

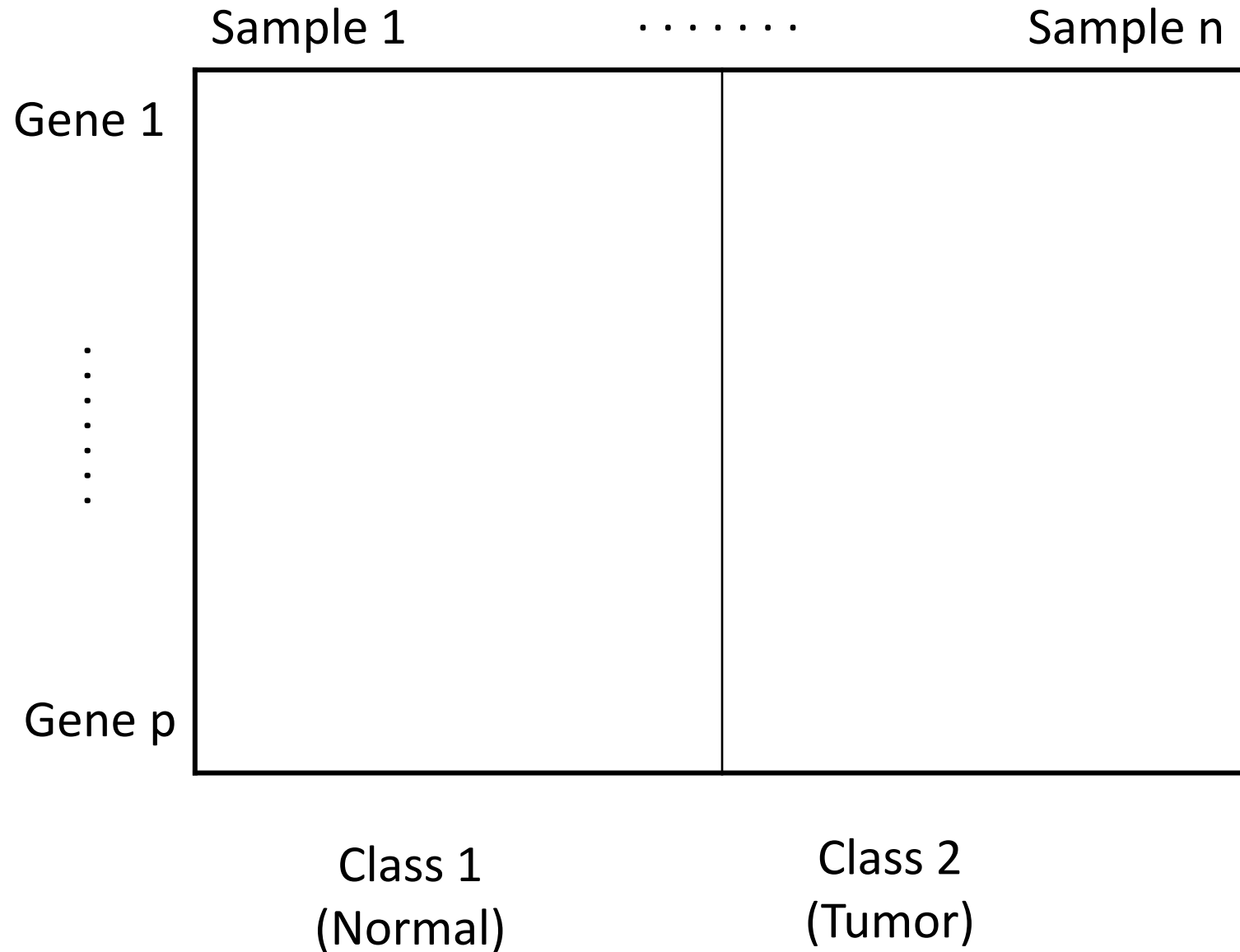
第6-4章: Classification and Prediction

- Bayesian decision rule
- Fisher linear discriminant analysis
- SVM
- Aggregating classifiers
- Reference
 - T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, E. S. Lander. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science 15 October 1999: Vol. 286 no. 5439 pp. 531-537.
 - Ji Zhu and Trevor Hastie. Classification of gene microarrays by penalized logistic regression. Biostatistics (2004), 5, 3, pp. 427-443.

Classification

- **Task:** assign objects to classes (groups) on the basis of measurements made on the objects
- **Unsupervised:** classes unknown, want to discover them from the data (cluster analysis)
- **Supervised:** classes are predefined, want to use a (training or learning) set of labeled objects to form a classifier for classification of future observations

Supervised Classification (Two Classes)



Example: Tumor Classification

- Reliable and precise classification essential for successful cancer treatment
- Current methods for classifying human malignancies rely on a variety of morphological, clinical and molecular variables
- Characterize molecular variations among tumors by monitoring gene expression (microarray)
- Hope: that microarrays will lead to more reliable tumor classification (and therefore more appropriate treatments and better outcomes)

Tumor Classification Using Array Data

Three main types of statistical problems associated with tumor classification:

- Identification of new/unknown tumor classes using gene expression profiles (**unsupervised learning – clustering**)
- Classification of malignancies into known classes (**supervised learning – discrimination**)
- Identification of “marker” genes that characterize the different tumor classes (**feature or variable selection**).

Classifiers

- A **predictor** or **classifier** partitions the space of gene expression profiles into K disjoint subsets, A_1, \dots, A_K , such that for a sample with expression profile $\mathbf{X}=(X_1, \dots, X_G) \in A_k$ the predicted class is k
- Classifiers are built from a **learning set (LS)**
$$L = (\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)$$
- **Classifier** C built from a learning set L :
$$C(\cdot, L): \mathbf{X} \rightarrow \{1, 2, \dots, K\}$$
- **Predicted class** for observation \mathbf{X} :
$$C(\mathbf{X}, L) = k \text{ if } \mathbf{X} \text{ is in } A_k$$

Bayes公式

- 后验概率: $p(\omega_k|\vec{x})$
- 先验概率: π_k , 其中 $\sum \pi_k=1$.
- Bayes公式

$$p(\omega_k|\vec{x}) = \frac{p(\vec{x}|\omega_i)\pi_i}{\sum_{l=1}^K p(\vec{x}|\omega_l)\pi_l}$$

Bayes决策

- 将观测向量 \vec{x} 归入后验概率最大的类，即

$$\hat{\omega} = \underset{k}{\operatorname{Argmax}} p(\omega_k | \vec{x})$$

- 利用Bayes公式，上式等价于

$$\hat{\omega} = \underset{k}{\operatorname{Argmax}} p(\vec{x} | \omega_k) p(\omega_k)$$

最小错误Bayes决策

- 错误函数

$$\Pr(\text{error}) = \sum_{i=1}^K \Pr(\text{error}|\omega_i)p(\omega_i)$$

- 单个类别的错误函数

$$\Pr(\text{error}|\omega_i) = \int_{\overline{\Omega_i}} \Pr(\vec{x}|\omega_i)d\vec{x}$$

其中 Ω_i 为归入类别 ω_i 的区域。

最小错误Bayes决策

$$\begin{aligned}\Pr(\text{error}) &= \sum_{i=1}^K \int_{\overline{\Omega}_i} p(\vec{x}|\omega_i) p(\omega_i) d\vec{x} \\ &= \sum_{i=1}^K p(\omega_i) \left(1 - \int_{\Omega_i} p(\vec{x}|\omega_i) d\vec{x}\right) \\ &= 1 - \sum_{i=1}^K p(\omega_i) \int_{\Omega_i} p(\vec{x}|\omega_i) d\vec{x}\end{aligned}$$

