Kernels and Regularized Learning

Ying Nian Wu and Quanshi Zhang

Contents

1 Regularized learning

1.1 Over-fitting & under-fitting

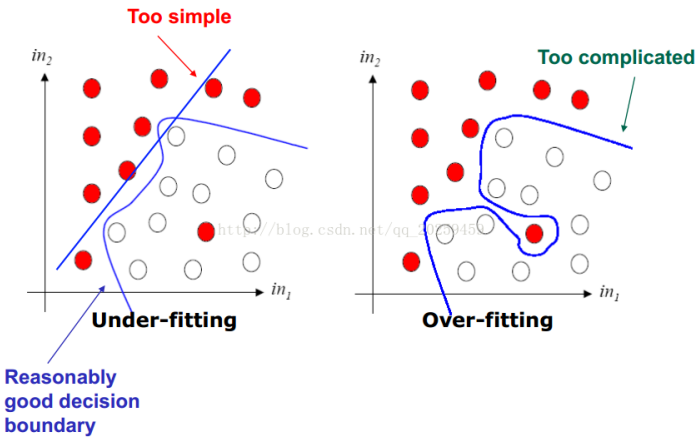

Figure 1: Over-fitting and under-fitting. `https://blog.csdn.net/qq_20259459/article/details/70316511`

credit: `https://en.wikipedia.org/wiki/Overfitting`

Overfitting is "the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably." An overfitted model is a statistical model that contains more parameters than can be justified by the data. The essence of overfitting is to have unknowingly extracted some of the residual variation (*i.e.* the noise) as if that variation represented underlying model structure.

- Well fit to training samples, but badly fit to testing samples
- The model usually has much more parameters than necessary.
- The model considers feature noises as meaningful information.

training set	validation set	Performance
error too large	irrelevant	Underfitting
error small	error too large	Overfitting
error small	error small	Ideal (Good generalization)

Figure 2: Over-fitting and under-fitting. `https://blog.csdn.net/qq_20259459/article/details/70316511`

Underfitting occurs when a statistical model cannot adequately capture the underlying structure of the data. An underfitted model is a model where some parameters or terms that would appear in a correctly specified model are missing. Underfitting would occur, for example, when fitting a linear model to non-linear data. Such a model will tend to have poor predictive performance.

- Badly fit to both training samples and testing samples.
- The model is not flexible enough (*e.g.* parameters in the model are not enough), or the optimization method is not powerful enough.
- The model misses meaningful information.

1.2 Ridge Regression

Consider the training data (Y, X), where X is $n \times p$. The ridge regression estimates β by minimizing $\|Y - X\beta\|^2 + \lambda\|\beta\|^2$ for a tuning parameter $\lambda > 0$. The $\lambda\|\beta\|^2$ is a penalty or regularization term.

Let $\ell(\beta)$ denote the loss function of ridge regression

$$\ell(\beta) = \|Y - X\beta\|^2 + \lambda\|\beta\|^2.$$

We want to find $\hat{\beta}_\lambda$ with minimal loss

$$\hat{\beta}_\lambda = \arg\min_\beta \ell(\beta).$$

By taking the derivative of the loss function w.r.t. β, we have

$$\ell(\beta) = \|Y - X\beta\|^2 + \lambda\|\beta\|^2 = (Y - X\beta)^\top (Y - X\beta) + \lambda\beta^\top\beta.$$

$$0 = \frac{\partial \ell(\beta)}{\partial \beta} \mid_{\beta=\hat{\beta}_\lambda} = -2X^\top (Y - X\hat{\beta}_\lambda) + 2\lambda\hat{\beta}_\lambda.$$

By convexity of the loss function, the stationary point gives the minimum. Hence

$$\hat{\beta}_\lambda = (X^\top X + \lambda I_p)^{-1} X^\top Y.$$

The resulting estimator $\hat{\beta}_\lambda$ is called shrinkage estimator. Figure 3 depicts how λ controls the amount of shrinkage.

The regularization term $\lambda\|\beta\|^2$ penalizes the weight dimensions with large absolute values, which prevents the regression from being conducted based on a few feature dimensions and boosts the robustness of the model.

For example, if

$$\beta = [0.01, 0.03, 1.80, 2.41, 0.02]^\top$$

then the model mainly uses the third and fourth dimensions of the feature for regression; if

$$\beta = [0.71, 0.92, 1.20, 1.42, 0.83]^\top$$

then the model uses all dimension of the feature for regression.
If we choose $\lambda = 0$, then β is not 'restricted'. If we let $\lambda \to \infty$, then $\beta \to 0$.

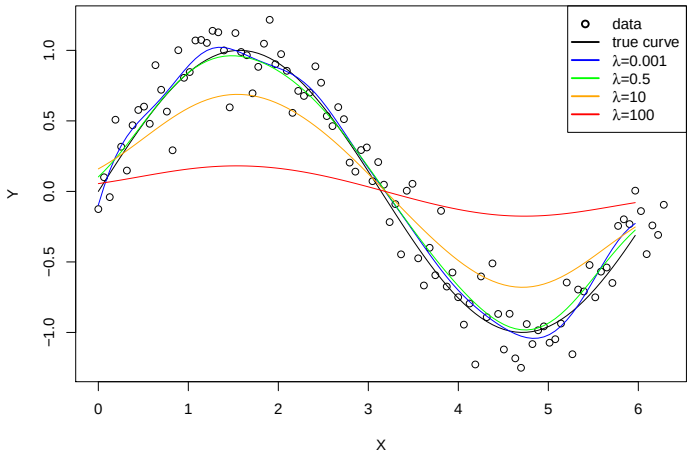

Figure 3: Ridge regression with varying λ where $y_i = \sin x + \epsilon_i$ and $\epsilon_i \sim N(0, \sigma^2)$. Each sample x_i locates in a p-dimensional space, and $X\beta$ is still a linear regression.

1.3 Kernel Regression

Sometimes, we need to use $\phi(x)$ instead of x as features to deal with the regression problem.

$$\begin{aligned} f(x) &= \sum_{i=1}^{n} c_i K(x, x_i) \\ &= \sum_{i=1}^{n} c_i \langle \phi(x), \phi(x_i) \rangle \\ &= \langle \phi(x), \sum_{i=1}^{n} c_i \phi(x_i) \rangle \end{aligned}$$

where we define

$$K(x, x_i) \stackrel{\text{def}}{=} \phi(x)^\top \phi(x_i)$$

as the **Kernel**.

