Statistical Learning and Data Science

Xinwei Deng

xdeng@vt.edu

Department of Statistics

Classification Methods and Support Vector Machines

- Several Classification Methods
  - LDA and QDA
  - Naïve Bayes
- Support Vector Machines (SVM)

Assumption of Linear Discriminant Analysis

- Need to know $\delta_k(x) = \Pr(G = k|X = x), k = 1, \ldots, K$.
- $f_k(x)$ – the class-conditional density of X in class $G = k$, π_k – the prior probability of class k, i.e., $\Pr(G = k) = \pi_k$, thus $\sum_k^K \pi_k = 1$.
- According to Bayes theorem:

$$\Pr(G = k|X = x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^K f_l(x)\pi_l}.$$

- Estimate the density of X for each class, $f_k(x)$.

Linear Discriminant Analysis (LDA)

- Assume that in each class $G = k$, X follows multiple variate normal distribution.

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \exp(-\frac{1}{2}(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k))$$

- One more assumption for LDA: $\Sigma_k = \Sigma$ for all k, i.e., common covariance matrix.
- Comparing any two classes k and l, it is sufficient to look at their log-ratio

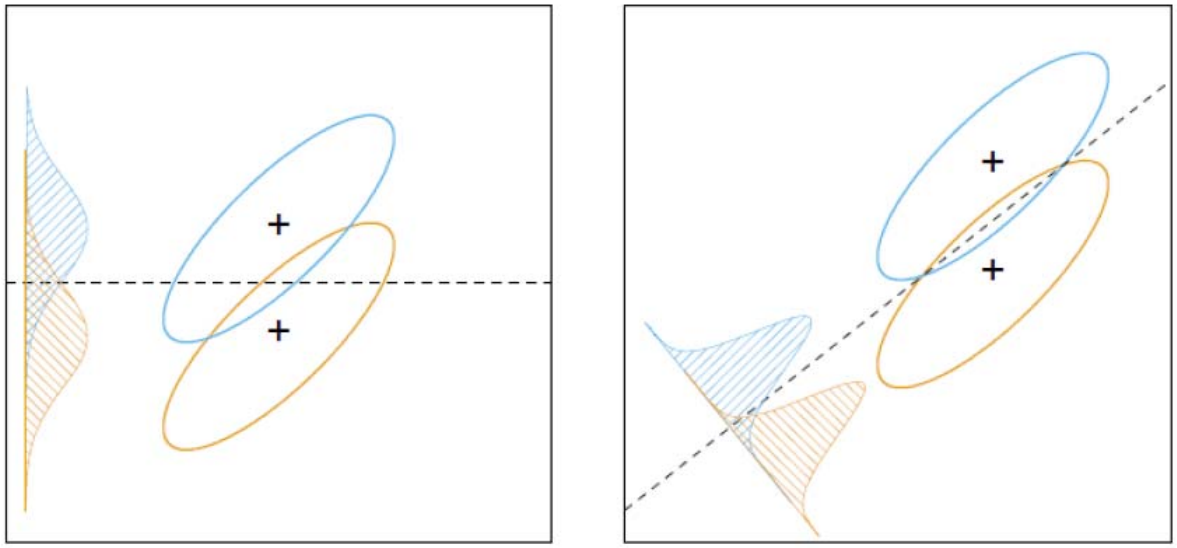
$$\begin{aligned}\log\frac{\Pr(G=k|X=x)}{\Pr(G=l|X=x)} &= \log\frac{f_k(x)}{f_l(x)} + \log\frac{\pi_k}{\pi_l} \\ &= \log\frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k+\mu_l)'\Sigma^{-1}(\mu_k-\mu_l) \\ &\quad + x'\Sigma^{-1}(\mu_k-\mu_l)\end{aligned}$$

Illustration for Two-Class LDA

- For two class, we classify to class 1 if

$$x'\hat{\Sigma}^{-1}(\hat{\mu}_1 - \hat{\mu}_2) > \frac{1}{2}(\hat{\mu}_1 + \hat{\mu}_2)'\hat{\Sigma}(\hat{\mu}_1 - \hat{\mu}_2) + \log\frac{N_1}{N_2},$$

where N_1, N_2 are numbers of observations in each class.