最小错误Bayes决策

- 因此,为了使 $p(\text{error})$ 最小,只要选择区域 Ω_i ,使得正确分类概率最大

$$\sum_{i=1}^K p(\omega_i) \int_{\Omega_i} p(\vec{x}|\omega_i) d\vec{x}$$

- Bayes决策使得正确分类概率最大

$$c = \int \max_i p(\omega_i) p(\vec{x}|\omega_i) d\vec{x}$$

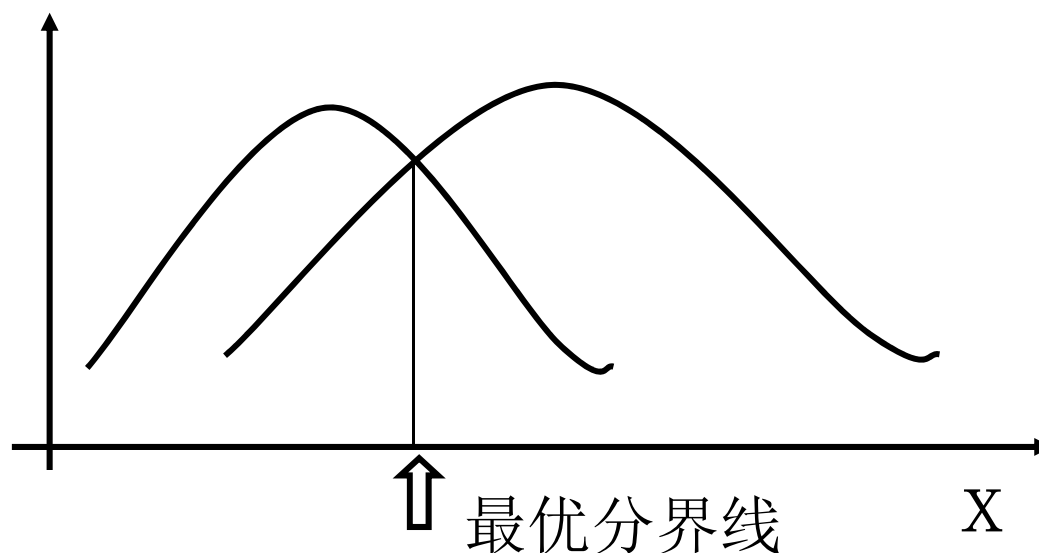
- Bayes错分概率为

$$e_B = 1 - c = 1 - \int \max_i p(\omega_i) p(\vec{x}|\omega_i) d\vec{x}$$

两类问题Bayes决策

- 对于一维观测量 x , 最小错误率对应于如下的最优分界线下的错误率,

$$e_B = p(\omega_2) \int_{\Omega_1} p(x|\omega_2)dx + p(\omega_1) \int_{\Omega_2} p(x|\omega_1)dx$$



正态分布判别

- 对多元正态分布

$$\Pr(\vec{x}|\omega_i) = \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (\vec{x} - \vec{\mu}_i)^T \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i) \right\}$$

- 按照Bayes原则,

$$\begin{aligned} \log(\Pr(\omega_i|\vec{x})) &= \log(\Pr(\vec{x}|\omega_i)) + \log(\Pr(\omega_i)) - \log(\Pr(x)) \\ &= -\frac{1}{2} (\vec{x} - \vec{\mu}_i)^T \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i) - \frac{1}{2} \log(|\Sigma_i|) \\ &\quad - \frac{p}{2} \log(2\pi) + \log(\Pr(\omega_i)) - \log(\Pr(x)) \end{aligned}$$

正态分布判别

- 二次判别函数

$$g_i(x) = \log(\Pr(\omega_i)) - \frac{1}{2} \log(|\Sigma_i|) - \frac{1}{2}(\vec{x} - \vec{\mu}_i)^T \Sigma_i^{-1}(\vec{x} - \vec{\mu}_i)$$

- 判别规则是:若对所有的 $j \neq i$ 有 $g_i > g_j$,则将样本归入 ω_i .

正态分布判别分界面

$$g_{ij}(\vec{x}) = 0$$

$$\begin{aligned} g_{ij}(\vec{x}) &= g_i(\vec{x}) - g_j(\vec{x}) \\ &= \log\left(\frac{\Pr(\omega_i)}{\Pr(\omega_j)}\right) - \frac{1}{2} \log\left(\frac{|\Sigma_i|}{|\Sigma_j|}\right) \\ &\quad - \frac{1}{2}(\vec{x} - \vec{\mu}_i)^T \Sigma_i^{-1}(\vec{x} - \vec{\mu}_i) \\ &\quad + \frac{1}{2}(\vec{x} - \vec{\mu}_j)^T \Sigma_j^{-1}(\vec{x} - \vec{\mu}_j) \end{aligned}$$

从数据估计参数

- 设有一组样本 $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n\}$ 服从正态分布特征 $\vec{\theta} = (\vec{\mu}, \Sigma)$ 那么似然函数

$$L(\vec{x}_1, \dots, \vec{x}_n | \vec{\theta}) = \prod_{i=1}^n \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (\vec{x}_i - \vec{\mu})^T \Sigma^{-1} (\vec{x}_i - \vec{\mu}) \right\}$$

- 求 $\log(L)$ 对 θ 的微分得到,

$$\frac{\partial \log(L)}{\partial \mu} = \frac{1}{2} \sum_{i=1}^n (\Sigma^{-1} + (\Sigma^{-1})^T) (\vec{x}_i - \vec{\mu})$$

$$\frac{\partial \log(L)}{\partial \Sigma} = -\frac{n}{2} \Sigma^{-1} + \frac{1}{2} \sum_{i=1}^n (\vec{x}_i - \vec{\mu})(\vec{x}_i - \vec{\mu})^T \Sigma^{-1} \Sigma^{-1}$$

从数据估计参数

- 这里用到公式

$$\frac{\partial |A|}{\partial A} = |A|(A^{-1})^T$$

- 于是,极大似然估计为

$$\vec{\mu} = \frac{1}{n} \sum_{i=1}^n \vec{x}_i$$

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^n (\vec{x}_i - \vec{\mu})(\vec{x}_i - \vec{\mu})^T$$

正态分布二次判别函数

- 用估计值带入到判别函数,得到二次判别函数 (有时甚至用每个类型数据个数所占比例来估计先验概率)

$$g_i(x) = \log(p(\omega_i)) - \frac{1}{2} \log(|\hat{\Sigma}_i|) - \frac{1}{2} (\vec{x} - \vec{\mu}_i)^T \hat{\Sigma}_i^{-1} (\vec{x} - \vec{\mu}_i)$$

正态分布线性判别函数

- 特别地,假定类协方差相等时,判别函数简化为线性函数,

$$g_i(x) = \log(\Pr(\omega_i)) - \frac{1}{2} \vec{\mu}_i^T S_W^{-1} \vec{\mu}_i + \vec{x}^T S_W^{-1} \vec{\mu}_i$$

- 其中 S_W 为共群协方差矩阵,其估计为

$$S_W = \sum_{i=1}^K \frac{n_i}{n} \hat{\Sigma}_i$$

正态分布两类判别函数

- 对于两类问题,当类协方差相等时,判别函数有更简单的形式,

如果 $\vec{W}^T \vec{x} + W_0 > 0$ 样本归入 ω_1 ; 否则归入 ω_2 ,
其中

$$W = S_W^{-1}(\vec{\mu}_1 - \vec{\mu}_2)$$

$$W_0 = -\log\left(\frac{\Pr(\omega_2)}{\Pr(\omega_1)}\right) - \frac{1}{2}(\vec{\mu}_1 + \vec{\mu}_2)^T \vec{W}$$

判别分析

- 最初由 R.A. Fisher提出.



Fisher判别分析

寻找一个投影方向，使得在该方向上

- 极小化组内距离。
- 极大化组间距离。

➔ 基本的目标在于提取最有效的分类特征

Fisher判别分析：分散矩阵

- 组间距

$$S_B = \sum_{i=1}^K \frac{n_i}{n} (\mu_i - \mu)(\mu_i - \mu)^T$$

- 组内距

$$S_W = \sum_{i=1}^K \frac{n_i}{n} \hat{\Sigma}_i$$

投影方向上的分散矩阵

- 投影变换 $\mathbf{u} = \alpha^T \mathbf{x}$
- 投影后组间距 $S_B(\mathbf{u}) = \alpha^T S_B \alpha$.
- 投影后组内距 $S_W(\mathbf{u}) = \alpha^T S_W \alpha$.

