REGULARIZATION

Nonlinear Feature Maps

One way to create nonlinear estimators or classifiers is to first transform the data via a nonlinear feature map

$$\Phi : \mathbb{R}^d \longrightarrow \mathcal{H}$$

and apply a linear method to the transformed data $\Phi(x_1), \ldots, \Phi(x_n)$

Example 1 $\quad y_i = f(x_i) + \varepsilon_i \,, \quad i = 1, \ldots, n$

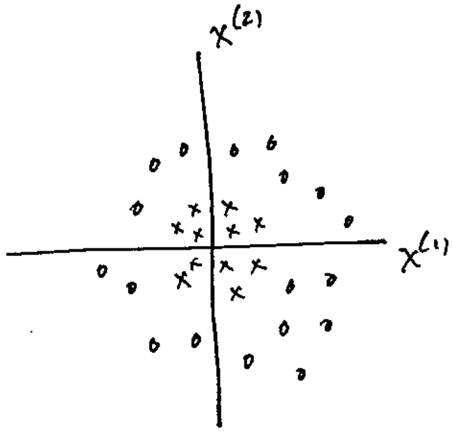
$$f(x) = \sum_{j=0}^{p} \beta_j x^j \qquad \text{(degree } p \text{ polynomial)}$$

To estimate f, we may apply least-squares regression to the transformed data $\Phi(x_i)$, where $\Phi(x) = [1 \ x \ x^2 \ \cdots \ x^p]^T$

$$\Rightarrow \hat{\beta} = (A^T A)^{-1} A^T \underline{y} \,, \quad A =$$

(A)

Example 2



$$x \mapsto \Phi(x) = \begin{bmatrix} 1 \\ x^{(1)} \\ x^{(2)} \\ x^{(1)} \cdot x^{(2)} \\ (x^{(1)})^2 \\ (x^{(2)})^2 \end{bmatrix}$$

Then the data are linearly separable in the new feature space. They are correctly classified by

$$\text{sign}\{ w^T \Phi(x) \}$$

where

Ⓑ $\qquad w =$

In many applications, we don't know exactly how to design $\Phi(x)$, we just know that some nonlinear features are probably important.

In such situations, it is common to include a large number of nonlinear features, in hopes that some of them are relevant.

Unfortunately, this practice can lead to ______

© _________ problems.

Example 1, revisited | In least squares polynomial regression, we have

$$\hat{\beta} = (A^T A)^{-1} A^T \underline{y}$$

where

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^p \\ 1 & x_2 & x_2^2 & \cdots & x_2^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^p \end{bmatrix}$$

As p increases, the smallest eigenvalue of A^TA gets very, very small, so that matrix

Ⓓ inversion is numerically ________.

Alternatively, the least-squares criterion becomes very ______ near the minimizer.

Essentially, when there are many features, it is possible that a huge coefficient on one feature could be cancelled out by other features.

To remedy the situation, we can incorporate a regularization term into design criteria that will keep coefficients small.

Owing to computational convenience, the most common kind of regularization is ________. We'll examine two cases in detail.

Ridge Regression

Given $y_i = f(x_i) + \varepsilon_i$, where $f(x_i) = \beta^T x_i + \beta_0$.

Instead of minimizing the sum of squared errors,

in ridge regression, we minimize

$$\sum_{i=1}^{n} (y_i - \beta^T x_i - \beta_0)^2 + \lambda \|\beta\|^2$$

where $\lambda > 0$ is a tuning parameter.

<u>Note</u>: β_0 is not penalized so that our solution is independent of where the origin

Let's derive the solution. First, let's eliminate β_0

$$\frac{\partial}{\partial \beta_0} = -2 \sum_{i=1}^{n} (y_i - \beta^T x_i - \beta_0) = 0$$

$$\Rightarrow \hat{\beta}_0 = \frac{1}{n} \sum_i y_i - \hat{\beta}^T x_i$$

E

$$=$$

Thus, we are left to minimize

$$\sum_{i=1}^{n} \left(y_i - \bar{y} - \beta^T (x_i - \bar{x}) \right)^2 + \lambda \beta^T \beta$$

wrt β. For convencience, assume $\bar{y} = 0$, $\bar{x} = 0$.

The criterion may be written

$$(\underline{y} - A\beta)^T (\underline{y} - A\beta) + \lambda \beta^T \beta, \quad A = \begin{bmatrix} x_1^{(1)} & \cdots & x_1^{(d)} \\ \vdots & & \\ & & \\ x_n^{(1)} & \cdots & x_n^{(d)} \end{bmatrix}$$

$$= \underline{y}^T \underline{y} + \beta^T A^T A \beta - 2\beta^T A^T \underline{y} + \lambda \beta^T \beta$$

$$= \beta^T \left[A^T A + \lambda I \right] \beta - 2\beta^T A^T \underline{y} + \underline{y}^T \underline{y}$$

$$\frac{\partial}{\partial \beta} = 0 \implies (A^T A + \lambda I)\beta = A^T \underline{y}$$

(F) $\implies$

<u>Observations</u> • $\lambda = 0$ recovers least-squares linear regr.