Linear Discriminant Analysis (Con't)

- Discriminant functions:

$$\delta_k(x) = x'\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k'\Sigma^{-1}\mu_k + \log \pi_k$$

- Estimate:
  - $\hat{\pi}_k = N_k/N$, where N_k is the number of observations of class-k. 1
  - $\hat{\mu}_k = \sum_{g_i=k} x_i/N_k$. p
  - $\hat{\Sigma} = \sum_{k=1}^{K} \sum_{g_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)'/(N-K)$.
- Number of parameters (apart from $\hat{\Sigma}$): only need the difference $\delta_k(x) - \delta_K(x)$,

$$(p+1) \times (K-1).$$

Quadratic Discriminant Analysis (QDA)

- If Σ_k are not assumed to be equal,

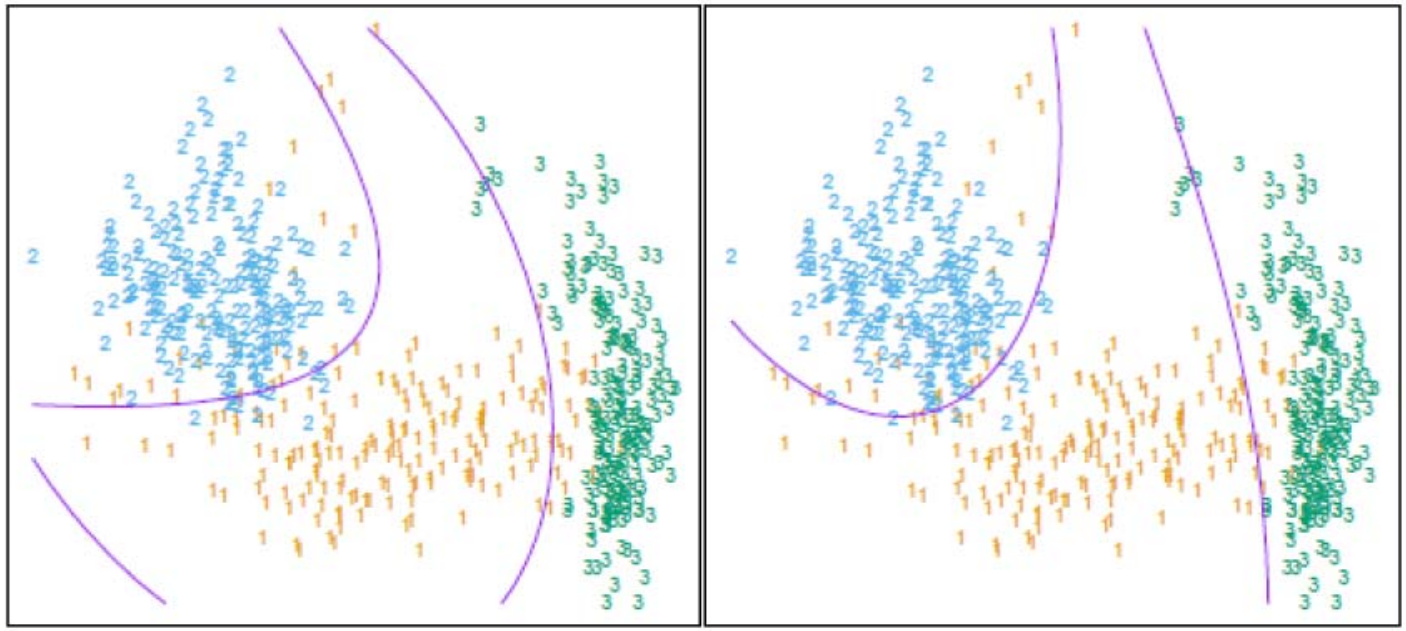
$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k) + \log\pi_k.$$

- Number of parameters: only need the difference $\delta_k(x) - \delta_K(x)$

$$\{1(\pi_k)+p(\mu_k)+\frac{p(p+1)}{2}(\Sigma)\}\times(K-1) = \{\frac{p(p+3)}{2}+1\}\times(K-1)$$

- Both LDA and QDA have good records of performances.