Fisher判别分析

- Fisher准则：投影后的组间距和组内距比值最大，

$$J_F(\alpha) = \frac{\alpha^T S_B \alpha}{\alpha^T S_W \alpha}$$

$$\hat{\alpha} = \underset{\alpha}{\operatorname{Argmax}} J_F(\alpha)$$

投影方向求解

- 为保证唯一性，可以要求 α 满足，

$$\alpha^T S_W \alpha = 1$$

- 采用Lagrange乘子法, 变为下面的函数的优化问题

$$L(\alpha, \lambda) = \alpha^T S_B \alpha - \lambda(\alpha^T S_W \alpha - 1)$$

投影方向求解

- 目标函数对 α, λ 微分,

$$\frac{\partial L}{\partial \alpha} = 2(S_B - \lambda S_W)\alpha = 0$$

$$\frac{\partial L}{\partial \lambda} = -\alpha^T S_W \alpha + 1 = 0$$

- 得方程

$$S_B \alpha = \lambda S_W \alpha$$

$$\alpha^T S_W \alpha = 1$$

投影方向求解

- 求解特征值问题

$$S_W^{-1} S_B \alpha = \lambda \alpha$$

- S_B 的秩最大为K-1，可以最多提取K-1个特征向量。

两类问题的Fisher判别分析

- 两类时的Fisher准则,

$$J_F(\alpha) = \frac{|\alpha^T(\mu_1 - \mu_2)|^2}{\alpha^T S_W \alpha}$$

- 对 $J_F(\alpha)$ 微分得,

$$\frac{\alpha^T(\mu_1 - \mu_2)}{\alpha^T S_W \alpha} \left\{ 2(\mu_1 - \mu_2) + \frac{\alpha^T(\mu_1 - \mu_2)}{\alpha^T S_W \alpha} S_W \alpha \right\} = 0$$

- 得

$$\alpha \propto S_W^{-1}(\mu_1 - \mu_2)$$

两类问题的Fisher判别分析

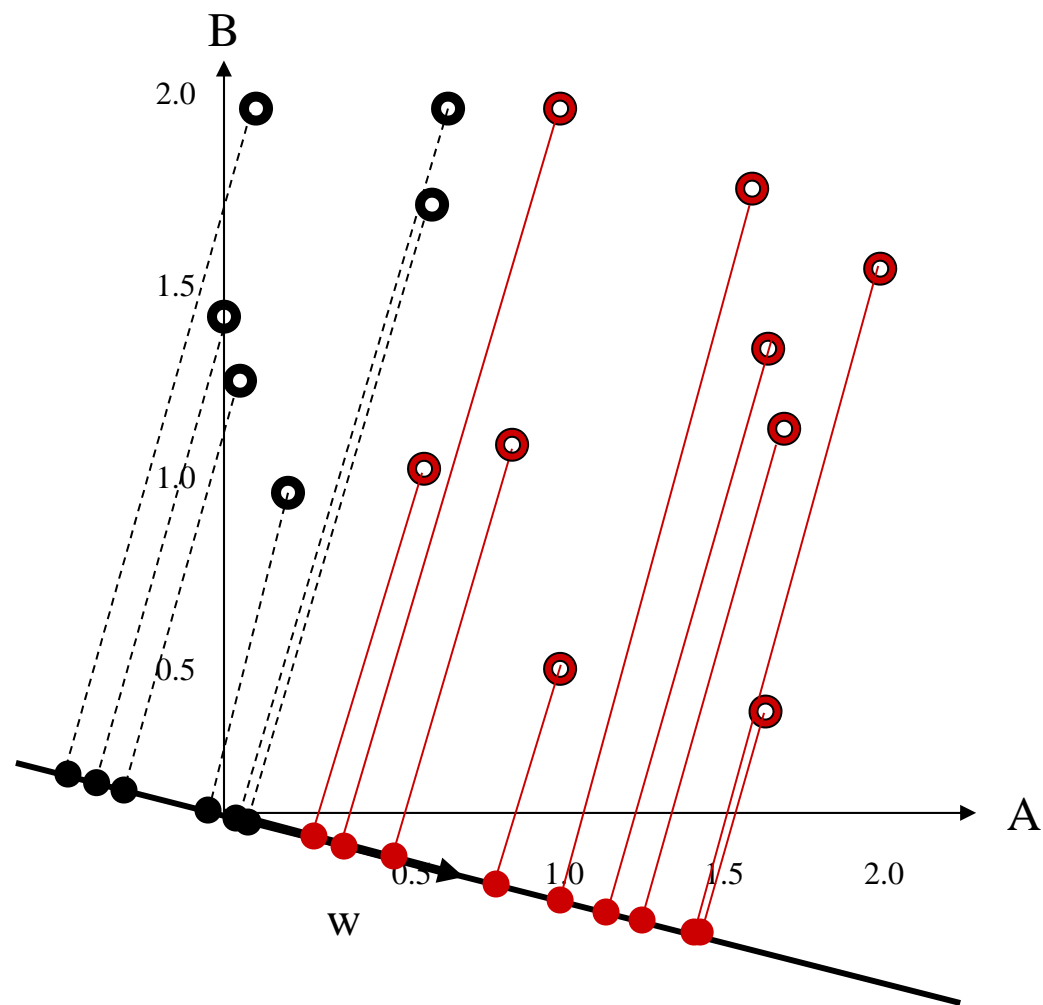
- 判别规则

$$(S_W^{-1}(\mu_1 - \mu_2))^T x + \alpha_0 > 0$$

- 和前面等协方差正态分布Bayes决策形式一样,有时候就取Bayes决策之阈值

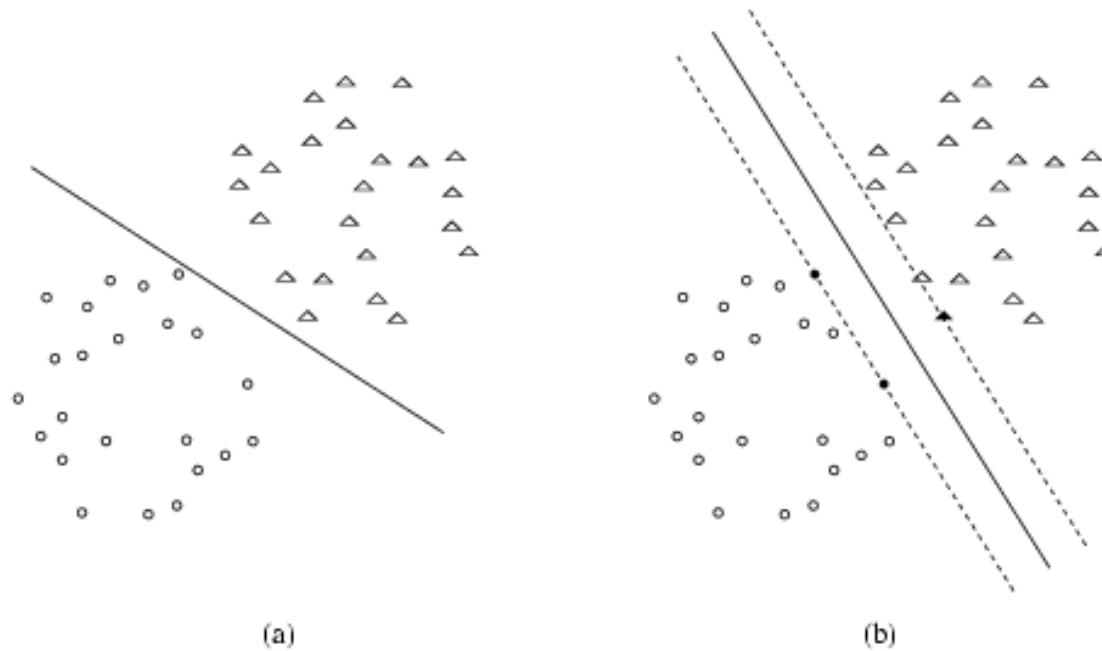
$$\alpha_0 = -\frac{1}{2}(\mu_1 + \mu_2)^T S_W^{-1}(\mu_1 - \mu_2) - \log\left(\frac{\Pr(\omega_2)}{\Pr(\omega_1)}\right)$$