• λI increases the eigenvalues of $A^T A$ by λ, so that $A^T A + \lambda I$ is not ill-conditioned.

Soft Margin Hyperplane

The training error of a linear classifier may be bounded as

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{y_i(w^T x_i + b) < 0\}}$$

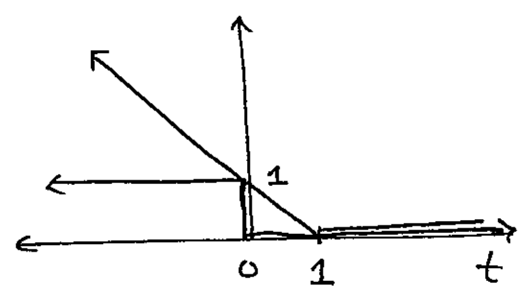
$$\leq \frac{1}{n}\sum \phi\left(y_i(w^T x_i + b)\right)$$

where $\phi(t)$ is any upper bound on $\mathbb{1}_{\{t<0\}}$.

Let's take

$$\phi(t) = \max\{0, 1-t\}$$

$$=: (1-t)_+$$



In addition, let's add a quadratic penalty to keep the coefficients small.

$$\Longrightarrow \min_{w,b} \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \left(1 - y_i(w^T x_i + b)\right)_+$$

Compare this to the quadratic program for the optimal soft-margin hyperplane:

$$\min_{w,b,\xi_i} \quad \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i$$

$$\text{s.t.} \quad y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i=1,\dots,n$$

$$\xi_i \geq 0, \quad i=1,\dots,n$$

<u>Claim</u> If $C = \frac{1}{\lambda}$, these two optimization problems are solved by the same w, b.

(G) Proof:

$\boxed{\text{Key}}$ A.

$$A = \begin{bmatrix} \Phi(x_1)^T \\ \Phi(x_2)^T \\ \vdots \\ \Phi(x_n)^T \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^p \\ 1 & x_2 & x_2^2 & \cdots & x_2^p \\ & & \vdots & & \\ 1 & x_n & x_n^2 & \cdots & x_n^p \end{bmatrix}$$

B.

$$w = \begin{bmatrix} -r^2 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$ (circle of radius r)

C. ill-conditioned

D. unstable, flat, quadratic

E. $\bar{y} - \hat{\beta}^T \bar{x}$ F. $\hat{\beta} = (A^T A + \lambda I)^{-1} A^T \underline{y}$

G. Since scaling an objective function by a positive constant does not change the solution, it suffices to show that

$$\text{QP1} \quad \min_{(w,b)} \ \frac{1}{2}\|w\|^2 + \frac{1}{n\lambda} \sum \left(1 - y_i(w^T x_i + b)\right)_+$$

$$\text{QP2} \quad \min_{(w,b,\xi)} \ \frac{1}{2}\|w\| + \frac{1}{n\lambda} \sum \xi_i$$

$$\text{s.t.} \quad y_i(w^T x_i + b) \geq 1 - \xi_i, \ \xi_i \geq 0$$

are solved by the same (w, b).

- Suppose (w^*, b^*) is an optimizer of QP1

<u>Claim</u>: (w^*, b^*, ξ^*) is an optimizer of QP2, where

$$\xi_i^* = \max(0, 1 - y_i (w^{*T} x_i + b^*))$$

Suppose not, and let (w, b, ξ) be an optimizer of QP2.

Since (w, b, ξ) is a global optimizer

- If $\xi_i > 0$, then $y_i (w^T x_i + b) = 1 - \xi_i$

  [otherwise, we could decrease the objective function without violating the constraints]

- If $\xi_i = 0$, then $y_i (w^T x_i + b) \geq 1$

Thus $\sum \xi_i = \sum (1 - y_i (w^T x_i + b))_+$ and so

$$\frac{1}{2}\|w\|^2 + \frac{1}{n\lambda} \sum (1 - y_i (w^T x_i + b))_+$$

$$= \frac{1}{2}\|w\|^2 + \frac{1}{n\lambda} \sum \xi_i$$

$$< \frac{1}{2}\|w^*\|^2 + \frac{1}{n\lambda} \sum \xi_i^*$$

$$= \frac{1}{2}\|w^*\| + \frac{1}{n\lambda} \sum (1 - y_i (w^{*T} x_i + b))_+$$

which contradicts optimality of (w^*, b^*) for QP1.

- Suppose (w^*, b^*, ξ^*) is an optimizer of QP2.

<u>Claim</u>: (w^*, b^*) is an optimizer of QP1.

The argument is similar and is left as an exercise.