Illustration of QDA



Two methods for fitting quadratic boundaries. [Left] Quadratic decision boundaries, obtained using LDA in the five-dimensional "quadratic" space. [Right] Quadratic decision boundaries found by QDA. The differences are small, as is usually the case.

Regularized Discriminant Analysis (RDA)

- A compromise between LDA and QDA. The regularized covariance matrices have the form,

$$\hat{\Sigma}_k(\alpha) = \alpha\hat{\Sigma}_k + (1-\alpha)\hat{\Sigma}, \quad \alpha \in [0,1].$$

- Similar modification allow $\hat{\Sigma}$ itself to be shrunk toward the scalar covariance,

$$\hat{\Sigma}(\gamma) = \gamma\hat{\Sigma} + (1-\gamma)\hat{\sigma}^2\mathbf{I}, \quad \gamma \in [0,1].$$

- Together can lead to $\hat{\Sigma}(\alpha,\gamma)$.

Computations for LDA and QDA

- Simplify the computation by matrix decomposition.
- Let $\hat{\Sigma}_k = \mathbf{U}_k \mathbf{D}_k \mathbf{U}_k^T$, where $\mathbf{U}_k$ is the $p \times p$ orthogonal. $\mathbf{D}_k$ is a diagonal matrix of positive eigenvalues d_{lk}. Then

$$(\mathbf{x} - \hat{\boldsymbol{\mu}}_k)^T \hat{\Sigma}_k^{-1} (\mathbf{x} - \hat{\boldsymbol{\mu}}_k) = [\mathbf{U}_k^T (\mathbf{x} - \hat{\boldsymbol{\mu}}_k)]^T \mathbf{D}_k^{-1} [\mathbf{U}_k^T (\mathbf{x} - \hat{\boldsymbol{\mu}}_k)]$$

$$\log |\hat{\Sigma}_k| = \sum_l \log(d_{kl})$$

Fisher Discriminant Analysis

- Find the linear combination $\mathbf{z} = \mathbf{a}^T\mathbf{x}$ such that the between-class variance is maximized relative to the within-class variance, i.e.,

$$\max_{\mathbf{a}} \frac{\mathbf{a}'\mathbf{B}\mathbf{a}}{\mathbf{a}'\mathbf{W}\mathbf{a}}.$$

  - Between class variance: $\mathbf{B} = \sum_{k=1}^{K} N_k(\hat{\boldsymbol{\mu}}_k - \bar{\boldsymbol{\mu}})(\hat{\boldsymbol{\mu}}_k - \bar{\boldsymbol{\mu}})^T$, where $\bar{\boldsymbol{\mu}} = \sum_{i=1}^{N} \mathbf{x}_i/N$.
  - Within class variance: $\mathbf{W} = \hat{\Sigma} = \sum_{k=1}^{K} \sum_{g_i=k} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T$.
- Think what is $\mathbf{B} + \mathbf{W} = ?$.

Naive Bayes Model

Suppose we estimate the class densities $f_1(X)$ and $f_2(X)$ for the features in class 1 and 2 respectively.

Bayes Formula tells us how to convert these to class posterior probabilities:

$$\Pr(Y = 1|X) = \frac{f_1(X)\pi_1}{f_1(X)\pi_1 + f_2(X)\pi_2},$$

where $\pi_1 = \Pr(Y = 1)$ and $\pi_2 = 1 - \pi_1$.