两类问题的Fisher判别分析

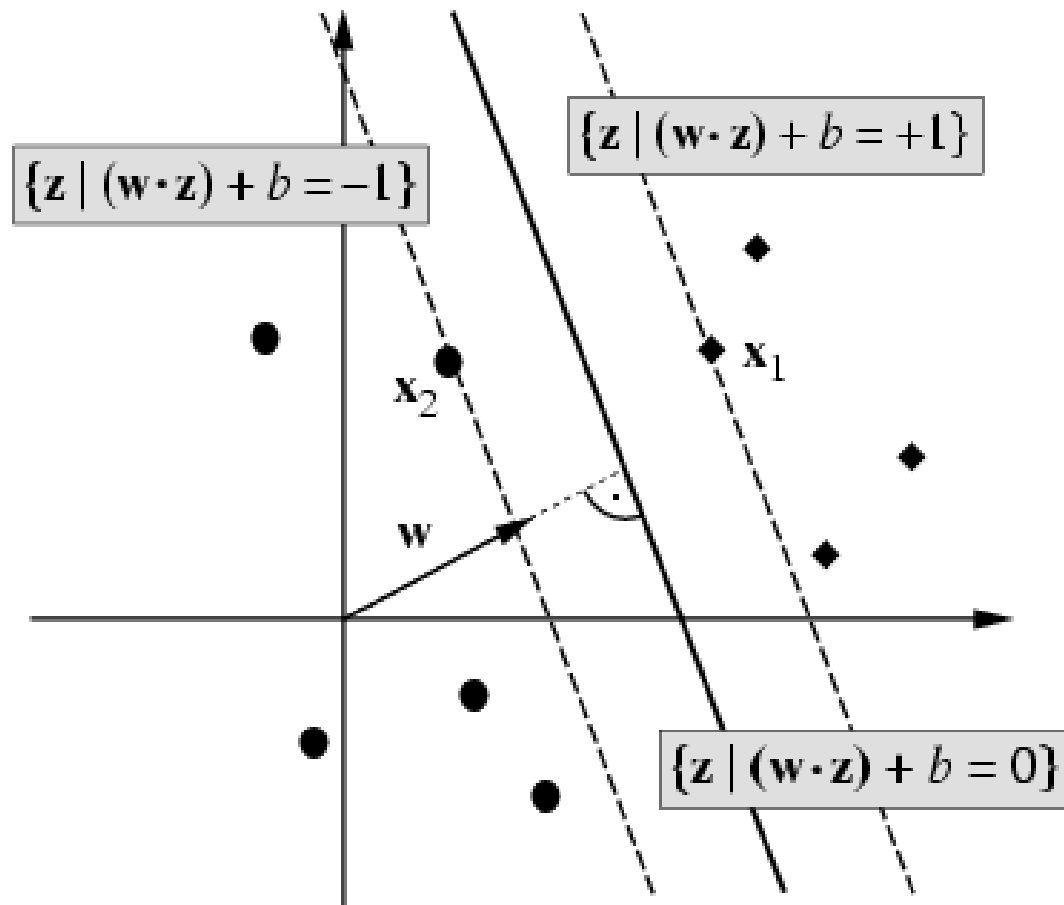


Find Optimal Hyperplane

- Assumption: samples are linearly separable.
- A better generalization is expected from (b).



Geometric Margin



Note:

$$(w \cdot z_1) + b = +1$$

$$(w \cdot z_2) + b = -1$$

$$\Rightarrow (w \cdot (z_1 - z_2)) = 2$$

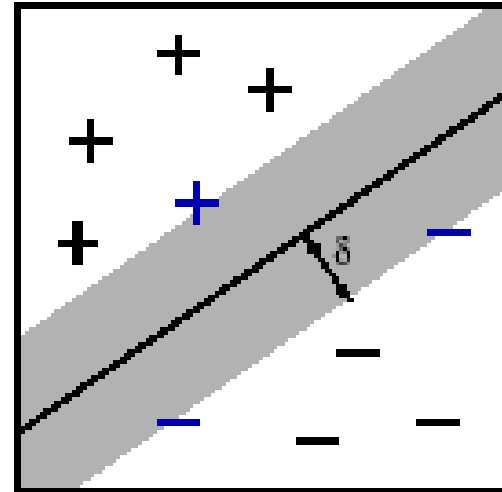
$$\Rightarrow \left(\frac{w}{\|w\|} \cdot (z_1 - z_2) \right) = \frac{2}{\|w\|}$$

Maximal Margin Separation

- Find hyperplane with the largest distance to closest samples.

$$\begin{cases} \text{Min :} & \frac{1}{2} \|\omega\|^2 \\ \text{s.t. :} & y_i (\omega \cdot z_i + b) \geq 1, \\ & i = 1, \dots, N. \end{cases}$$

Support vectors: samples with minimal distance.



Lagrange Theory with Inequality Constraints (Kuhn-Tucker)

- The primal problem with convex domain $\Omega \subset \mathbb{R}^n$ and $f \in C^1$ convex and g_i, h_i affine.

$$\begin{cases} \text{Min:} & f(\omega) \quad \omega \in \Omega, \\ \text{s.t.:} & g_i(\omega) \leq 0, i = 1, \dots, k, \\ & h_i(\omega) = 0, i = 1, \dots, m. \end{cases}$$

$$\iff \begin{cases} \text{Max:} & \inf_{\omega \in \Omega} L(\omega, \alpha, \beta) \\ \text{s.t.:} & \alpha \geq 0 \end{cases}$$

$$L(\omega, \alpha, \beta) = f(\omega) + \sum_{i=1}^k \alpha_i g_i(\omega) + \sum_{i=1}^m \beta_i h_i(\omega)$$

Lagrange theory with inequality constraints (Kuhn-Tucker)

i.e. Finding α^* , β^* such that

$$\left\{ \begin{array}{l} \frac{\partial L(\omega^*, \alpha^*, \beta)^*}{\partial \omega} = 0 \\ \frac{\partial L(\omega^*, \alpha^*, \beta)^*}{\partial \beta} = 0 \\ \alpha_i^* g_i(\omega^*) = 0, \quad i = 1, \dots, k, \\ g_i(\omega^*) \leq 0, \quad i = 1, \dots, k, \\ \alpha_i^* \geq 0, \quad i = 1, \dots, k, \end{array} \right.$$

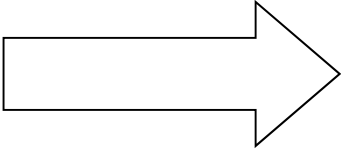
Solving the Optimization Problem

Lagrangian function

$$L(\omega, \alpha, \beta) = \frac{1}{2} \|\omega\|^2 - \sum_{i=1}^N \alpha_i (y_i [(\omega \cdot z_i + b) - 1])$$

Taking derivative,

$$\frac{\partial L(\omega, b, \alpha)}{\partial \omega} = 0, \quad \frac{\partial L(\omega, b, \alpha)}{\partial b} = 0,$$


$$\left\{ \begin{array}{l} \omega = \sum_{i=1}^N \alpha_i y_i z_i, \\ \sum_{i=1}^N \alpha_i y_i = 0. \end{array} \right.$$

Solving the Optimization Problem

- Substituting the above relations into the Lagrangian function to obtain,

$$\begin{aligned} L(\omega, \alpha, \beta) &= \frac{1}{2} \|\omega\|^2 - \sum_{i=1}^N \alpha_i (y_i [(\omega \cdot z_i + b) - 1]) \\ &= \frac{1}{2} \|\omega\|^2 + \sum_{i=1}^N \alpha_i y_i (\omega \cdot z_i) - b \sum_{i=1}^N \alpha_i y_i + \sum_{i=1}^N \alpha_i \\ &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j (z_i \cdot z_j) \end{aligned}$$

Dual Problem

$$\left\{ \begin{array}{ll} \text{Max:} & W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \alpha_i \alpha_j y_i y_j (z_i \cdot z_j) \\ \text{s.t.:} & \alpha_i \geq 0, \quad i = 1, \dots, N, \\ & \sum_{i=1}^N \alpha_i y_i = 0 \end{array} \right.$$

The Solutions

- Suppose α^* is the solution of the dual problem, then

$$\alpha_i^* [y_i(\omega \cdot z_i + b) - 1] = 0$$

- Only those support vectors z_i have no-zero α_i^* , and

$$\begin{cases} \omega^* = \sum_{i=1}^N \alpha_i^* y_i z_i, \\ b^* = -\frac{\max_{y_i=-1} \omega^* \cdot z_i + \min_{y_i=1} \omega^* \cdot z_i}{2}. \end{cases}$$

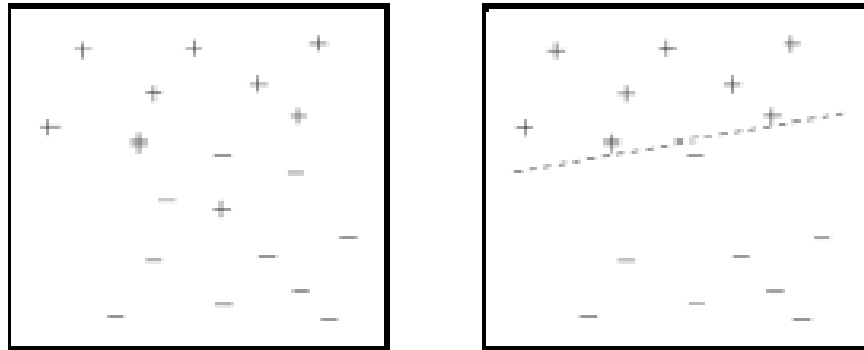
The Solutions

- To classify a new sample z

$$\begin{aligned} f(z) &= \text{sgn}(\omega^* \cdot z + b^*) \\ &= \text{sgn} \left(\sum_{i=1}^N \alpha_i^* y_i (z \cdot z_i) + b^* \right) \end{aligned}$$

Nonseparable Training Samples

- For some training samples, there is no separating hyperplane.
- Complete separation is suboptimal for many training samples.