Suppose we have training examples (y_i, x_i), $i = 1, ..., n$. Consider the Gaussian kernel $K(x, x') = e^{-\gamma \|x - x'\|^2}$. Suppose we want to learn a regression curve of the form

$$f(x) = \sum_{i=1}^{n} c_i K(x, x_i)$$

by minimizing

$$\sum_{i=1}^{n} ||y_i - \sum_{j=1}^{n} c_j K(x_i, x_j)||^2 + \lambda \sum_{i,j} c_i c_j K(x_i, x_j).$$

Let kernel K be an $n \times n$ matrix with $K_{ij} = K(x_i, x_j)$, then the objective function in matrix notation is

$$\ell(c) = \sum_{i=1}^{n} \|y_i - \sum_{j=1}^{n} c_j K_{ij}\|^2 + \lambda \sum_{i,j} c_i c_j K_{ij} = \|Y - Kc\|^2 + \lambda c^\top K c.$$

We want to find $\hat{c}_\lambda$ with minimal loss

$$\hat{c}_\lambda = \arg\min_c \ell(c).$$

By taking the derivative of the loss function w.r.t. c, we get the first order condition

$$\ell(c) = \|Y - Kc\|^2 + \lambda c^\top K c = (Y - Kc)^\top (Y - Kc) + \lambda c^\top K c$$

$$0 = \frac{\partial \ell(c)}{\partial c} \mid_{c=\hat{c}_\lambda} = -2K^\top (Y - K\hat{c}_\lambda) + 2\lambda K \hat{c}_\lambda.$$

Note $K = K^T$ by definition of $K(\cdot, \cdot)$. Hence, the estimate of c is

$$\begin{aligned}
& 0 = -2K^\top (Y - K\hat{c}_\lambda) + 2\lambda K \hat{c}_\lambda \\
\Rightarrow \quad & K^\top K \hat{c}_\lambda + \lambda K \hat{c}_\lambda = K^\top Y \\
\Rightarrow \quad & (K^\top)^{-1} \left[K^\top K \hat{c}_\lambda + \lambda K \hat{c}_\lambda\right] = (K^\top)^{-1} K^\top Y \\
\Rightarrow \quad & (K^\top)^{-1} K^\top K \hat{c}_\lambda + \lambda (K^\top)^{-1} K \hat{c}_\lambda = (K^\top)^{-1} K^\top Y \\
\Rightarrow \quad & K \hat{c}_\lambda + \lambda I_n \hat{c}_\lambda = Y \qquad \text{because for kernel } K = K^\top, \text{ so } (K^\top)^{-1} K = (K^\top)^{-1} K^\top = I_n \\
\Rightarrow \quad & \hat{c}_\lambda = (K + \lambda I_n)^{-1} Y.
\end{aligned}$$

The regularization term $\lambda c^\top K c$ penalizes the weight for $\phi(x)$ with large absolute values, which prevents the regression from being conducted based on a few feature dimensions and boosts the robustness of the model.

Understanding the kernel regression from the perspective of the Ridge regression: If we set $\beta = \sum_i c_i \phi(x_i)$, then

$$f(x) = \sum_{i=1}^{n} c_i K(x, x_i) = \langle \phi(x), \sum_{i=1}^{n} c_i \phi(x_i) \rangle = \langle \phi(x), \beta \rangle$$

$$\begin{aligned}
c^\top K c &= \sum_{i,j} c_i c_j K(x_i, x_j) \\
&= \sum_{i,j} c_i c_j \phi(x_i)^\top \phi(x_j) \\
&= (\sum_i c_i \phi(x_i))^\top (\sum_i c_i \phi(x_i)) \\
&= \|\beta\|^2
\end{aligned}$$

For example, if

$$\sum_i c_i \phi(x_i) = \beta = [0.01, 0.03, 1.80, 2.41, 0.02]^\top$$

then the model mainly uses the third and fourth dimensions of $\phi(x)$ for regression; if

$$\sum_i c_i \phi(x_i) = \beta = [0.71, 0.92, 1.20, 1.42, 0.83]^\top$$

then the model uses all dimension of $\phi(x)$ for regression.

1.4 Spline Regression

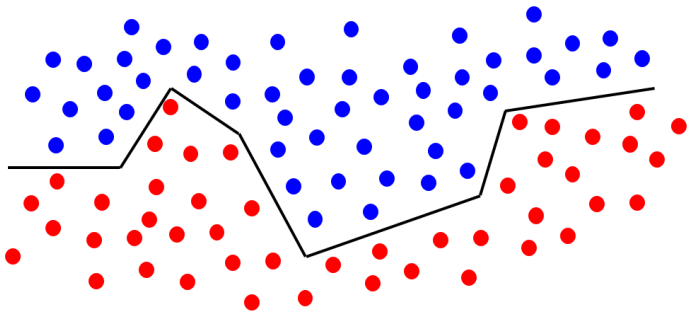

Figure 4: Spline classification.

Suppose the training examples are (y_i, x_i), $i = 1, \ldots, n$ where x_i is one dimensional. Suppose we have a set of knots $k_j, j = 1, \ldots, p$. Suppose we fit a linear spline of the form $f(x) = \alpha_0 + \sum_{j=1}^p \alpha_j \max(0, x - k_j)$ by minimizing

$$\sum_{i=1}^n \|y_i - \alpha_0 - \sum_{j=1}^p \alpha_j \max(0, x_i - k_j)\|^2 + \lambda \sum_{j=1}^p \alpha_j^2.$$

Thus,

$$y_i - \alpha_0 - \sum_{j=1}^p \alpha_j \max(0, x_i - k_j) = y_i - [1, \max(0, x_i - k_1), \max(0, x_i - k_2), \ldots, \max(0, x_i - k_p)]\alpha$$

where $\alpha = [\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_p]^\top$.

