

Linear and kernel regression

How should we model continuous responses? The linear function of the input already produces a “mean prediction” or $\underline{\theta}^T \underline{x} + \theta_0$. By treating this as a mean prediction more formally, we are stating that the expected value of the response variable, conditioned on \underline{x} , is $\underline{\theta}^T \underline{x} + \theta_0$. More succinctly, we say that $E\{y|\underline{x}\} = \underline{\theta}^T \underline{x} + \theta_0$. 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

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

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

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

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 $\underline{\theta}$ 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:

$$y = \underline{\theta}^T \underline{x} + \theta_0 + \epsilon \quad (3)$$

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 $E\{y|\underline{x}\} = \underline{\theta}^T \underline{x} + \theta_0$ since ϵ has zero mean. Moreover, adding Gaussian noise to a deterministic prediction $\underline{\theta}^T \underline{x} + \theta_0$ makes y normally distributed with mean $\underline{\theta}^T \underline{x} + \theta_0$ and variance σ^2 , as before. So, in particular, for the training inputs $\underline{x}_1, \dots, \underline{x}_n$ and outputs y_1, \dots, y_n , the model relating them is

$$y_t = \underline{\theta}^T \underline{x}_t + \theta_0 + \epsilon_t, \quad t = 1, \dots, n \quad (4)$$

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

Regardless of how we choose to 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

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

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

$$l(\underline{\theta}, \theta_0, \sigma^2) = \sum_{t=1}^n \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y_t - \underline{\theta}^T \underline{x}_t - \theta_0)^2\right) \right] \quad (6)$$

$$= \sum_{t=1}^n \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} \log \sigma^2 - \frac{1}{2\sigma^2} (y_t - \underline{\theta}^T \underline{x}_t - \theta_0)^2 \right] \quad (7)$$

$$= \text{const.} - \frac{n}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{t=1}^n (y_t - \underline{\theta}^T \underline{x}_t - \theta_0)^2 \quad (8)$$

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

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

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 $[\underline{x}_t^T, 1]$ (\underline{x}_t turned into a row vector and 1 added at the end). In terms of this matrix, the minimization problem becomes

$$\sum_{t=1}^n \left(y_t - \begin{bmatrix} \underline{\theta} \\ \theta_0 \end{bmatrix}^T \begin{bmatrix} \underline{x}_t \\ 1 \end{bmatrix} \right)^2 = \sum_{t=1}^n \left(y_t - [\underline{x}_t^T, 1] \begin{bmatrix} \underline{\theta} \\ \theta_0 \end{bmatrix} \right)^2 \quad (10)$$

$$= \left\| \begin{bmatrix} y_1 \\ \dots \\ y_n \end{bmatrix} - \begin{bmatrix} \underline{x}_1^T, 1 \\ \dots \\ \underline{x}_n^T, 1 \end{bmatrix} \begin{bmatrix} \underline{\theta} \\ \theta_0 \end{bmatrix} \right\|^2 \quad (11)$$

$$= \left\| \mathbf{y} - \mathbf{X} \begin{bmatrix} \underline{\theta} \\ \theta_0 \end{bmatrix} \right\|^2 \quad (12)$$

$$= \mathbf{y}^T \mathbf{y} - 2 \begin{bmatrix} \underline{\theta} \\ \theta_0 \end{bmatrix}^T \mathbf{X}^T \mathbf{y} + \begin{bmatrix} \underline{\theta} \\ \theta_0 \end{bmatrix}^T \mathbf{X}^T \mathbf{X} \begin{bmatrix} \underline{\theta} \\ \theta_0 \end{bmatrix} \quad (13)$$

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

$$\begin{bmatrix} \hat{\theta} \\ \hat{\theta}_0 \end{bmatrix} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (14)$$

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 $\underline{x}_1, \dots, \underline{x}_n$ (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

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

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.

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 $P(\underline{\theta}, \theta_0)$. 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

$$P(\underline{\theta}, \theta_0; \sigma'^2) = N\left(\begin{bmatrix} \underline{\theta} \\ \theta_0 \end{bmatrix}; \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \sigma'^2 \mathbf{I}\right) = N(\theta_0; 0, \sigma'^2) \prod_{j=1}^d N(\theta_j; 0, \sigma'^2) \quad (16)$$

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):

$$\begin{aligned}
 l'(\underline{\theta}, \theta_0, \sigma^2) &= \sum_{t=1}^n \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp \left(-\frac{1}{2\sigma^2} (y_t - \underline{\theta}^T \underline{x}_t - \theta_0)^2 \right) \right] + \log P(\underline{\theta}, \theta_0; \sigma'^2) \quad (17) \\
 &= \text{const.} - \frac{n}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{t=1}^n (y_t - \underline{\theta}^T \underline{x}_t - \theta_0)^2 \\
 &\quad - \frac{1}{2\sigma'^2} (\theta_0^2 + \sum_{j=1}^d \theta_j^2) - \frac{d+1}{2} \log \sigma'^2 \quad (18)
 \end{aligned}$$