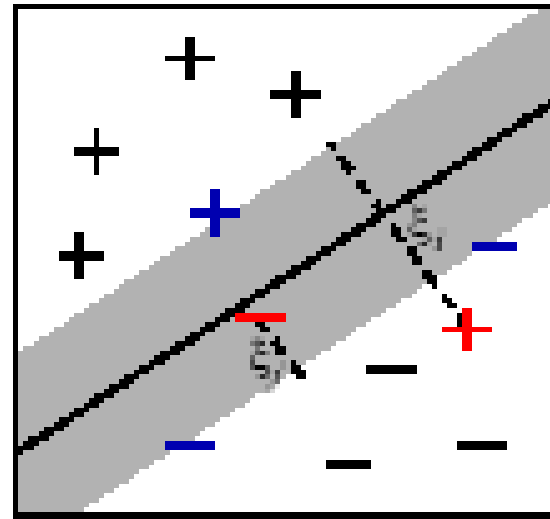
➡ Minimize trade-off between margin and training error.

Soft-margin Separation

By introducing N nonnegative variables ξ_1, \dots, ξ_N , such that

$$y_i(\omega \cdot z_i + b) \geq 1 - \xi_i, i = 1, \dots, N.$$

- Slack variables ξ_i measures by how much example (z_i, y_i) fails to achieve a target margin.
- ξ_i is an upper bound on the number of training errors.



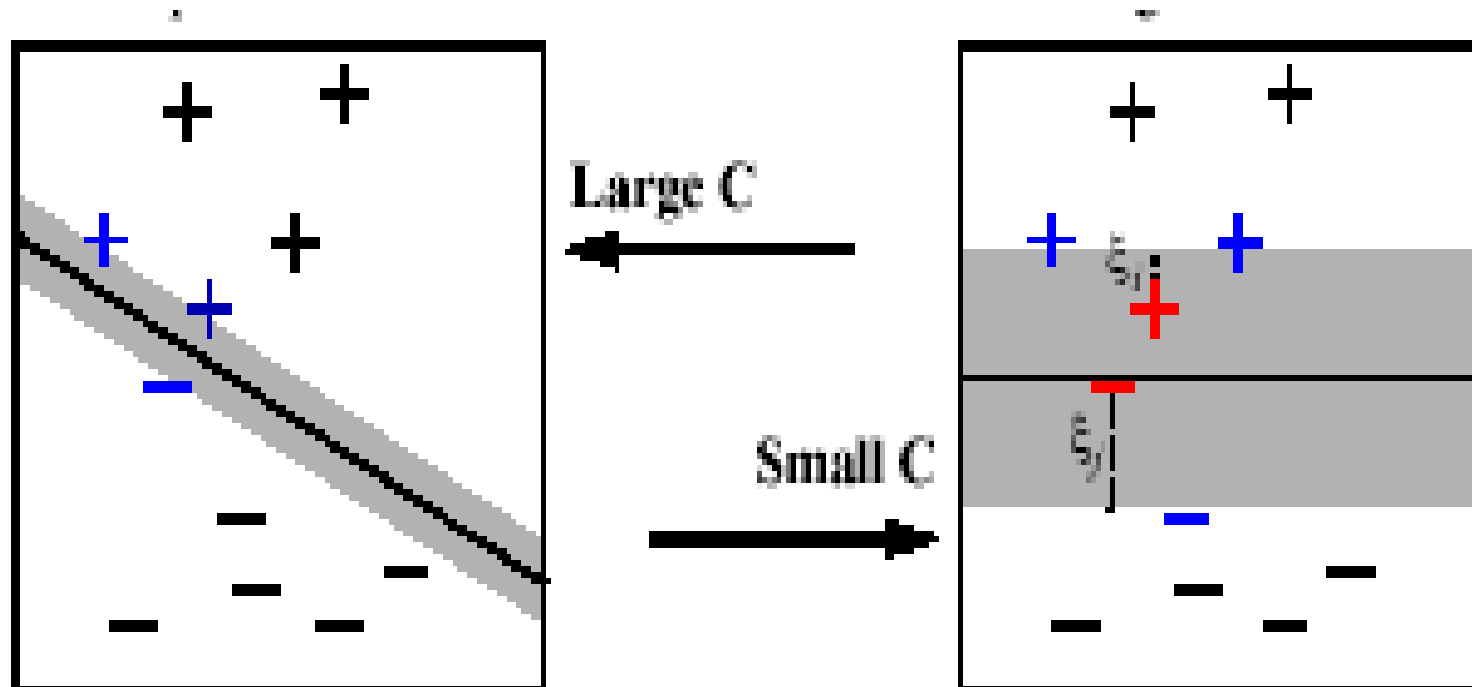
Soft-margin Separation

Maximize margin and minimize training error simultaneously (1-norm soft-margin).

$$\left\{ \begin{array}{l} \text{Min: } \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^N \xi_i, \\ \text{s.t.: } y_i(\omega \cdot z_i + b) \geq 1 - \xi_i, i = 1, \dots, N, \\ \quad \xi_i \geq 0, i = 1, \dots, N. \end{array} \right.$$

C is a parameter that controls trade-off between margin and training error.

Soft-margin Separation



Solving the Optimization Problem

$$L(\omega, b, \xi, \alpha, \gamma) = \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i [y_i(\omega \cdot z_i + b) - 1 + \xi_i] - \sum_{i=1}^N \gamma_i \xi_i$$

$$\frac{\partial L(\omega, b, \xi, \alpha, \gamma)}{\partial \omega} = 0, \frac{\partial L(\omega, b, \xi, \alpha, \gamma)}{\partial \xi} = 0, \frac{\partial L(\omega, b, \xi, \alpha, \gamma)}{\partial b} = 0.$$

$$\Rightarrow \begin{cases} \omega = \sum_{i=1}^N y_i \alpha_i z_i, \\ \alpha_i + \gamma_i = C, \quad i = 1, \dots, N, \\ \sum_{i=1}^N y_i \alpha_i = 0. \end{cases}$$

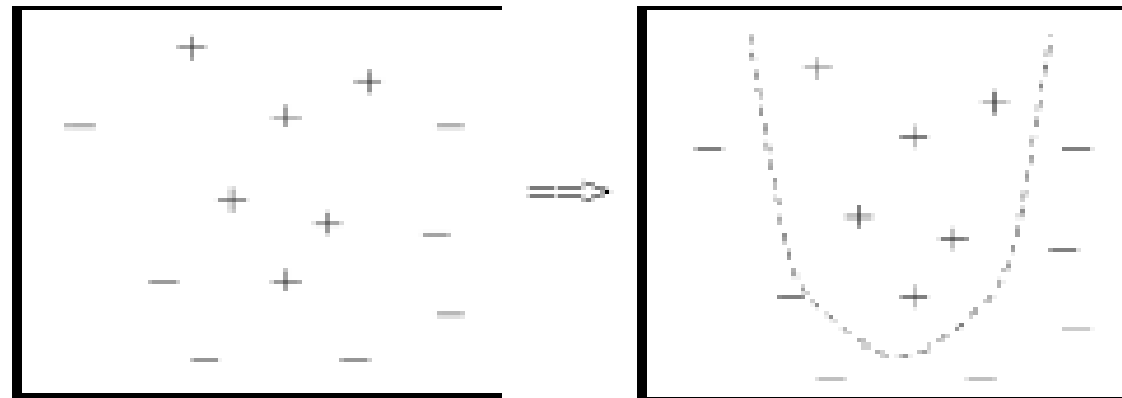
Dual Problem

$$\left\{ \begin{array}{l} \text{Max: } W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j z_i \cdot z_j, \\ \text{s.t.: } \sum_{i=1}^N y_i \alpha_i = 0, \\ \qquad \qquad 0 \leq \alpha_i \leq C, i = 1, \dots, N. \end{array} \right.$$