Relations to the Ridge regression

Let

$$\begin{aligned} \widetilde{X}_{ij} &= \max(0, x_i - k_j) \\ Z &= [1_n \ \widetilde{X}] \qquad \text{is a } n \times (p+1) \text{ matrix.} \\ D &= \text{diag}(0, 1, \ldots, 1) \qquad \text{is a } (p+1) \times (p+1) \text{ matrix.} \end{aligned}$$

then the objective function can be written as

$$\ell(\alpha) = \|Y - Z\alpha\|^2 + \lambda \|D\alpha\|^2.$$

Homework: Please provide the process to derive the above loss function $\ell(\alpha) = \|Y - Z\alpha\|^2 + \lambda\|D\alpha\|^2$ based on the initial loss function $\sum_{i=1}^{n}\|y_i - \alpha_0 - \sum_{j=1}^{p}\alpha_j \max(0, x_i - k_j)\|^2 + \lambda\sum_{j=1}^{p}\alpha_j^2$.

We want to find $\hat{\alpha}_\lambda$ with minimal loss

$$\hat{\alpha}_\lambda = \arg\min_{\alpha} \ell(\alpha).$$

By taking the derivative of the loss function w.r.t. α, we have

$$\frac{\partial \ell(\alpha)}{\partial \alpha} |_{\alpha=\hat{\alpha}_\lambda} = -2Z^\top(Y - Z\hat{\alpha}_\lambda) + 2\lambda D\hat{\alpha}_\lambda = 0. \qquad \text{here, } D = D^\top D$$

Hence, the estimate of α is

$$\hat{\alpha}_\lambda = (Z^\top Z + \lambda D)^{-1} Z^\top Y.$$

Relations to the kernel regression

Suppose we let the knots be $x_j, j = 1, \ldots, n$, and let $K(x_i, x_j) = \max(0, x_i - x_j)$, then the spline regression coincides with a kernel regression (up to a constant term)

$$\hat{f}(x) = \sum_{j=1}^{n} \hat{\alpha}_j \langle x, x_j \rangle = \sum_{j=1}^{n} \hat{\alpha}_j K(x, x_j), \qquad \text{for spline regressin with } \alpha_0 = 0.$$

In both spline and kernel regressions, we seek coefficients that when multiplied by basis functions allow us to potentially reconstitute the original function $f(x)$. In the case of spline regression, the basis functions are $\max(0, x - k)$ whereas in kernel regression we can have a variety of kernels.
Figure 5 depicts how λ controls the amount of regularization on α.

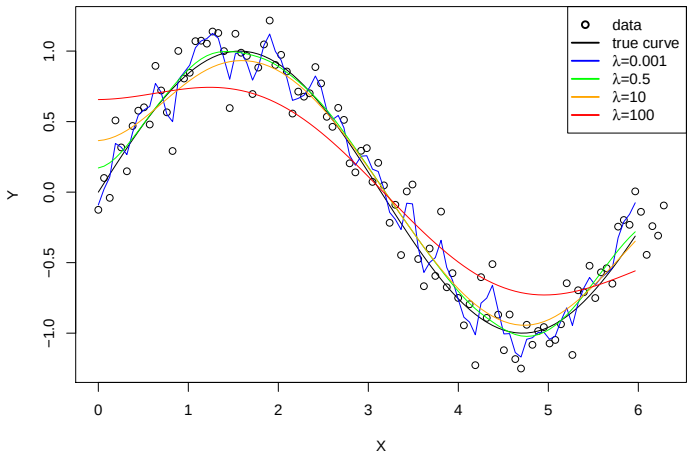

Figure 5: Spline regression with varying λ where $y_i = \sin x + \epsilon_i$ and $\epsilon_i \sim N(0, \sigma^2)$.

In comparison to Figure 3, Spline regression is prune to over-fitting with a large number of knots k_j and

low λ.

1.5 Lasso regression

The Lasso regression estimates β by

$$\hat{\beta}_\lambda = \arg\min_\beta \left[\frac{1}{2} \|\mathbf{Y} - \mathbf{X}\beta\|_{\ell_2}^2 + \lambda \|\beta\|_{\ell_1} \right],$$

where $\|\beta\|_{\ell_1} = \sum_{j=1}^p |\beta_j|$. Lasso stands for "least absolute shrinkage and selection operator." There is no closed form solution for general p.

When $p = 1$, *i.e.* $\mathbf{X}$ is an $n \times 1$ vector for n samples, and each sample X_i is a scalar, we do have closed form solution.

$$\hat{\beta}_\lambda = \begin{cases} (\langle \mathbf{Y}, \mathbf{X} \rangle - \lambda)/\|\mathbf{X}\|_{\ell_2}^2, & \text{if } \langle \mathbf{Y}, \mathbf{X} \rangle > \lambda; \\ (\langle \mathbf{Y}, \mathbf{X} \rangle + \lambda)/\|\mathbf{X}\|_{\ell_2}^2, & \text{if } \langle \mathbf{Y}, \mathbf{X} \rangle < -\lambda; \\ 0 & \text{if } |\langle \mathbf{Y}, \mathbf{X} \rangle| \le \lambda. \end{cases}$$

We can write it as

$$\hat{\beta}_\lambda = \text{sign}(\hat{\gamma}) \max(0, |\hat{\gamma}| - \lambda/\|\mathbf{X}\|_{\ell_2}^2),$$

where $\hat{\gamma} = \langle \mathbf{Y}, \mathbf{X} \rangle / \|\mathbf{X}\|_{\ell_2}^2$ is the least squares estimator. The above transformation from $\hat{\gamma}$ to $\hat{\beta}_\lambda$ is called soft thresholding.

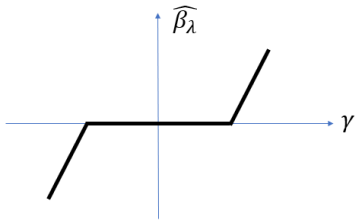

Figure 6: Soft thresholding.

Compare Lasso with ridge regression in one-dimensional situation, the latter being

$$\text{Ridge regression} \quad \longleftarrow \quad \hat{\beta}_\lambda = \langle \mathbf{Y}, \mathbf{X} \rangle / (\|\mathbf{X}\|_{\ell_2}^2 + \lambda),$$

$$\text{Lasso regression} \quad \longleftarrow \quad \hat{\beta}_\lambda = \text{sign}(\hat{\gamma}) \max(0, |\hat{\gamma}| - \lambda/\|\mathbf{X}\|_{\ell_2}^2),$$

the behavior of Lasso is richer, including

- shrinkage (by subtracting λ)
- selection (via thresholding at λ)

Comparisons between the ridge regression and the Lasso regression.

