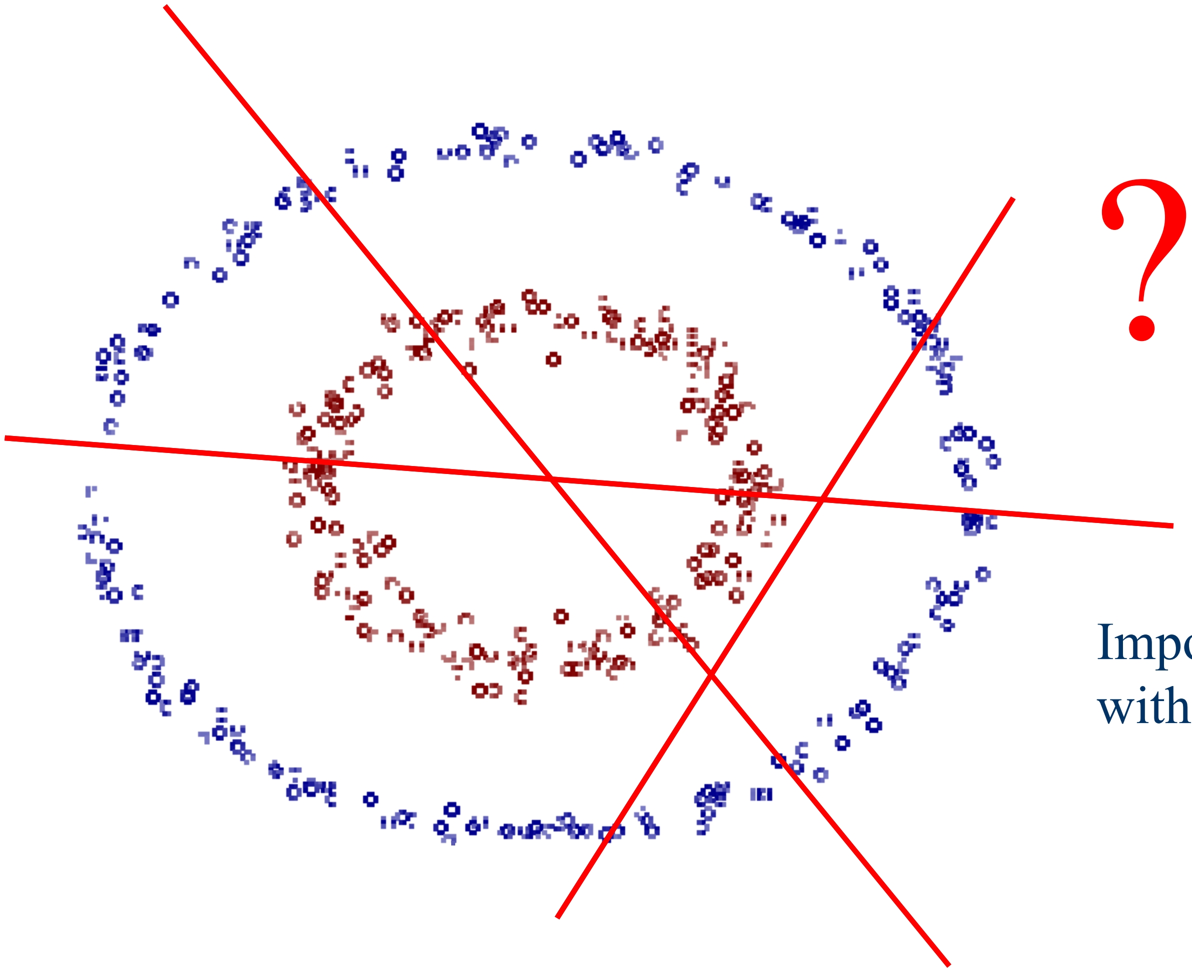
So Far…

- Our goal (supervised learning):
  - To learn a set of discriminant functions
- Bayesian framework
  - We could design an optimal classifier if we knew:
    - $P(\omega_i)$: priors and $P(x \mid \omega_i)$: class-conditional densities
    - Using training data to estimate $P(\omega_i)$ and $P(x \mid \omega_i)$
- Directly learning discriminant functions from the training data
  - We only know the form of the discriminant functions
  - Linear Regression
  - Logistic Regression
  - SVM