Nonlinear Problem

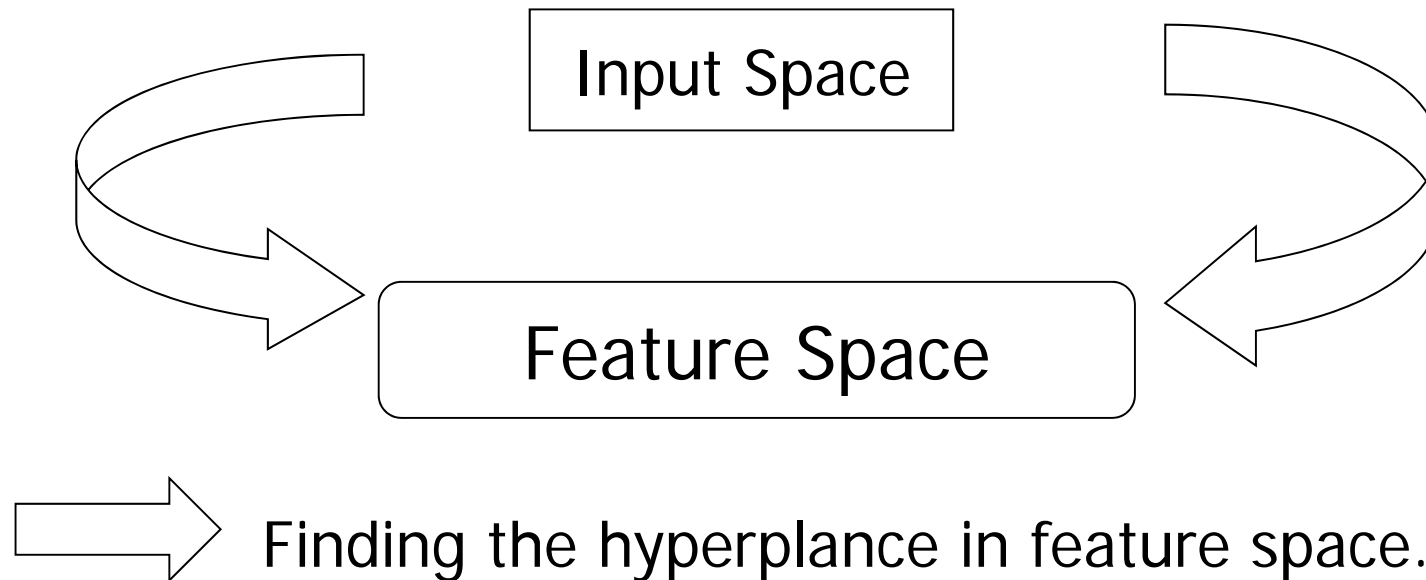
- Some tasks have non-linear structure.
- No hyperplane is sufficiently accurate.

How can SVMs learn nonlinear classification rules?



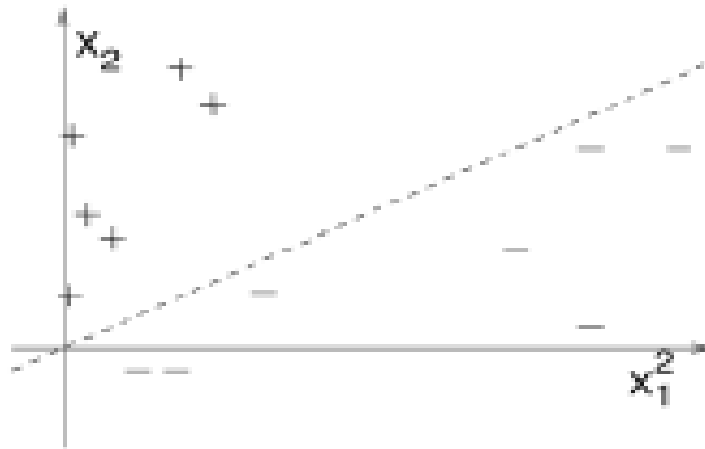
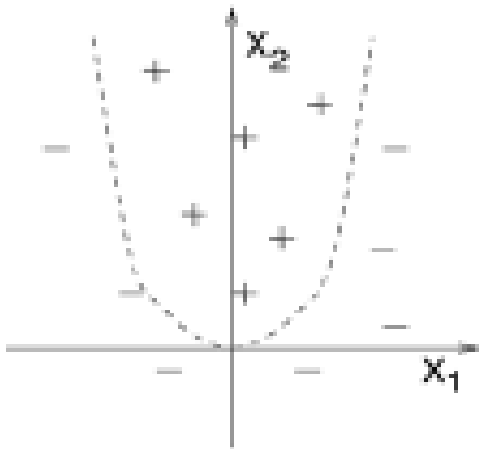
Extending the Hyperplane

- Mapping the input samples to feature space in higher dimension.



Example

- Input space with two attributes: (x_1, x_2) .
- Feature space with 6 attributes: $(x_1^2, x_2^2, x_1, x_2, x_1x_2, 1)$



Mercer's Theorem

- If K is a continuous kernel of a positive definite integral operator on $L_2(X)$ (where X is some compact space).

$$\int_x k(x, x') f(x) f(x') dx dx' \geq 0$$

- Then there exists a mapping Φ to a high dimension space H , such that $K(x, x')$ is the inner product in H .

Kernels

- The dual problem depends only on inner products, so we do not need to represent the feature space explicitly. Only the inner product in feature space need to be considered.

For example,

$$\Phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, 1)$$

gives the inner product in feature space,

$$K(a, b) = [a \cdot b + 1]^2 = \Phi(a) \cdot \Phi(b)$$

SVM With Kernels

- Defined by the dual problem with 1-norm soft margin.

$$\left\{ \begin{array}{l} \text{Max: } W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j k(z_i, z_j), \\ \text{s.t.: } \sum_{i=1}^N y_i \alpha_i = 0, \\ \qquad \qquad 0 \leq \alpha_i \leq C, i = 1, \dots, N. \end{array} \right.$$

Classification for sample z ,

$$f(z) = \text{sgn} \left(\sum_{i=1}^N \alpha_i^* y_i k(z, z_i) + b^* \right)$$

Kernels Choice

- Linear: $K(x_i, x_j) = x_i \cdot x_j$
- Polynomial: $K(x_i, x_j) = [x_i \cdot x_j + 1]^d$
- Radial basis functions:

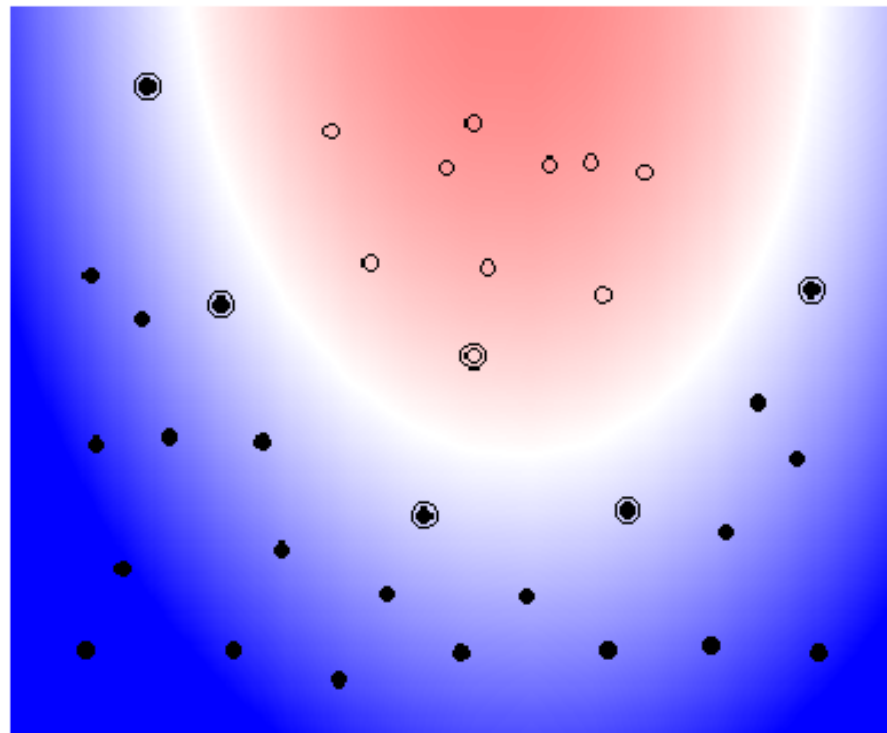
$$K(x_i, x_j) = \exp(-\gamma |x_i - x_j|^2)$$