- **Ridge regression $\longrightarrow$ no dominating features:** The ridge regression penalizes elements with large absolute values in β, i.e., avoiding using very few feature dimensions for regression. The model uses a large number of feature dimensions for regression, and each dimension contributes a little to the regression result.

- **Lasso regression $\longrightarrow$ sparse features:** The Lasso regression prefers sparse β, i.e., only a small number of components of β are non-zero.

1.6 Primal form of Lasso

- **Primal form of Lasso:** $\min \|\mathbf{Y} - \mathbf{X}\beta\|_{\ell_2}^2/2$ subject to $\|\beta\|_{\ell_1} \leq t$
- **Dual form of Lasso:** $\min \|\mathbf{Y} - \mathbf{X}\beta\|_{\ell_2}^2/2 + \lambda\|\beta\|_{\ell_1}$

where $\|\beta\|_{\ell_1} = \sum_{j=1}^p |\beta_j|$, $\|\beta\|_{\ell_2} = \sqrt{\sum_{j=1}^p |\beta_j|^2}$.

The two forms are equivalent with a one-to-one correspondence between t and λ. If $\hat{\beta}_\lambda$ is the solution to the dual form, then it must be the solution to the primal form with $t = \|\hat{\beta}_\lambda\|_{\ell_1}$. **Otherwise**, let us assume there exist a different the solution $\hat{\beta}$ to the primal form, which is different from the solution $\hat{\beta}_\lambda$ to the dual form. Then,

$$
\begin{aligned}
&\text{Let} \quad \hat{\beta}_\lambda = \text{argmin} \|\mathbf{Y} - \mathbf{X}\beta\|_{\ell_2}^2/2 + \lambda\|\beta\|_{\ell_1} \\
&\text{Let} \quad t = \|\hat{\beta}_\lambda\|_{\ell_1} \\
&\text{Let} \quad \hat{\beta} = \underset{\beta}{\text{argmin}} \|\mathbf{Y} - \mathbf{X}\beta\|_{\ell_2}^2/2 \qquad \text{s.t.} \qquad \|\beta\|_{\ell_1} \leq t \\
\Rightarrow &\|\mathbf{Y} - \mathbf{X}\hat{\beta}\|_{\ell_2}^2/2 < \|\mathbf{Y} - \mathbf{X}\hat{\beta}_\lambda\|_{\ell_2}^2/2, \qquad \|\hat{\beta}\|_{\ell_1} \leq \|\hat{\beta}_\lambda\|_{\ell_1} \\
\Rightarrow &\|\mathbf{Y} - \mathbf{X}\hat{\beta}\|_{\ell_2}^2/2 + \lambda\|\hat{\beta}\|_{\ell_1} < \|\mathbf{Y} - \mathbf{X}\hat{\beta}_\lambda\|_{\ell_2}^2/2 + \lambda\|\hat{\beta}_\lambda\|_{\ell_1}
\end{aligned}
$$

which conflicts with the assumption that $\hat{\beta}_\lambda = \text{argmin}_\beta \|\mathbf{Y} - \mathbf{X}\beta\|_{\ell_2}^2/2 + \lambda\|\beta\|_{\ell_1}$ is the solution to the dual form.

The primal form also reveals the sparsity inducing property of ℓ_1 regularization in that the ℓ_1 ball has low-dimensional corners, edges, and faces, but is still barely convex.

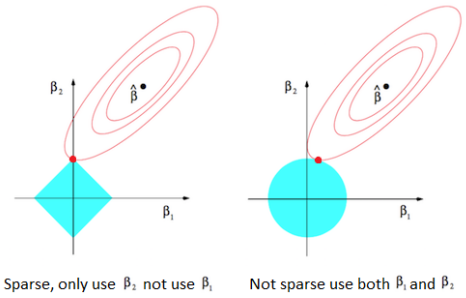

Figure 7: Lasso. Source: web.

The above is the well known figure of Lasso taken from the web. Take the left plot for example. The blue region is $\|\beta\|_{\ell_1} \leq t$. The red curves is the contour plot, where each red elliptical circle consists of those β that have the same value of $\|\mathbf{Y} - \mathbf{X}\beta\|_{\ell_2}^2$. The circle on the outside has bigger$\|\mathbf{Y} - \mathbf{X}\beta\|_{\ell_2}^2$ than the circle inside. The solution to the problem of $\min \|\mathbf{Y} - \mathbf{X}\beta\|_{\ell_2}^2$ subject to $\|\beta\|_{\ell_1} \leq t$ is where the red circle touches the blue region. Any other points in the blue region will be outside the outer red circle and thus have bigger values of $\|\mathbf{Y} - \mathbf{X}\beta\|_{\ell_2}^2$. The reason that the ℓ_1 regularization induces sparsity is that it is likely for the red circle to touch the blue region at a corner, which is a sparse solution. If we use ℓ_2 regularization, as is the case with the plot on the right, then the solution is not sparse in general.

1.7 Coordinate descent for Lasso solution path

for $\lambda = 10^a, 10^{a-\Delta}, 10^{a-2\Delta}, 10^{a-3\Delta}, \ldots, 10^b$ **do**
  for *Feature dimension* $j = 1, 2, \ldots, p$ **do**
    Compute the residual, $\mathbf{R}_j = \mathbf{Y} - \sum_{k \neq j} \mathbf{X}_k \beta_k$;
    Update the parameter of the j-th dimension, $\beta_j = \text{sign}(\hat{\gamma}_j) \max(0, |\hat{\gamma}_j| - \lambda / \|\mathbf{X}\|_{\ell_2}^2)$, where $\hat{\gamma}_j = \langle \mathbf{R}_j, \mathbf{X}_j \rangle / \|\mathbf{X}_j\|_{\ell_2}^2$
  end
end

For multi-dimensional $\mathbf{X} = (\mathbf{X}_j, j = 1, ..., p)$, we can use the coordinate descent algorithm to compute $\hat{\beta}_\lambda$. The algorithm updates one component at a time, i.e., given the current values of $\beta = (\beta_j, j = 1, ..., p)$, let $\mathbf{R}_j = \mathbf{Y} - \sum_{k \neq j} \mathbf{X}_k \beta_k$, we can update $\beta_j = \text{sign}(\hat{\gamma}_j) \max(0, |\hat{\gamma}_j| - \lambda / \|\mathbf{X}\|_{\ell_2}^2)$, where $\hat{\gamma}_j = \langle \mathbf{R}_j, \mathbf{X}_j \rangle / \|\mathbf{X}_j\|_{\ell_2}^2$.