Linear

Nonlinear Distributed Data



Impossible to separate
with a hyperplane

Generalized Linear Function & Kernel Methods

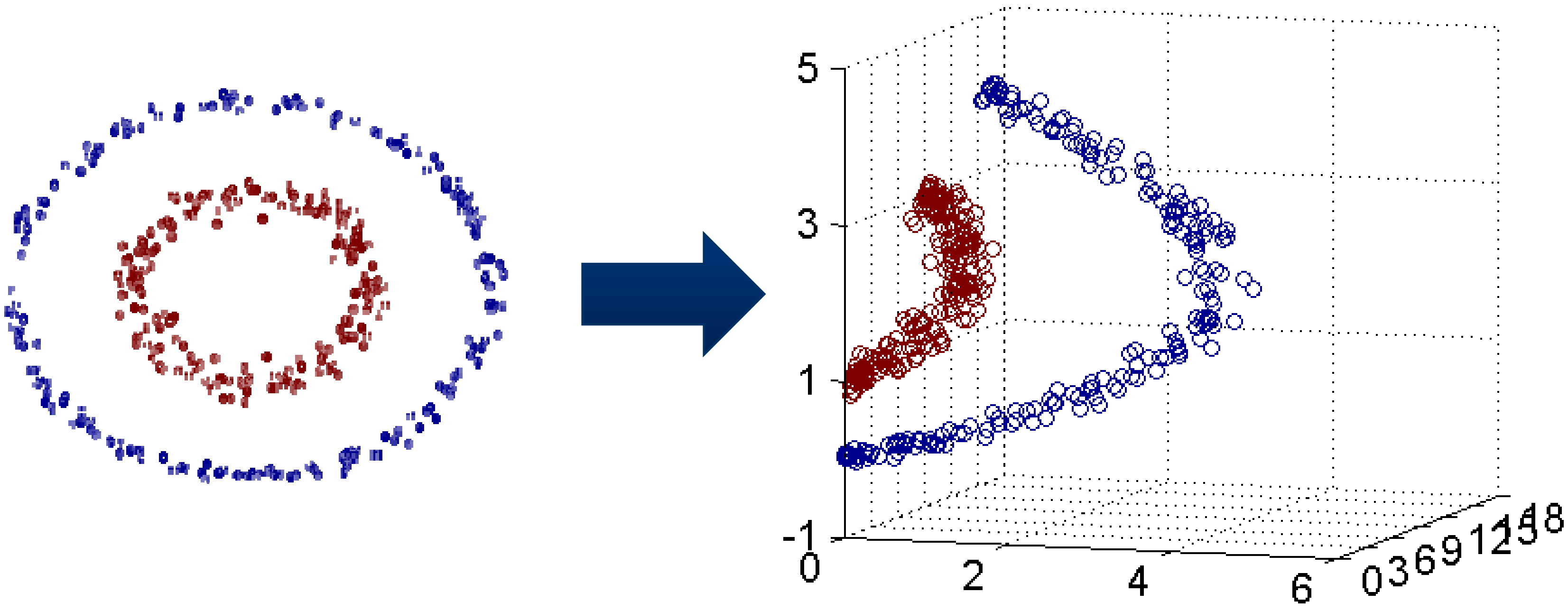
Deng Cai (蔡登)

College of Computer Science
Zhejiang University

dengcai@gmail.com

A Circle from 2D to 3D

- Here is an example of mapping a (special case) circle in 2D to 3D (the result is linear separable):



Generalized Linear Discriminant Functions

- Recall the Linear Discriminant Function

$$g(\boldsymbol{x}) = \sum_{i=1}^{d} w_i x_i + w_0$$

  - $g(\boldsymbol{x})$ positive implies class 1
  - $g(\boldsymbol{x})$ negative implies class 2

- Generalized Linear Discriminant
  - Add additional terms involving the products of features
  - For example,
    - Given: $[x_1,\ x_2,\ x_3]$
    - Make it: $[\ x_1,\ x_2,\ x_3,\ x_1x_2,\ x_2x_3,\ x_1x_2x_3\ ]$ by adding products of features.
  - Learn a discriminant function that is linear in the new feature space