Since X is often high dimensional, the following *independence* model is convenient:

$$f_j(X) \approx \prod_{m=1}^{p} f_{jm}(X_m)$$

Works for more than two classes as well.

The Naive Bayes Classifier

The naive Bayes model assumes that given a class $G = j$, the features X_k are independent: $f_j(X) = \prod_{k=1}^{p} f_{jk}(X_k)$.

$$\begin{aligned}
\log \frac{\Pr(G = l|X)}{\Pr(G = J|X)} &= \log \frac{\pi_l f_l(X)}{\pi_J f_J(X)} \\
&= \log \frac{\pi_l \prod_{k=1}^{p} f_{lk}(X)}{\pi_J \prod_{k=1}^{p} f_{Jk}(X)} \\
&= \log \frac{\pi_l}{\pi_J} + \sum_{k=1}^{p} \log \frac{f_{lk}(X)}{f_{Jk}(X)} \\
&= \alpha_l + \sum_{k=1}^{p} g_{lk}(X_k)
\end{aligned}$$

Remark: It is an additive model.

Mixture Models for Classification

- Gaussian mixture model:

$$f(x) = \sum_{m=1}^{M} \alpha_m \phi(x; \mu_m, \Sigma_m), \quad \sum_m \alpha_m = 1$$

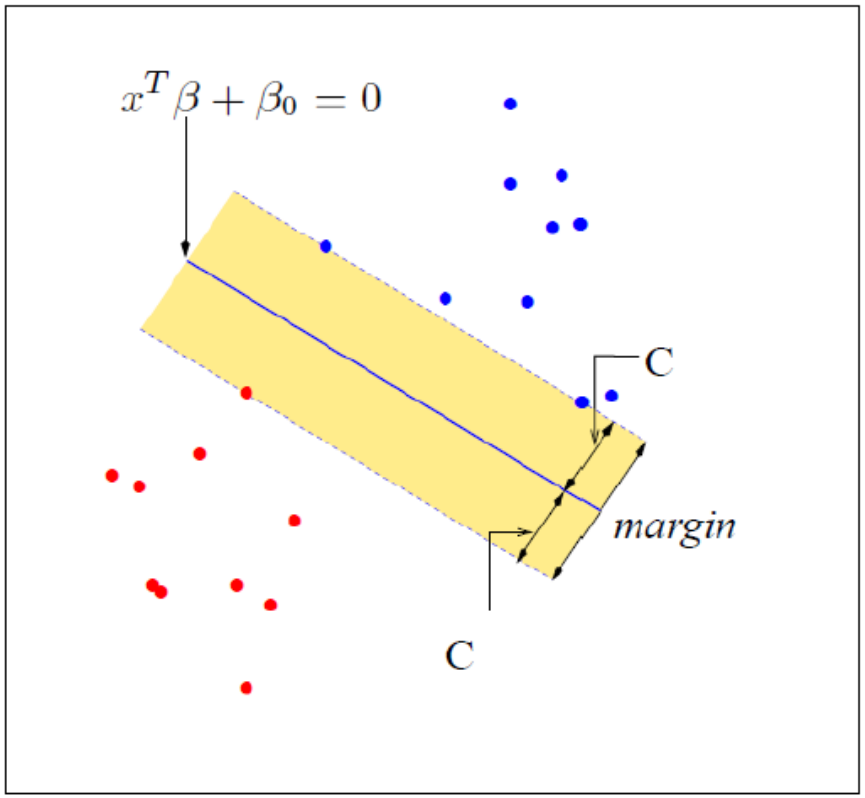
  $\phi(\cdot)$ Gaussian density.
  - If $\Sigma_m = \sigma \mathbf{I}$, and $M \to N$, then the maximum likelihood estimate approaches the kernel density where $\hat{\alpha}_m = 1/N$ and $\hat{\mu}_m = x_m$.
- Density estimation: probability of the observation i belongs to component m,

$$\hat{r}_{im} = \frac{\hat{\alpha}_m \phi(x_i; \hat{\mu}_m, \hat{\Sigma}_m)}{\sum_{k=1}^{M} \hat{\alpha}_k \phi(x_i; \hat{\mu}_k, \hat{\Sigma}_k)}$$