We can find the solution path of Lasso by starting from a big λ so that all of the estimated β_j are zeros. Then we gradually reduce λ. For each λ, we cycle through $j = 1, ..., p$ for coordinate descent until convergence, and then we lower λ. This gives us $\hat{\beta}(\lambda)$ for the whole range of λ. The whole process is a forward selection process, which sequentially selects new variables and occasionally removes selected variables.

1.8 Bayesian regression

There is also a Bayesian interpretation of regularization.

For example, the ridge regression has a Bayesian interpretation. Let $\beta \sim \text{N}(0, \tau^2 \mathbf{I}_p)$ be the prior distribution of β. The log probability density of β and $\mathbf{Y}$ is

$$\log p(\beta | X, Y) = -\frac{1}{2\sigma^2} \|\mathbf{Y} - \mathbf{X}\beta\|_{\ell_2}^2 - \frac{1}{2\tau^2} \|\beta\|_{\ell_2}^2 + C,$$

up to an additive constant. The above function is quadratic in β. By setting the first derivative to 0, we get the mode of β,

$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X} + \frac{\sigma^2}{\tau^2} \mathbf{I}_p)^{-1} \mathbf{X}^\top \mathbf{Y}.$$

which corresponds to the ridge regression with $\lambda = \sigma^2 / \tau^2$.

Homework: Please write the process to derive $\log p(\beta | X, Y) = -\frac{1}{2\sigma^2} \|\mathbf{Y} - \mathbf{X}\beta\|_{\ell_2}^2 - \frac{1}{2\tau^2} \|\beta\|_{\ell_2}^2 + C$. You may learn it from the video.

1.9 SVM and ridge logistic regression

Given the regularized loss function

$$loss(\beta) = \sum_{i=1}^{n} \max(0, 1 - y_i X_i^\top \beta) + \frac{\lambda}{2} \|\beta\|^2,$$

we can solve β by gradient descent. The gradient is

$$loss'(\beta) = -\sum_{i=1}^{n} 1(y_i X_i^\top \beta < 1) y_i X_i + \lambda\beta,$$

where $1(\cdot)$ is the indicator function.

This is similar to the ridge logistic regression

$$loss(\beta) = \sum_{i=1}^{n} \underbrace{\log[1 + \exp(-y_i X_i^\top \beta)]}_{\text{logistic loss}} + \frac{\lambda}{2}\|\beta\|^2,$$

whose gradient is

$$loss'(\beta) = -\sum_{i=1}^{n} sigmoid(-y_i X_i^\top \beta) y_i X_i + \lambda\beta.$$

where $sigmoid(a) = \frac{1}{1+e^{-a}}$.

Homework: Please draw the graph, which summarizes relationships between the ridge regression, the kernel regression, the spline regression, the LASSO regression, the Gaussian prior, the SVM, and the generative model. This graph is shown in the video.

1.10 Linear Version

Consider the loss function

$$\sum_{i=1}^{n} L(y_i; x_i^\top \beta) + \lambda\|\beta\|^2.$$

We will show that the minimizer can be written in the form of $\hat{\beta} = \sum_{i=1}^{n} \alpha_i x_i$.

Proof:

Suppose that we need additional vectors, orthogonal to the x_i's ($i \in \{1...n\}$), to generate the solution to the problem. Call that alternative solution $\tilde{\beta}$. Without loss of generality, assume we need K additional perpendicular vectors to generate $\tilde{\beta}$, where K can be any positive integer different from 0. Then, we can write $\tilde{\beta} = \sum_{i=1}^{n} \alpha_i x_i + \sum_{k=1}^{K} \kappa_k x_k$, with $x_k \perp x_i$, for all i and k.

(i) Notice then that

$$\begin{aligned}
x_i^T \tilde{\beta} &= x_i^T (\sum_{i=1}^{n} \alpha_i x_i + \sum_{k=1}^{K} \kappa_k x_k) \\
&= \sum_{i=1}^{n} \alpha_i x_i^T x_i + \sum_{k=1}^{K} \kappa_k x_i^T x_k \\
&= \sum_{i=1}^{n} \alpha_i x_i^T x_i + \sum_{k=1}^{K} \kappa_k 0 \\
&= \sum_{i=1}^{n} \alpha_i x_i^T x_i \\
&= x_i^T \hat{\beta}
\end{aligned}$$

, given the x_k's are perpendicular to the x_i's. So the individual loss functions $L(y_i; x_i^T\beta)$ under $\tilde{\beta}$ take on the same values than under $\hat{\beta}$.

(ii) Let $\gamma \stackrel{\text{def}}{=} \sum_{k=1}^{K} \kappa_k x_k$. We have $\|\tilde{\beta}\|^2 = \|\hat{\beta}\|^2 + \|\gamma\|^2 \geq \|\hat{\beta}\|^2$, where this follows again from the fact that the $x_k's$ are perpendicular to the $x_i's$. So the penalty term $(\lambda\|\beta\|^2)$ under $\tilde{\beta}$ is at greater than or equal to the penalty term under $\hat{\beta}$.

$$\sum_{i=1}^{n} L(y_i; x_i^\top \hat{\beta}) + \lambda\|\hat{\beta}\|^2 \leq \sum_{i=1}^{n} L(y_i; x_i^\top \tilde{\beta}) + \lambda\|\tilde{\beta}\|^2.$$

Therefore, $\tilde{\beta}$ is not the minimizer.

