1-dimensional regression model $y = \theta x + \theta_0 + \epsilon$ where x is constrained to lie within $[-1, 1]$. Assume we have already observed responses for $x_1 = 1$ and $x_2 = -1$. Thus

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix},\ \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix},\ \mathbf{A} = (\mathbf{X}^T\mathbf{X})^{-1} = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{9}$$

$\mathbf{v} = [x, 1]^T$ and therefore $\mathbf{v}^T\mathbf{A}\mathbf{v} = (x^2+1)/2$ and $\mathbf{v}^T\mathbf{A}\mathbf{A}\mathbf{v} = (x^2+1)/4$. The criterion to be maximized becomes

$$\frac{\mathbf{v}^T\mathbf{A}\mathbf{A}\mathbf{v}}{(1+\mathbf{v}^T\mathbf{A}\mathbf{v})} = \frac{(x^2+1)/4}{1+(x^2+1)/2} \tag{10}$$

Since $z/(1+z)$ is an increasing function of z, it follows that the criterion is maximized when $(x^2+1)/2$ is maximized. Given the constraints, the maximizing point is $x = 1$ or $x = -1$. Either choice would do but, after selecting one, the other one would be preferred at the next step. The result is consistent with the intuition that for linear models the inputs should be as far from each other as possible (cf. Figure 1).

We have so far used the mean squared error in the parameters as the selection criterion. What about the variance in the predictions? Let's try to find the point $\mathbf{x}$ whose response we are the most uncertain about. We again write $\mathbf{v} = [\mathbf{x}^T, 1]^T$ so that

$$\begin{aligned}
Var\{y|\mathbf{x},\mathbf{X}\} &= E\left\{\left(\hat{\theta}^T\mathbf{x} + \hat{\theta}_0 - \theta^{*T}\mathbf{x} - \theta^{*}{}_0\right)^2 |\mathbf{x},\mathbf{X}\right\} && (11)\\
&= E\left\{\begin{bmatrix}\mathbf{x}\\1\end{bmatrix}^T \left(\begin{bmatrix}\hat{\theta}\\\hat{\theta}_0\end{bmatrix} - \begin{bmatrix}\theta^*\\\theta_0^*\end{bmatrix}\right)\left(\begin{bmatrix}\hat{\theta}\\\hat{\theta}_0\end{bmatrix} - \begin{bmatrix}\theta^*\\\theta_0^*\end{bmatrix}\right)^T \begin{bmatrix}\mathbf{x}\\1\end{bmatrix} |\mathbf{x},\mathbf{X}\right\} && (12)\\
&= \begin{bmatrix}\mathbf{x}\\1\end{bmatrix}^T \sigma^{*2}(\mathbf{X}^T\mathbf{X})^{-1}\begin{bmatrix}\mathbf{x}\\1\end{bmatrix} && (13)\\
&= \sigma^{*2}\cdot \mathbf{v}^T\mathbf{A}\mathbf{v} && (14)
\end{aligned}$$

where the expectation is over responses for existing training examples, again assuming that there is a correct underlying linear model. So the largest variance corresponds to the input $\mathbf{x}$ that maximizes $\mathbf{v}^T\mathbf{A}\mathbf{v}$ where $\mathbf{v} = [\mathbf{x}^T, 1]^T$. In the unconstrained case where there are few or no restrictions on $\mathbf{v}$, this maximizing point is exactly the one we would query according the previous selection criterion.

Non-linear predictions, kernels

Essentially all of what we have discussed can be extended to non-linear models, models that remain linear in the parameters as before but perform non-linear operations in the original input space. This is achieved by mapping the input examples to a higher dimensional feature space where the dimensions in the new feature vectors include non-linear functions of the inputs. The simplest setting for demonstrating this is linear regression in one dimension. Consider therefore the linear model $y = \theta x + \theta_0 + \epsilon$ where $\epsilon \sim N(0, \sigma^2)$. We can obtain a quadratic model by simply mapping the input x to a longer feature vector that includes a term quadratic in x. A third order model can be constructed by including all terms up to degree three, and so on. Explicitly, we would make linear predictions using feature vectors

$$\begin{aligned}
x &\xrightarrow{\phi} [1, \sqrt{2}x, x^2]^T = \phi(x) && (15)\\
x &\xrightarrow{\phi} [1, \sqrt{3}x, \sqrt{3}x^2, x^3]^T && (16)\\
&\cdots && (17)
\end{aligned}$$

The role of $\sqrt{2}$ and other constants will become clear shortly. The new polynomial regression model is then given by

$$y = \theta^T\phi(x) + \theta_0 + \epsilon, \;\; \epsilon \sim N(0, \sigma^2) \qquad (18)$$