Quadratic Discriminant Function

- Quadratic Discriminant Function
  - Obtained by adding pair-wise products of features

$$g(\mathbf{x}) = \underbrace{w_0 + \sum_{i=1}^{d} w_i x_i}_{\substack{\text{Linear Part} \\ \text{(d+1) parameters}}} + \underbrace{\sum_{i=1}^{d} \sum_{j=1}^{d} w_{ij} x_i x_j}_{\substack{\text{Quadratic part, d(d+1)/2} \\ \text{additional parameters}}}$$

    - $g(x)$ positive implies class 1; $g(x)$ negative implies class 2

- $g(x) = 0$, represents a hyperquadric (hyperparaboloid, hyperellipsoid, hyperhyperboloids), as opposed to hyperplanes in linear discriminant case.
- Adding more terms such as $w_{ijk}x_i x_j x_k$ results in polynomial discriminant functions.

Quadratic Discriminant Function

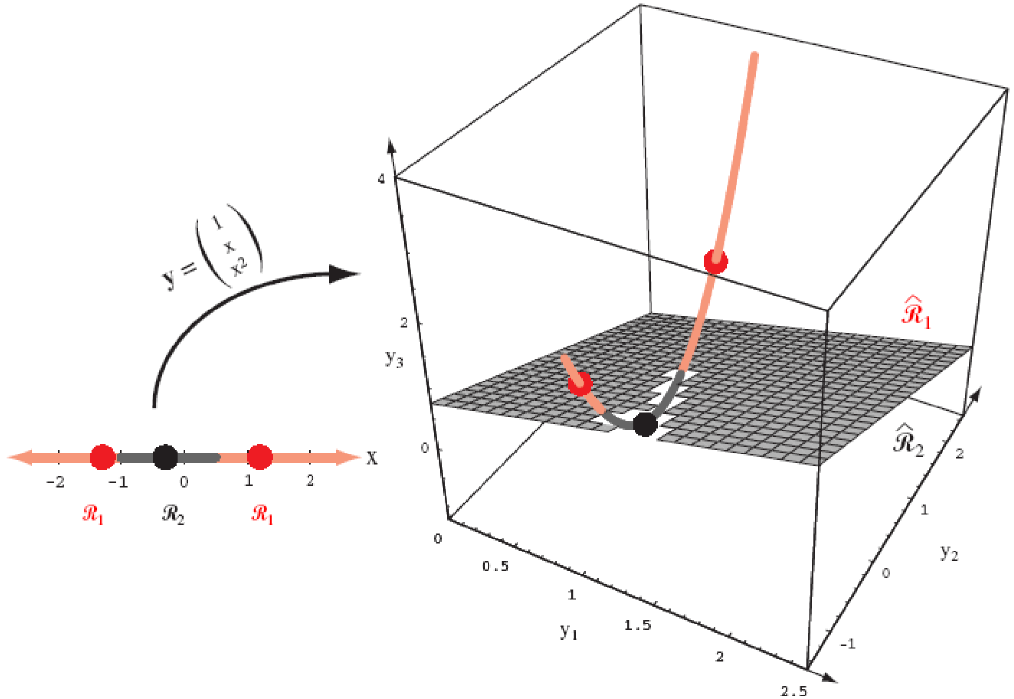


Figure 5.5: The mapping $\mathbf{y} = (1, x, x^2)^t$ takes a line and transforms it to a parabola in three dimensions. A plane splits the resulting $\mathbf{y}$ space into regions corresponding to two categories, and this in turn gives a non-simply connected decision region in the one-dimensional x space.