These two facts directly imply that the solution must take on the form $\hat{\beta} = \sum_{i=1}^{n} \alpha_i x_i$ (there is an implicit assumption here that the solution is unique, i.e., that the loss function is convex, so that if we could write a solution in the form of $\tilde{\beta}$ - i.e., if we could find a $\tilde{\beta}$ that gave us the same penalty value and loss as $\hat{\beta}$ - then we could also write that solution in the form of $\hat{\beta} = \sum_{i=1}^{n} \alpha_i x_i$.

This implies that vectors perpendicular to our observations are not helpful in deriving the solution. Note the interesting case where X is $p \times n$, with $p > n$. One could conjecure that n observations in this case are not enough to derive the solution, and that one would need p observations instead. We can look at this conjecture by letting K be $p - n$ in the proof above, to see that the n observations are enough, so that the solution takes the form $\hat{\beta} = \sum_{i=1}^{n} \alpha_i x_i$.

1.11 Feature version

Consider the loss function

$$\sum_{i=1}^{n} L(y_i; \phi(x_i)^\top \beta) + \lambda\|\beta\|^2.$$

The minimizer can be written as $\hat{\beta} = \sum_{i=1}^{n} \alpha_i \phi(x_i)$.

Proof:

The proof is analogous to the one above, except that here our features are not x_i, but the vectors $\phi(x_i)$. As above, suppose that we need additional vectors, perpendicular to the $\phi(x_i)'s$ $(i \in \{1...n\})$, to generate the solution to the problem. Denote that alternative solution $\tilde{\beta}$. Again, assume we need K additional perpendicular vectors to generate $\tilde{\beta}$, where K can be any positive integer different from 0. Then, here we can write $\tilde{\beta} = \sum_{i=1}^{n} \alpha_i \phi(x_i) + \sum_{k=1}^{K} \kappa_k \phi(x_k)$, with $\phi(x_k) \perp \phi(x_i)$, for all i and k.

(i) Notice that $\phi(x_i)^T\tilde{\beta} = \phi(x_i)^T\hat{\beta}$, given the $\phi(x_k)'s$ are perpendicular to the $\phi(x_i)'s$. So the individual loss functions $L(y_i; \phi(x_i)^T\beta)$ under $\tilde{\beta}$ take on the same values than under $\hat{\beta}$.

(ii) Let $\gamma \stackrel{\text{def}}{=} \sum_{k=1}^{K} \kappa_k \phi(x_k)$. We have $\|\tilde{\beta}\|^2 = \|\hat{\beta}\|^2 + \|\gamma\|^2 \geq \|\hat{\beta}\|^2$, where this follows from the fact that the $\phi(x_k)'s$ are perpendicular to the $\phi(x_i)'s$. So the penalty term under $\tilde{\beta}$ is at greater than or equal to the penalty term under $\hat{\beta}$.

$$\sum_{i=1}^{n} L(y_i; \phi(x_i)^\top \hat{\beta}) + \lambda\|\hat{\beta}\|^2 \leq \sum_{i=1}^{n} L(y_i; \phi(x_i)^\top \tilde{\beta}) + \lambda\|\tilde{\beta}\|^2.$$

Therefore, $\tilde{\beta}$ is not the minimizer.

These two facts directly imply that the solution must take on the form $\hat{\beta} = \sum_{i=1}^{n} \alpha_i \phi(x_i)$.

1.12 Gaussian Process and Bayesian Estimation

1.12.1 Linear version

Suppose $Y = X\beta + \epsilon$, where $\beta \sim \mathrm{N}(0, \tau^2 I_p)$, $\epsilon \sim \mathrm{N}(0, \sigma^2 I_n)$, and ϵ is independent of β. We want to find the posterior distribution $\Pr[\beta|Y, X]$.

Notice that if some arbitrary random vectors X_1 and X_2 are multivariate normal with joint-distributions given by $\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right)$, where

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

Then it follows that:

$$\Pr[X_2|X_1] \sim N(\mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(X_1 - \mu_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}]$$

We will use this property further below. Let's compute the distribution of $\Pr[Y|X]$, using our knowledge about β. I will omit the conditioning on X here, to facilitate notation. Then, we have:

$$E[Y] = XE[\beta] + E[\epsilon] = 0$$

Also:

$$\begin{aligned} Var[Y] &= Var[X\beta] + Var[\epsilon] \\ &= XVar[\beta]X^T + \sigma^2 I_n \\ &= \tau^2 XX^T + \sigma^2 I_n \end{aligned}$$

When A is independent on B,

$$\begin{aligned} Var[A + B] &= E_i[(a_i + b_i - E_j[a_j] - E_j[b_j])^2] \\ &= E_i[(a_i - E_j[a_j])^2] + E_i[(b_i - E_j[b_j])^2] + 2E_i[(a_i - E_j[a_j])(b_i - E_j[b_j])] \\ &= E_i[(a_i - E_j[a_j])^2] + E_i[(b_i - E_j[b_j])^2] + 2E_i[a_i - E_j[a_j]]E_i[b_i - E_j[b_j]] \\ &\quad \text{because } A \text{ is independent on } B. \\ &= Var[A] + Var[B] + 0 \times 0 \\ &= Var[A] + Var[B] \end{aligned}$$

$\Pr[Y|X]$ is just a sum of Gaussian distributions, so $\Pr[Y|X]$ also follows a Gaussian distribution. This, the marginal we are looking for is:

$$Y \sim N(0, \tau^2 X X^T + \sigma^2 I_n)$$

$$\begin{aligned}
Cov(Y, \beta) =& E_i[(Y_i - E_j[Y_j])(\beta_i - E_j[\beta_j])^\top] \\
=& E_i[(X\beta_i + \epsilon_i - E_j[X\beta_j + \epsilon_j])(\beta_i - E_j[\beta_j])^\top] \\
=& E_i[(X\beta_i + \epsilon_i - E_j[X\beta_j] - E_j[\epsilon_j])(\beta_i)^\top] \qquad \text{because } E_j[\beta_j] = 0 \\
=& E_i[(X\beta_i + \epsilon_i - E_j[X\beta_j])\beta_i^\top] \qquad \text{because } E_j[\epsilon_j] = 0 \\
=& E_i[X\beta_i\beta_i^\top] + E_i[\epsilon_i\beta_i^\top] - E_i[E_j[X\beta_j]\beta_i^\top] \\
=& E_i[X\beta_i\beta_i^\top] + 0 - E_i[0\beta_i^\top] \\
=& X E_i[\beta_i\beta_i^\top] \\
=& X(\tau^2 I_p) \\
=& \tau^2 X
\end{aligned}$$