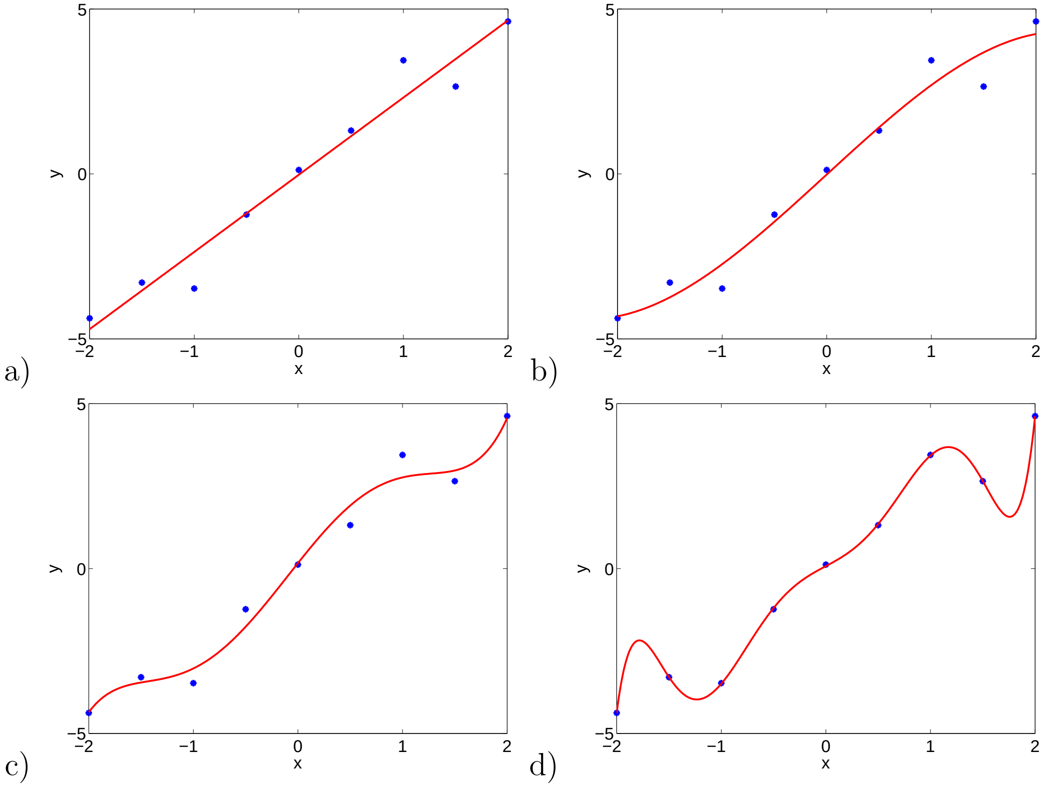

Figure 2: a) a linear model fit; b) a third order polynomial model fit to the same data; c) a fifth order polynomial model; d) a seventh order polynomial model.

where the dimensionality of $\phi(x)$ (and therefore also θ) depends on the order of the polynomial expansion. Regularization of the parameters is almost always used in conjunction with higher dimensional feature vectors. This is necessary since otherwise the higher order models could seriously *overfit* to the data. Examples of fitted polynomial regression models without regularization are shown in Figure 2a-d (the data were generated from a linear model). Note that all these models are linear in the parameters but non-linear in x, save the standard linear regression model in Figure 2a.

The polynomial expansion of input vectors works the same in higher dimensions, e.g.,

$$\mathbf{x} = [x_1, x_2]^T \xrightarrow{\phi} [1, x_1, x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2]^T = \phi(\mathbf{x}) \tag{19}$$

One limitation of explicating the feature vectors is that the dimensionality can increase rapidly with the degree of polynomial expansion, especially when the dimension of the input vector is already high. The table below gives some indicating of this though the effect is more dramatic with higher input dimensions. Can we somehow avoid explicating the dimensions of the feature vectors? In the context of models we have discussed thus far, yes, we can. We can define the feature vectors implicitly by focusing on specifying the values of their inner products or *kernel* instead. To get a sense of the power of this approach, let's evaluate the inner product between two feature vectors corresponding to the specific cubic expansions of 1-dimensional inputs shown before:

$$\begin{aligned} \phi(x) &= [1, \sqrt{3}x, \sqrt{3}x^2, x^3]^T, & (20) \\ \phi(x') &= [1, \sqrt{3}x', \sqrt{3}x'^2, x'^3]^T, & (21) \\ \phi(x)^T\phi(x') &= 1 + 3xx' + 3(xx')^2 + (xx')^3 = (1+xx')^3 & (22) \end{aligned}$$

So it seems we can compactly evaluate the inner products between polynomial feature vectors. The effect is more striking with higher dimensional inputs and higher polynomial degrees. (We did have to specify the constants appropriately in the feature vectors to make this work). To shift the modeling from explicit feature vectors to inner products (kernels) we obviously have to first turn the estimation problem into a form that involves only inner products between feature vectors.

$dim(\mathbf{x}) = 2$		$dim(\mathbf{x}) = 3$	
degree p	# of features	degree p	# of features
2	6	2	10
3	10	3	20
4	15	4	35
5	21	5	56

Linear regression and kernels

Let's simplify the model slightly by omitting the offset parameter θ_0, reducing the model to $y = \theta^T\phi(\mathbf{x}) + \epsilon$ where $\phi(\mathbf{x})$ is a particular feature expansion (e.g., polynomial). Our goal here is to turn both the estimation problem and the subsequent prediction task into forms that involve only inner products between the feature vectors.

We have already emphasized that regularization is necessary in conjunction with mapping examples to higher dimensional feature vectors. The regularized least squares objective to be minimized, with parameter λ, is given by

$$J(\theta) = \sum_{t=1}^{n} \left(y_t - \theta^T\phi(\mathbf{x}_t)\right)^2 + \lambda\|\theta\|^2 \qquad (23)$$

This form can be derived from penalized log-likelihood estimation (see previous lecture notes). The effect of the regularization penalty is to pull all the parameters towards zero. So any linear dimensions in the parameters that the training feature vectors do not pertain to are set explicitly to zero. We would therefore expect the optimal parameters to lie in the span of the feature vectors corresponding to the training examples. This is indeed the case.

We will continue with the derivation of the kernel (inner product) form of the linear regression model next time.