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

$$\begin{aligned}
 l'(\underline{\theta}, \theta_0, \sigma^2) &= \text{const.} - \frac{n}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{t=1}^n (y_t - \underline{\theta}^T \underline{x}_t - \theta_0)^2 \\
 &\quad - \frac{\lambda}{2\sigma^2} (\theta_0^2 + \sum_{j=1}^d \theta_j^2) - \frac{d+1}{2} \log(\sigma^2/\lambda) \quad (19)
 \end{aligned}$$

$$= \text{const.} - \frac{n+d+1}{2} \log \sigma^2 + \frac{d+1}{2} \log \lambda \quad (20)$$

$$- \frac{1}{2\sigma^2} \left[\sum_{t=1}^n (y_t - \underline{\theta}^T \underline{x}_t - \theta_0)^2 + \lambda (\theta_0^2 + \sum_{j=1}^d \theta_j^2) \right] \quad (21)$$

where again the estimation of $\underline{\theta}$ 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 $\hat{\underline{\theta}}$ 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):

$$\begin{bmatrix} \hat{\underline{\theta}} \\ \hat{\theta}_0 \end{bmatrix} = (\lambda \mathbf{I} + \mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (22)$$

Linear regression and kernels

Consider a slightly simpler model where we omit the offset parameter θ_0 , reducing the model to $y = \underline{\theta}^T \underline{\phi}(x) + \epsilon$ where $\underline{\phi}(x)$ 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

$$J(\underline{\theta}) = \frac{1}{2} \sum_{t=1}^n (y_t - \underline{\theta}^T \underline{\phi}(x_t))^2 + \frac{\lambda}{2} \|\underline{\theta}\|^2 \quad (23)$$

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.

As before, the optimality condition for $\underline{\theta}$ follows from setting the gradient to zero:

$$\frac{dJ(\underline{\theta})}{d\underline{\theta}} = - \sum_{t=1}^n \overbrace{(y_t - \underline{\theta}^T \underline{\phi}(x_t))}^{\alpha_t} \underline{\phi}(x_t) + \lambda \underline{\theta} = 0 \quad (24)$$

We can therefore construct the optimal $\underline{\theta}$ in terms of prediction differences α_t and the feature vectors:

$$\underline{\theta} = \frac{1}{\lambda} \sum_{t=1}^n \alpha_t \underline{\phi}(x_t) \quad (25)$$

The implication is that the optimal $\underline{\theta}$ (however high dimensional) will lie in the span of the feature vectors corresponding to the training examples. This is due to the regularization penalty we added. But how do we set α_t ? The values for α_t can be found by insisting that they indeed can be interpreted as prediction differences:

$$\alpha_t = y_t - \underline{\theta}^T \underline{\phi}(x_t) = y_t - \frac{1}{\lambda} \sum_{t'=1}^n \alpha_{t'} \underline{\phi}(x_{t'})^T \underline{\phi}(x_t) \quad (26)$$

Thus α_t depends only on the actual responses y_t and the inner products between the training examples, the *Gram matrix*:

$$\mathbf{K} = \begin{bmatrix} \phi(\underline{x}_1)^T \phi(\underline{x}_1) & \cdots & \phi(\underline{x}_1)^T \phi(\underline{x}_n) \\ \vdots & \ddots & \vdots \\ \phi(\underline{x}_n)^T \phi(\underline{x}_1) & \cdots & \phi(\underline{x}_n)^T \phi(\underline{x}_n) \end{bmatrix} \quad (27)$$

In a vector form,

$$\underline{a} = [\alpha_1, \dots, \alpha_n]^T, \quad (28)$$

$$\underline{y} = [y_1, \dots, y_n]^T, \quad (29)$$

$$\underline{a} = \underline{y} - \frac{1}{\lambda} \mathbf{K} \underline{a} \quad (30)$$

the solution is

$$\hat{\underline{a}} = \lambda \left(\lambda \mathbf{I} + \mathbf{K} \right)^{-1} \underline{y} \quad (31)$$

Note that finding the estimates $\hat{\alpha}_t$ requires inverting a $n \times n$ matrix. This is the cost of dealing with inner products as opposed to handing feature vectors directly. In some cases, the benefit is substantial since the feature vectors in the inner products may be infinite dimensional but never needed explicitly.

As a result of finding $\hat{\alpha}_t$ we can cast the predictions for new examples also in terms of inner products:

$$\underline{y} = \hat{\underline{\theta}}^T \phi(\underline{x}) = \sum_{t=1}^n (\hat{\alpha}_t / \lambda) \phi(\underline{x}_t)^T \phi(\underline{x}) = \sum_{t=1}^n (\hat{\alpha}_t / \lambda) K(\underline{x}_t, \underline{x}) \quad (32)$$

where we view $K(\underline{x}_t, \underline{x})$ as a *kernel function*, a function of two arguments \underline{x}_t and \underline{x} .