Note that β is independent of X. We then have the joint distribution:

$$\begin{bmatrix} Y \\ \beta \end{bmatrix} \sim N(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 X X^T + \sigma^2 I_n & \tau^2 X \\ \tau^2 X^T & \tau^2 I_p \end{bmatrix})$$

Using the property of multivariate normals discussed above, we have that the posterior for β is:

$$\Pr[\beta|Y, X] = N(\tau^2 X^T (\tau^2 X X^T + \sigma^2 I_n)^{-1} Y, \tau^2 I_p - \tau^2 X^T (\tau^2 X X^T + \sigma^2 I_n)^{-1} \tau^2 X)$$

This distribution is closely related to ridge regression.
The relationship lies in that $\hat{\beta}_{ridge}$ is the mode of the posterior $\Pr[\beta|Y,X]$, which corresponds to the mean of $\Pr[\beta|Y,X]$ given this a multivariate Gaussian distribution. So doing ridge regression is the same as maximizing the posterior probability of observing β, given your data Y and X (MAP). It is easier to see this if we look at the posterior in a different way.

$$\beta \sim N(0, \tau^2 I_p)$$
$$Y - X\beta = \epsilon \sim N(0, \sigma^2 I_n)$$

We have:

$$\begin{aligned} p(\beta|Y,X) &\propto p(\beta)p(Y|X,\beta) \\ &\propto \exp(-\frac{1}{2\tau^2}|\beta|^2)\exp(-\frac{1}{2\sigma^2}|Y-X\beta|^2) \\ &= \exp(-\frac{1}{2}\left[\frac{1}{\sigma^2}|Y-X\beta|^2 + \frac{1}{\tau^2}|\beta|^2\right]) \end{aligned}$$

If we take the derivative of $\log(p(\beta|Y,X))$ and set it to 0, we will get the mode of $\Pr[\beta|Y,X]$, which here is the same as the mean of $\Pr[\beta|Y,X]$ (Except that it will be written here in a different format). Note that this is the same as maximizing the posterior probability of beta, given Y and X (MAP). This gives us:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmax}} \log(p(\beta|Y,X)) = (X^T X + \lambda I_p)^{-1}(X^T Y)$$

With $\lambda = \sigma^2/\tau^2$. Which is the same as $\hat{\beta}_{ridge}$.

1.12.2 Feature version

Suppose $y_i = \phi(x_i)^\top \beta + \epsilon_i$, with the same prior as above. Let's look for the distribution of $\Pr[\beta|Y,X]$. Let $f(x) = \phi(x)^\top \beta$.

Firstly, define $\phi(X) = \begin{bmatrix} \phi(x_1)^\top \\ \phi(x_2)^\top \\ \dots \\ \phi(x_n)^\top \end{bmatrix}_{n \times d}$, assuming without loss of generality that ϕ is d-dimensional. It should be clear here that one can simply replace X with $\phi(X)$ in the derivations above, and all steps would follow in an identical fashion. In other words, we can simply let $\phi(X)$ represent our new set of features, i.e., our new X matrix. Then, following the same steps as above, we get:

$$\Pr[\beta|Y,X] = N(\tau^2\phi(X)^T(\tau^2\phi(X)\phi(X)^T + \sigma^2 I_n)^{-1}Y, V)$$

Where V here is $V = \tau^2 I_p - \tau^2\phi(X)^T(\tau^2\phi(X)\phi(X)^T + \sigma^2 I_n)^{-1}\tau^2\phi(X)$.
And note there are other ways to express this distribution.

Also, notice that we can write Y as:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ ... \\ y_n \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ ... \\ f(x_n) \end{bmatrix} + \epsilon$$

We have $Cov(f(x), f(x')) = Cov(\phi(x)^T\beta, \phi(x')^T\beta) = \tau^2\phi(x)^T\phi(x')$. Denote $\tau^2\phi(x)^T\phi(x')$ as $K(x, x')$ (with the additional weight τ^2). Then, it follows that the marginal distribution for Y is:

$$\boldsymbol{Y} \sim N(\boldsymbol{0}, \boldsymbol{K} + \sigma^2\boldsymbol{I_n})$$

Or:

$$Y \sim N(\begin{bmatrix} 0 \\ ... \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2\phi(x_1)^T\phi(x_1) + \sigma^2 & ... & \tau^2\phi(x_1)^T\phi(x_n) \\ ... & ... & ..... \\ \tau^2\phi(x_n)^T\phi(x_1) & ... & \tau^2\phi(x_n)^T\phi(x_n) + \sigma^2 \end{bmatrix}$$

Let's again use $K(x, x')$ to denote $\tau^2\phi(x)^T\phi(x') = Cov(f(x), f(x'))$. Then, we can write:

$$\begin{bmatrix} \boldsymbol{Y} \\ f(x_0) \end{bmatrix} = N\left(\begin{bmatrix} \boldsymbol{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \boldsymbol{K} + \sigma^2\boldsymbol{I_n} & \boldsymbol{K}(\boldsymbol{x}, x_0) \\ \boldsymbol{K}(x_0, \boldsymbol{x}) & K(x_0, x_0) \end{bmatrix}_{(n+1)\times(n+1)}\right)$$

To get the conditional distribution $\Pr[f(x_0)|Y, X]$, we can just apply our knowledge of multivariate Gaussian distributions. This gives us:

$$\Pr[f(x_0)|Y, X] \sim N(K(x_0, x)(K + \sigma^2 I_n)^{-1}Y, K(x_0, x_0) - K(x_0, x)(K + \sigma^2 I_n)^{-1}K(x_0, x)^T)$$

Where we abuse notation in that $K(x_0, x)$ here is a $1 \times n$ vector, and $K(x, x_0)$ is a $n \times 1$ vector, and each entry i, j of K, $K(x_0, x)$, and $K(x, x_0)$ is given by $\tau^2\phi(x_i)^T\phi(x_j)$.