Support Vector Machines (SVM): Two-class Classification

- Coded a response variable Y taking values $+1$ or -1. The feature vector $X = (X_1, X_2, \ldots, X_p)$.
- The objective to build a classification rule $G(X)$ to assign a class label to an individual with feature X.
- A sample of pairs $(Y_i, x_i), i = 1, \ldots, N$. Try to estimates $G(X)$ *directly*.

The Support Vector Classifier



- Support vector classifier in the separable case. The decision boundary is the solid line, while broken lines bound the shaded maximal margin of width 2C.

Find the Hyperplane: Separable Case

- When there are no overlap between classes (separable),

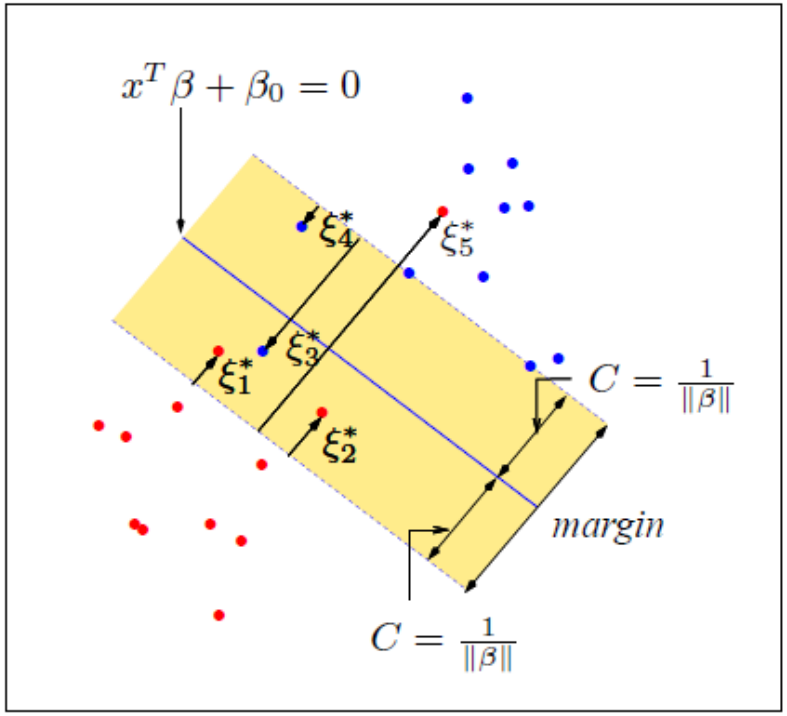
$$\max_{\beta,\beta_0,||\beta||=1} M$$
$$\text{subject to } y_i(x_i^T\beta + \beta_0) \geq M, \quad i = 1, \ldots, N$$

or equivalently

$$\min_{\beta,\beta_0} ||\beta||$$
$$\text{subject to } y_i(x_i^T\beta + \beta_0) \geq 1, \quad i = 1, \ldots, N.$$

- It is maximum margin classifier: find the linear decision boundary that creates the biggest gap between the blue points $(+1)$ and red points (-1).
- Underlying assumption: if the gap is big on the training data, it will also be big on any future test data.

Overlapping Case and Soft Margins



SVM in nonseparable (overlap) case. The points labeled ξ_j^* are on the wrong side of their margin by an amount $\xi_j^* = M\xi_j$; points on the correct side have $\xi_j^* = 0$.