- Sigmoid:

$$K(x_i, x_j) = \tanh(-\gamma |x_i - x_j| + c)$$

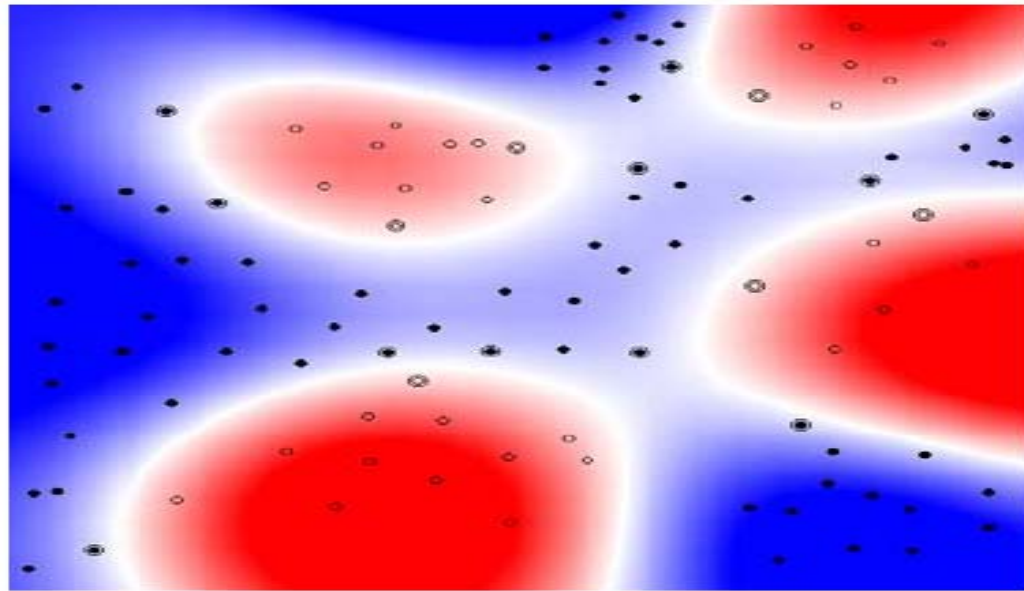
Example: Polynomial Kernel

$$K(x_i, x_j) = [x_i \cdot x_j + 1]^2$$



Example: Radial Basis Function (RBF) Kernel

$$K(x_i, x_j) = \exp(-\gamma|x_i - x_j|^2)$$



What is a valid kernel?

- Definition: Let X be nonempty set. A function $K(x_i, x_j)$ is a valid kernel in X if for all n and all $x_1, x_2, \dots, x_n \in X$ it produces a Gram matrix

$$G_{ij} = K(x_i, x_j)$$

That is symmetric and positive semi-definite.

How to Construct a Valid Kernel

- Theorem: Let K_1 and K_2 be valid kernels over $X \times X, X \subset R^N$, $\alpha > 0, 0 < \lambda < 1$, f a real-valued function on X , ϕ is a map from X to R^m with K_3 a kernel over $R^m \times R^m$, and G a symmetric positive semi-definite matrix, then the following functions are valid kernels

$$K(x, z) = \lambda K_1(x, z) + (1 - \lambda) K_2(x, z)$$

$$K(x, z) = \alpha K_1(x, z)$$

$$K(x, z) = K_1(x, z) K_2(x, z)$$

$$K(x, z) = f(x) f(z)$$

$$K(x, z) = K_3(\phi(x), \phi(z))$$

$$K(x, z) = x^T G z$$

Multiple Classes Problem

- To obtain a k -class classifier, we construct a set of binary classifiers f^1, \dots, f^k , each trained to separate one class from the rest, and combine them by doing the multiple-class classification according to the maximal output before applying the sign function.

Multiple Classes Problem

Taking,

$$\operatorname{Argmax}_{j=1,\dots,k} g^{(j)}(z)$$

$$g^{(j)}(z) = \sum_{i=1}^N y_i \alpha_i^{(j)} \cdot k(z, z_i) + b^{(j)}$$

These values can also be used to reject decisions, for instance by considering the difference between the maximum and the second highest value as a measure of confidence in the classification.

Why does SVM work well

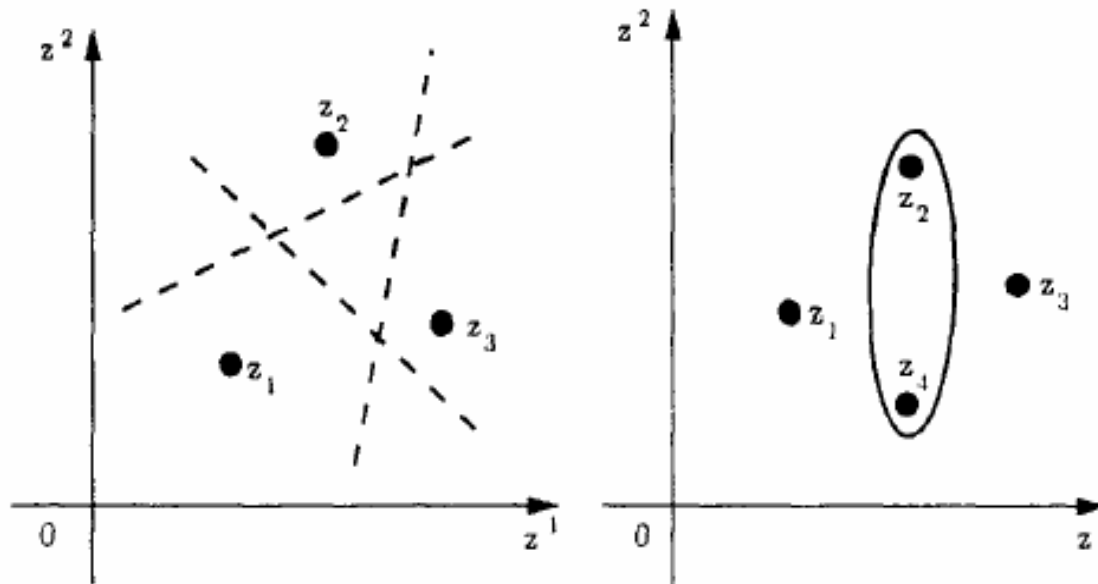
- Measurement: the risk of misclassifying of training samples and test samples (generalization problem).
- Principle: structural risk minimization.
- Key concept: VC dimension.

VC (Vapnik-Chervonenkis) Dimension

- **Definition:** the VC dimension for a set of function $f(\alpha)$ is defined as the maximum number of training points that can be shattered by $H=\{f(\alpha)\}$.
- **Shattered:** if a given set of N points can be labeled in all possible 2^N ways, and for each labeling, a number of the set $\{f(\alpha)\}$ can be found which correctly assigns those labels, we see that the set of points is shattered by that set of functions.

Example

VC dimension for lines in the plane is 3.



The VC dimension of a set of oriented hyperplane in \mathbb{R}^n is $n+1$

Example: Infinite VC Dimension

- $f(x, \alpha) = \text{sgn}(\sin \alpha x)$ can shatter arbitrary points.

$$x_i = 10^{-i}, i = 1, 2, \dots, l.$$

$$\alpha = \pi \left(1 + \sum_{i=1}^l \frac{(1 - y_i) 10^i}{2} \right)$$

- So the VC dimension of these function is infinite.

Expected Risk

- Given N observation $\{ (z_i, y_i), i=1,2,\dots,N \}$, and assume these observations are drawn from some unknown probability distribution $P(z,y)$. Suppose a classifier gives prediction $f(z,\alpha)$ for each z . The expectation of the test error for the classifier is called as **expected risk**,

$$R(\alpha) = \int \frac{1}{2} |y - f(z, \alpha)| dP(z, y)$$

Empirical Risk

- The empirical risk R_{emp} is

$$R_{emp}(\alpha) = \frac{1}{2N} \sum_{i=1}^N |y_i - f(z_i, \alpha)|$$

Relation Between Two Risks

- Under certain model, Vapnik proved the bound for the expected risk which holds with probability $1-\eta$.

$$R(\alpha) \leq R_{\text{emp}}(\alpha) + \sqrt{\frac{h(\log \frac{2N}{h} + 1) - \log \frac{\eta}{4}}{N}}$$

Where h is the VC dimension of $f(z, \alpha)$, and the second term on the right hand side is called *VC confidence*.