1.12.3 Kernel version

Let $f(x) = \phi(x)^\top\beta$. We then think of f as a random process, because of the randomness of β, and we write $f \sim GP(0, K)$. Suppose we observe $y_i = f(x_i) + \epsilon_i$. For any fixed point x_0, let's find the joint distribution of $(y_1, ..., y_n, f(x_0))$, and the conditional distribution of $\Pr[f(x_0)|Y, X]$.

Firstly, we want $K(x, x') = Cov(f(x), f(x'))$. We have:

$$\begin{aligned} Cov(f(x), f(x')) &= Cov(\phi(x)^T\beta, \phi(x')^T\beta) \\ &= E[\phi(x)^T\beta\beta^T\phi(x')] \\ &= \tau^2\phi(x)^T\phi(x'), \end{aligned}$$

The second line follows from the fact that $E[\phi(x)^T\beta] = E[\phi(x')^T\beta] = 0$. In other words, it follows from the fact that $f \sim GP(0, K)$, so the mean of f is 0 (this in turn follows from the fact that we generally assume here $\beta \sim N(0, \tau^2 I_d)$, taking $\phi(x)$ to be a $d - dimensional$ vector). The third line follows from recognizing that $E[\beta\beta^T]$ is simply the variance of β.

Next, notice that we can write:

$$\begin{bmatrix} y_1 \\ y_2 \\ ... \\ y_n \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ ... \\ f(x_n) \end{bmatrix} + \epsilon$$

Let $Var(\epsilon)$ now be $\sigma^2 I_n$. Then, the equality above implies $Y \sim N(0, K + \sigma^2 I_n)$, K here being an $n \times n$ matrix. We also have $E[f(x_0)] = 0$, $Var[f(x_0)] = K(x_0, x_0)$ - the function K was derived above. Lastly, $Cov(Y, f(x_0)) = K(x, x_0)$, where we abuse notation here a bit given K here is an $n \times 1$ vector. Similarly, $Cov(f(x_0), Y) = K(x_0, Y)$, K here being a $1 \times n$ vector. So:

$$\begin{bmatrix} \boldsymbol{Y} \\ f(x_0) \end{bmatrix} = N\left(\begin{bmatrix} \boldsymbol{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \boldsymbol{K} + \sigma^2 \boldsymbol{I_n} & \boldsymbol{K}(\boldsymbol{x}, x_0) \\ \boldsymbol{K}(x_0, \boldsymbol{x}) & K(x_0, x_0) \end{bmatrix}_{(n+1)\times(n+1)} \right)$$

Above, we used bold-faced letters to highlight matrices and vectors. To get the conditional distribution $\Pr[f(x_0)|Y, X]$, we can just apply our knowledge of multivariate normals. This gives us:

$$\Pr[f(x_0)|Y, X] \sim N(K(x_0, x)(K + \sigma^2 I_n)^{-1} Y, K(x_0, x_0) - K(x_0, x)(K + \sigma^2 I_n)^{-1} K(x_0, x)^T)$$

There is a connection with kernel regression. We know $\hat{c} = (K + \sigma^2 I_n)^{-1} Y$, so the estimated coefficients in the kernel regression show up in the mean of $\Pr[f(x_0)|Y, X]$. This mean, $K(x_0, x)\hat{c}$, is actually simply the prediction we would get if we did a kernel regression. In other words, $E[f(x_0)|X, Y] = \sum_{i=1}^{n} \hat{c}_i K(x_0, x_i)$. The variance of the posterior shown above actually allows us to deal with uncertainty, creating a posterior interval for our kernel regression estimate. We can use the idea that the function f is random here, to deal with uncertainty in our kernel regression estimates, writing, for example, that our estimate would be $\sum_{i=1}^{n} \hat{c}_i K(x_0, x_i) \pm 2V$, where V is the variance of the posterior of f, above. This means we can conduct kernel regressions and create posterior intervals for estimates by adding, say, $\pm 2V$ to them.

1.12.4 Marginal likelihood

The marginal distribution of Y is $\mathrm{N}(0, K_\gamma + \sigma^2 I_n)$, where $K = (K_{ij} = K(x_i, x_j))$, and γ is the parameter of the Gaussian kernel. Let's write down the log marginal likelihood for determining γ.

We can write the likelihood as:

$$\frac{1}{(2\pi)^{n/2} |\Sigma_\gamma|^{1/2}} exp(-\frac{1}{2} Y^T \Sigma_\gamma^{-1} Y),$$

Where $\Sigma_\gamma = K_\gamma + \sigma^2 I_n$.

The log-marginal-likelihood for determining γ is then:

$$l = -\frac{1}{2} Y^T \Sigma_\gamma^{-1} Y - \frac{1}{2} log(|\Sigma_\gamma|) - \frac{n}{2} log(2\pi)$$

2 Appendix

2.1 For Bayesian regression

$$P(\beta|X,Y) = \frac{P(\beta)P(Y|X,\beta)}{P(Y)}$$
$$\Longrightarrow \quad \log P(\beta|X,Y) = \log P(\beta) + \log P(Y|X,\beta) - \log P(Y)$$

2.2 The overview

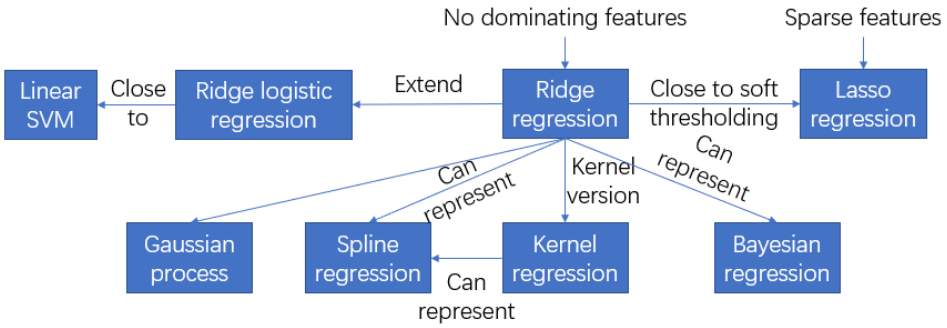

Figure 8: Overview.