Find the Hyperplane: Overlap Case

- When there are overlapping between classes,

$$\max_{\beta,\beta_0,||\beta||=1} M$$
$$\text{subject to } y_i(x_i^T\beta+\beta_0) \geq M(1-\xi_i),\ i=1,\ldots,N$$
$$\xi_i \geq 0, \sum \xi_i \leq B,$$

  which is equivalent to

$$\min ||\beta||$$
$$\text{subject to } y_i(x_i^T\beta+\beta_0) \geq 1-\xi_i\ \forall i$$
$$\xi_i \geq 0, \sum \xi_i \leq B$$

- Soft margin classifier: find the decision boundary that creates the biggest gap between the blue points $(+1)$ and red points (-1), but allowing a total budget B of cumulative overlap. i.e. $\sum_{i=1}^{N} \xi_i \leq B$.

Computation of SVM

To find the support vector classifier, we need to solve:

$$\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to } \xi_i \geq 0,\ \ y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i, \forall i.$$

The Lagrange (primal) function is

$$L_p = \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(x_i^T\beta + \beta_0) - (1-\xi_i)] - \sum_{i=1}^{N}\mu_i\xi_i,$$

which we minimize w.r.t. β, β_0 and ξ_i. Setting the respective derivatives to zero, we get

$$\beta = \sum_{i=1}^{N}\alpha_i y_i x_i, \quad 0 = \sum_{i=1}^{N}\alpha_i y_i, \quad \alpha_i = C - \mu_i, \forall i,$$

as well as the positivity constraints $\alpha_i,\ \mu_i,\ \xi_i \geq 0 \quad \forall i.$

Computing the Support Vector Classifier

Dual problem

$$\max_{\alpha_i} L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \alpha_i \alpha_{i'} y_i y_{i'} x_i^T x_{i'}\},$$

$$\text{subject to } 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0.$$

KKT condition

$$\begin{aligned} \alpha_i[y_i(x_i^T \beta + \beta_0) - (1 - \xi_i)] &= 0, \\ \mu_i \xi_i &= 0, \\ y_i(x_i^T \beta + \beta_0) - (1 - \xi_i) &\geq 0, \end{aligned}$$

for $i = 1, 2, \ldots, N$.

Solution

$$\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i, \quad \hat{G}(x) = \text{sign}[x^T \hat{\beta} + \hat{\beta}_0]$$

with nonzero coefficients $\hat{\alpha}_i$ only for those observations i for which the first constraint in KKT condition is exactly met.

KKT Condition

Given general problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \leq 0, \quad i = 1, \ldots m \\ & \ell_j(x) = 0, \quad j = 1, \ldots r \end{aligned}$$

The **Karush-Kuhn-Tucker conditions** or **KKT conditions** are:

- $0 \in \partial f(x) + \sum_{i=1}^{m} u_i \partial h_i(x) + \sum_{j=1}^{r} v_j \partial \ell_j(x)$ (stationarity)
- $u_i \cdot h_i(x) = 0$ for all i (complementary slackness)
- $h_i(x) \leq 0, \ \ell_j(x) = 0$ for all i, j (primal feasibility)
- $u_i \geq 0$ for all i (dual feasibility)

General SVM in Enlarged Feature Space

- Enlarged feature vectors $\boldsymbol{h}(x) = (h_1(x), \ldots, h_q(x))$.
- SVM classifier using input features

$$\hat{f}(x) = \boldsymbol{h}(x)^T \hat{\beta} + \hat{\beta}_0, \quad \hat{G}(x) = \text{sign}(\hat{f}(x)).$$

- Use inner product to replace:

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \alpha_i \alpha_{i'} y_i y_{i'} < \boldsymbol{h}(x_i), \boldsymbol{h}(x_{i'}) > .\},$$

$$f(x) = \boldsymbol{h}(x)^T \beta + \beta_0$$

$$= \sum_{i=1}^{N} \alpha_i y_i < \boldsymbol{h}(x), \boldsymbol{h}(x_i) > + \beta_0$$

Kernel Trick

- The transformation $\boldsymbol{h}(x)$ is only involved through inner product. In fact we don't need to specify the transformation $\boldsymbol{h}(x)$ at all, but only require knowledge of inner product.
- How to define the inner product?

$$K(x, x') =< \boldsymbol{h}(x), \boldsymbol{h}(x') > .$$

K should be a symmetric positive definite function.