Appendix (optional): Kernel linear regression with offset

Given a feature expansion specified by $\phi(\underline{x})$ we try to minimize

$$J(\underline{\theta}, \theta_0) = \sum_{t=1}^n (y_t - \underline{\theta}^T \phi(\underline{x}_t) - \theta_0)^2 + \lambda \|\underline{\theta}\|^2 \quad (33)$$

where we have chosen *not* to regularize θ_0 to preserve the similarity to classification discussed later on. Not regularizing θ_0 means, e.g., that we do not care whether all the

responses have a constant added to them; the value of the objective, after optimizing θ_0 , would remain the same with or without such constant.

Setting the derivatives with respect to θ_0 and $\underline{\theta}$ to zero gives the following optimality conditions:

$$\frac{dJ(\underline{\theta}, \theta_0)}{d\theta_0} = -2 \sum_{t=1}^n (y_t - \underline{\theta}^T \phi(\mathbf{x}_t) - \theta_0) = 0 \quad (34)$$

$$\frac{dJ(\underline{\theta}, \theta_0)}{d\underline{\theta}} = 2\lambda \underline{\theta} - 2 \sum_{t=1}^n \overbrace{(y_t - \underline{\theta}^T \phi(\mathbf{x}_t) - \theta_0)}^{\alpha_t} \phi(\mathbf{x}_t) = 0 \quad (35)$$

We can therefore construct the optimal $\underline{\theta}$ in terms of prediction differences α_t and the feature vectors as before:

$$\underline{\theta} = \frac{1}{\lambda} \sum_{t=1}^n \alpha_t \phi(\mathbf{x}_t) \quad (36)$$

Using this form of the solution for $\underline{\theta}$ and Eq.(34) we can also express the optimal θ_0 as a function of the prediction differences α_t :

$$\theta_0 = \frac{1}{n} \sum_{t=1}^n (y_t - \underline{\theta}^T \phi(\mathbf{x}_t)) = \frac{1}{n} \sum_{t=1}^n \left(y_t - \frac{1}{\lambda} \sum_{t'=1}^n \alpha_{t'} \phi(\mathbf{x}_{t'})^T \phi(\mathbf{x}_t) \right) \quad (37)$$

We can now constrain α_t to take on values that can indeed be interpreted as prediction differences:

$$\alpha_i = y_i - \underline{\theta}^T \phi(\mathbf{x}_i) - \theta_0 \quad (38)$$

$$= y_i - \frac{1}{\lambda} \sum_{t'=1}^n \alpha_{t'} \phi(\mathbf{x}_{t'})^T \phi(\mathbf{x}_i) - \theta_0 \quad (39)$$

$$= y_i - \frac{1}{\lambda} \sum_{t'=1}^n \alpha_{t'} \phi(\mathbf{x}_{t'})^T \phi(\mathbf{x}_i) - \frac{1}{n} \sum_{t=1}^n \left(y_t - \frac{1}{\lambda} \sum_{t'=1}^n \alpha_{t'} \phi(\mathbf{x}_{t'})^T \phi(\mathbf{x}_t) \right) \quad (40)$$

$$= y_i - \frac{1}{n} \sum_{t=1}^n y_t - \frac{1}{\lambda} \sum_{t'=1}^n \alpha_{t'} \left(\phi(\mathbf{x}_{t'})^T \phi(\mathbf{x}_i) - \frac{1}{n} \sum_{t=1}^n \phi(\mathbf{x}_{t'})^T \phi(\mathbf{x}_t) \right) \quad (41)$$

With the same matrix notation as before, and letting $\mathbf{1} = [1, \dots, 1]^T$, we can rewrite the above condition as

$$\underline{a} = \overbrace{(\mathbf{I} - \mathbf{1}\mathbf{1}^T/n)}^C \mathbf{y} - \frac{1}{\lambda} (\mathbf{I} - \mathbf{1}\mathbf{1}^T/n) \mathbf{K} \underline{a} \quad (42)$$

where $C = \mathbf{I} - \mathbf{1}\mathbf{1}^T/n$ is a *centering* matrix. Any solution to the above equation has to satisfy $\mathbf{1}^T \underline{a} = 0$ (just left multiply the equation with $\mathbf{1}^T$). Note that this is exactly the optimality condition for θ_0 in Eq.(34). Using this “summing to zero” property of the solution we can rewrite the above equation as

$$\underline{a} = C\mathbf{y} - \frac{1}{\lambda} C\mathbf{K}C\underline{a} \quad (43)$$

where we have introduced an additional centering operation on the right hand side. This cannot change the solution since $C\underline{a} = \underline{a}$ whenever $\mathbf{1}^T \underline{a} = 0$. The solution $\hat{\underline{a}}$ is then

$$\hat{\underline{a}} = \lambda(\lambda\mathbf{I} + C\mathbf{K}C)^{-1} C\mathbf{y} \quad (44)$$

Once we have $\hat{\underline{a}}$ we can reconstruct $\hat{\theta}_0$ from Eq.(37). $\hat{\theta}^T \phi(x)$ reduces to the kernel form as before.