Other Classifiers Include...

- Decision tree (CART, C4.5)
- Neural networks
- Nearest neighbour (KNN)
- Logistic regression
- Projection pursuit
- Bayesian belief networks

Measuring the Accuracy of the Classifier

- Hold-out test
 - Hold a certain fraction of samples for test.
- Leave one out cross-validation
- K-fold cross-validation
 - 将数据集分为 k 个子集;
 - 用 $k-1$ 个子集作训练集, 1个子集作测试集, 然后 k 次交叉验证;

Measuring the Accuracy of the Classifier

	Real Negative	Real Positive
Claimed Positive	False Positive (FP)	True Positive (TP)
Claimed Negative	True Negative (TN)	False Negative (FN)

Measuring the Accuracy of the Classifier

- Sensitivity (S_n)
- False Positive Rate (FPR)
- Correlation Coefficient (CC)
- Approximate Correlation (AC)

$$S_n = \frac{TP}{TP + FN}$$

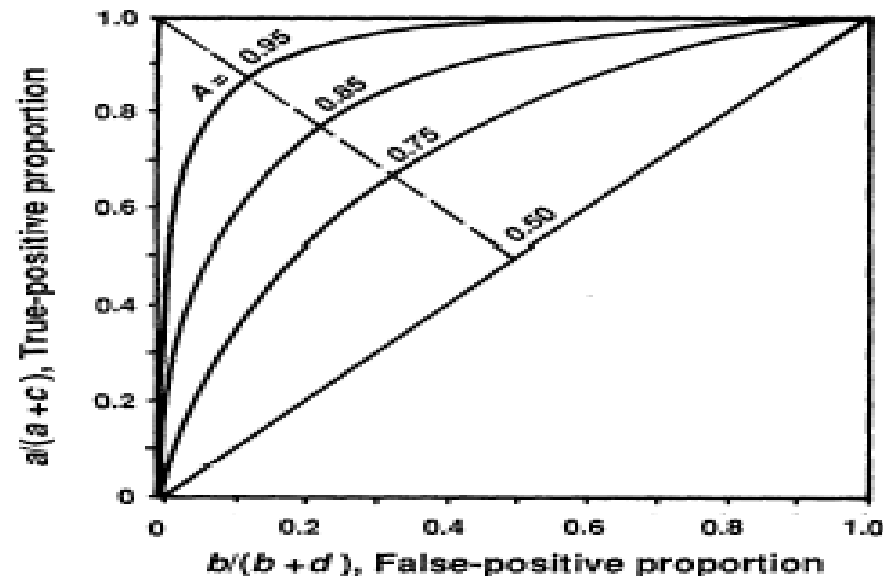
$$FPR = \frac{FP}{TN + FP}$$

$$CC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TN + FP)(TP + FP)(TN + FN)}}$$

$$AC = \frac{1}{2} \left(\frac{TP}{TP + FN} + \frac{TP}{TP + FP} + \frac{TN}{TN + FP} + \frac{TN}{TN + FN} \right) - 1$$

Measuring the Accuracy of the Classifier

- ROC: Receiver Operating Characteristic or Relative Operating Characteristic. True positive proportion v.s. False-positive proportion.

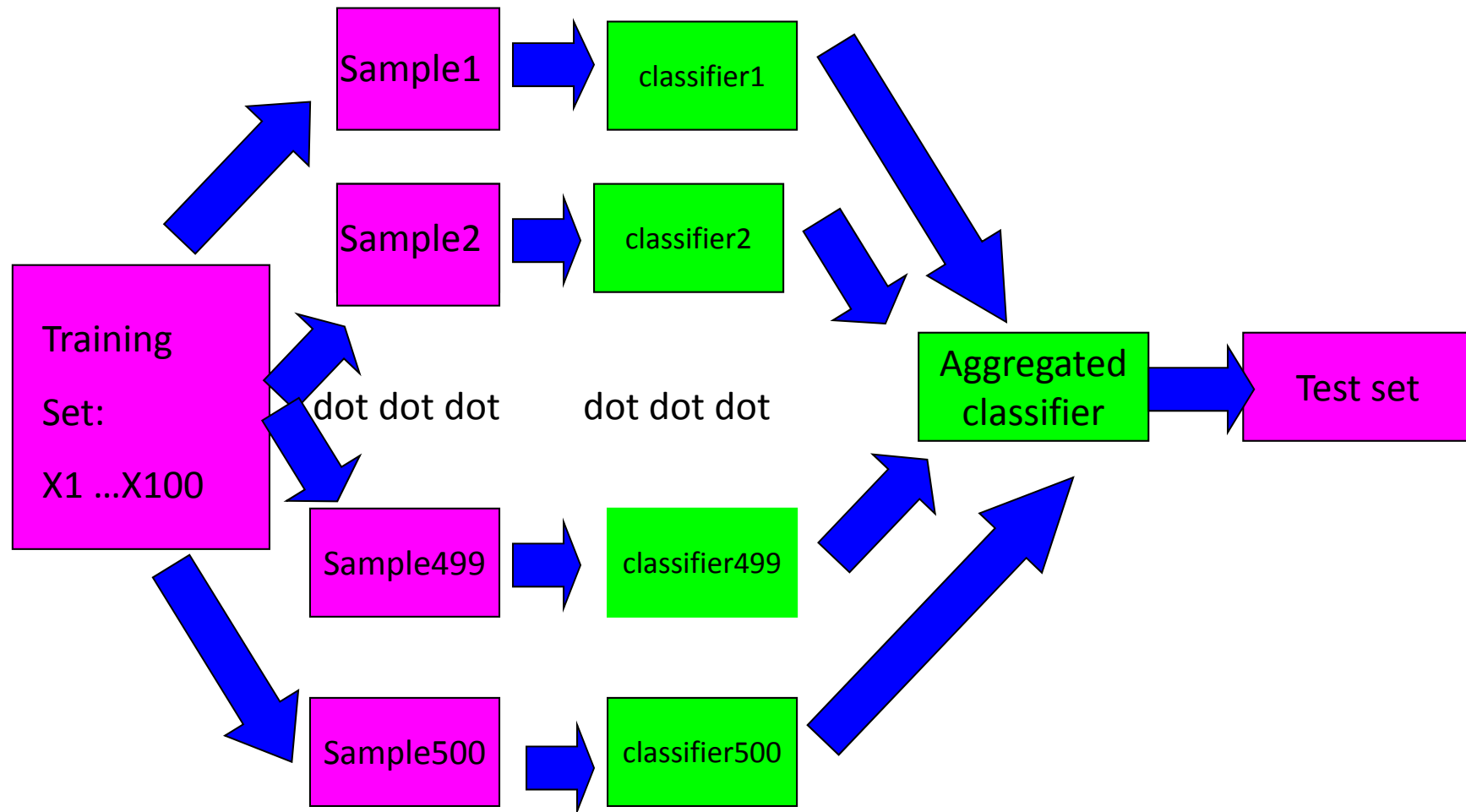


Aggregating classifiers

- Breiman (1996, 1998) found that gains in accuracy could be obtained by aggregating predictors built from perturbed versions of the learning set; the multiple versions of the predictor are aggregated by voting.
- Let $C(., L_b)$ denote the classifier built from the b th perturbed learning set L_b , and let w_b denote the weight given to predictions made by this classifier. The predicted class for an observation x is given by

$$\operatorname{argmax}_k \sum_b w_b I(C(x, L_b) = k)$$

Diagram of aggregating classifiers



Bagging

- Bagging = Bootstrap aggregating
- Non-parametric Bootstrap (standard bagging): perturbed learning sets drawn at random with replacement from the learning sets; predictors built for each perturbed dataset and aggregated by plurality voting ($w_b = 1$)
- Parametric Bootstrap: perturbed learning sets are multivariate Gaussian
- Convex pseudo-data (Breiman 1996)

Boosting

- Freund and Schapire (1997), Breiman (1998)
- Data **resampled adaptively** so that the weights in the resampling are increased for those cases most often misclassified
- Predictor aggregation done by **weighted voting**

Random Forests

- Perturbed learning sets are drawn at random with replacement from the learning sets
- The exploratory tree is built for each perturbed dataset in a way that at each node, the pre-specified number of features is randomly sub-sampled without replacement and only these variables are used to decide the split at that node.
- The resulting trees are aggregated by plurality voting ($w_b = 1$)