Theorem

The Representer Theorem: Let $\mathcal{X}$ be a set of endowed with a kernel K, and $S = \{x_1, x_2, \ldots, x_n\} \subset \mathcal{X}$ a finite set of objects. Let $\Phi : \mathbb{R}^{n+1} \to \mathcal{R}$ be a function of $n + 1$ argument, strictly monotonic increasing in its last argument. Then any solution of the problem

$$\min_{f \in \mathcal{H}_K} \Phi(f(x_1), \ldots, f(x_n), ||f||_{\mathcal{H}_K}),$$

where $(\mathcal{H}_K, ||\cdot||_{\mathcal{H}_K})$ is the reproducing kernel Hilbert space associated with K, admits a representation of the form

$$\forall x \in \mathcal{X}, \quad f(x) = \beta_0 + \sum_{i=1}^{N} \alpha_i K(x_i, x).$$

Kernel Functions

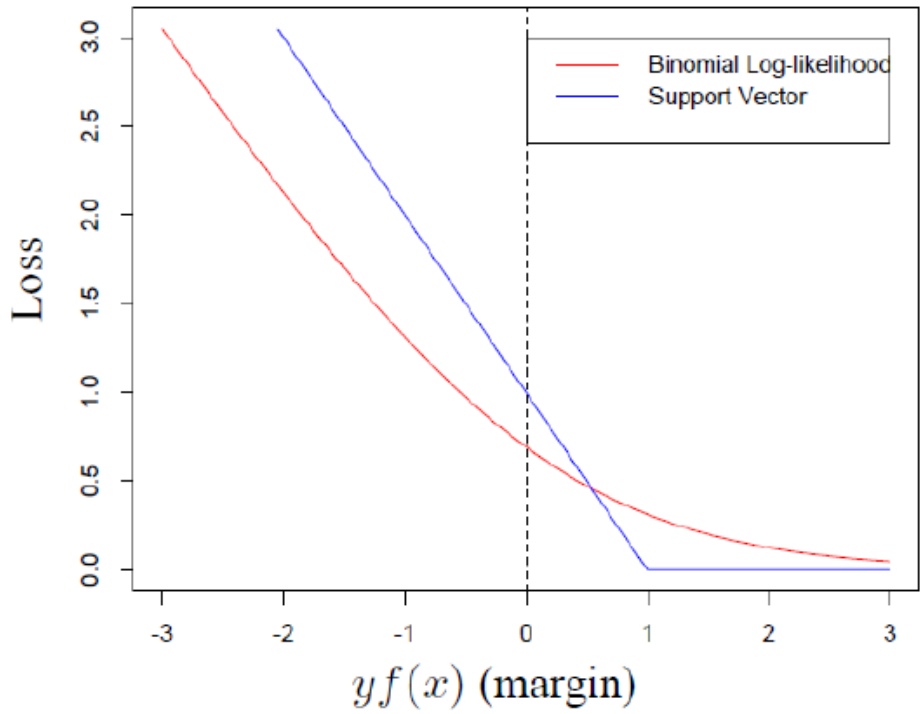
Some kennel functions:

$$K(x, x') = (1+ < x, x' >^d$$
$$K(x, x') = \exp(-\gamma||x - x'||^2)$$
$$K(x, x') = \tanh(\kappa_1 < x, x' > +\kappa_2).$$

Solution can be written as:

$$\hat{f} = \sum_{i=1}^{N} \hat{\alpha}_i y_i K(x, x_i) + \hat{\beta}_0.$$

SVM via "Loss + Penalty"



With $f(x) = x^T\beta + \beta_0$ and $y_i \in \{-1, 1\}$, consider

$$\min_{\beta_0,\,\beta} \sum_{i=1}^{N} [1 - y_i f(x_i)]_+ + \lambda \|\beta\|^2$$

This *hinge loss* criterion is equivalent to the SVM, with λ monotone in B.

Compare with

$$\min_{\beta_0,\,\beta} \sum_{i=1}^{N} \log\left[1 + e^{-y_i f(x_i)}\right] + \lambda \|\beta\|^2$$

This is *binomial deviance loss*, and the solution is "ridged" linear logistic regression.

The SVM and Kernels

- Recall Ch3 on the spline and kernel method using the "kernel" trick.
- With $f(x) = h(x)^T\beta + \beta_0$

$$\min_{\beta_0,\beta} \sum_{i=1}^{N} [1 - y_i f(x_i)]_+ + \frac{\lambda}{2}||\beta||^2$$

- With $f(x) = \beta_0 + \sum_{i=1}^{N} \alpha_i K(x, x_i)$ and $\beta = \boldsymbol{H}\alpha$,

$$\min_{\alpha_0,\alpha} \sum_{i=1}^{N} [1 - y_i f(x_i)]_+ + \frac{\lambda}{2}\alpha^T \mathbf{K}\alpha$$

R Codes

```
library(e1071)
library(vcd)
data(spam, package = "ElemStatLearn")
str(spam)
model = svm(spam ~ ., data = spam)
summary(model)
pred = fitted(model)
obs = spam$spam
table(pred, obs)
model = svm(spam ~ ., data = spam, cross = 10)
summary(model)
mosaic(table(pred, obs), shade = TRUE)
```

Kernel Machines

- The kernel representation $f(x) = \beta_0 + \sum_{i=1}^{N} \alpha_i K(x, x_i)$ has many applications:
  - Regression and generalized regressions: the logistic regression model

$$\log\left(\frac{P(Y=+1|x)}{P(Y=-1|x)}\right) = \beta_0 + \sum_{i=1}^{N} \alpha_i K(x, x_i).$$

- Kernel principal components and canonical correlation analysis.
- Kernel discriminant analysis.
- Any linear model can be kernelized.

Kernel Logistic Regression (KLR) vs SVM

- Kernel Logistic Regression (KLR) has very similar classification performance as SVM.
- KLR has limiting optimal margin properties like SVM.
- KLR provides estimates of the class probabilities. Often these are more useful than the classifications.
- Generalizes naturally to Multi-class classification through kernel multi-logit regression.
- KLR is more expensive to compute than SVM: $O(N^3)$ vs $O(N^2|S|)$.

SVM for Multi-class Classification

- SVMs do not generalize gracefully when the number of classes K are more than 2.
- Typically solve using the one vs rest method, or the one vs one method (all pairs).
- Despite issues, it is a very popular method for multi-class classification.

KLR for Multi-class Classification

- This is an easy generalization of two-class logistic regression:

$$Pr(Y = j|x) = \frac{e^{f_j(x)}}{e^{f_1(x)} + \ldots + e^{f_M(x)}}$$

  with $\sum_m f_m(x) = 0$.
- $f_m(x)$ can be linear: $f_m(X) = \beta_{0m} + \sum_{j=1}^{p} \beta_{jm} x_j$, or nonlinear through kernels: $f_m(x) = \beta_{0m} + \sum_{i=1}^{N} \alpha_{im} K(x, x_i)$, where x_i is the ith vector of variables.
- We regularizes these models exactly like we do SVMs.
- Performance similar to SVMs, but model is more interpretable in terms of probabilities.

Caveats for Kernel Machines

- Kernel methods do not do variable selection in any adaptive or automatic way.
- Potentially suffer if the number of features is large, and a large fraction of them are garbage.
- For linear SVMs, recursive feature selection is possible. Similar to stepwise deletion in regression.

Remarks

Thank you!

- **Questions and Comments?**