Quadratic Discriminant Functions

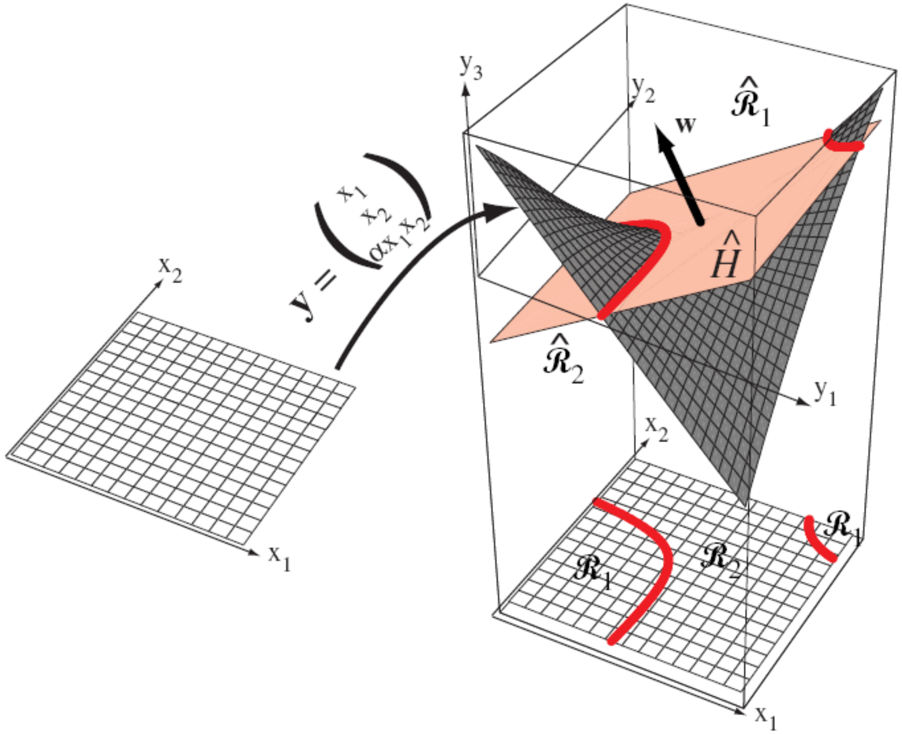


Figure 5.6: The two-dimensional input space $\mathbf{x}$ is mapped through a polynomial function f to $\mathbf{y}$. Here the mapping is $y_1 = x_1$, $y_2 = x_2$ and $y_3 \propto x_1x_2$. A linear discriminant in this transformed space is a hyperplane, which cuts the surface. Points to the positive side of the hyperplane $\hat{H}$ correspond to category ω_1, and those beneath it ω_2. Here, in terms of the $\mathbf{x}$ space, $\mathcal{R}_1$ is a not simply connected.

Generalized Discriminant Function

- A *generalized linear discriminant* function can be written as,

Dimensionality of the augmented feature space.

$$g(\mathbf{x}) = \sum_{i=1}^{\hat{d}} a_i y_i(\mathbf{x})$$

Setting $y_i(x)$ to be monomials results in polynomial discriminant functions

Weights in the augmented feature space. Note that the function is linear in *a*.

- Equivalently,

$$g(\mathbf{x}) = \mathbf{a}^t \mathbf{y}$$

$$\mathrm{a} = [a_1, a_2, \ldots, a_{\hat{d}}]^t \qquad \mathbf{y} = \underbrace{[y_1(x), y_2(x), \ldots, y_{\hat{d}}(x)]^t}$$

also called the augmented feature vector.

Phi Function

- The discriminant function $g(x)$ is not linear in x, but is linear in y.
- The mapping $y = [y_1(x), y_2(x), \ldots, y_{\hat{d}}(x)]^t$ is taking a d-dimensional vector x and mapping it to a $\hat{d}-$dimensional space. The mapping y is called the phi-function.
- When the input patterns x are non-linearly separable in the input space, mapping them using the *right* phi-function maps them to a space where the patterns are linearly separable.
- Unfortunately, the curse of dimensionality makes it hard to capitalize this in practice. A complete QDF involves (d+1) (d+2)/2 terms; for modest values of d, say d =50, this requires many terms

Representer Theorem

Theorem 4.2 (Representer Theorem) *Denote by* $\Omega : [0, \infty) \to \mathbb{R}$ *a strictly monotonic increasing function, by* $\mathcal{X}$ *a set, and by* $c : (\mathcal{X} \times \mathbb{R}^2)^m \to \mathbb{R} \cup \{\infty\}$ *an arbitrary loss function. Then each minimizer* $f \in \mathcal{H}$ *of the regularized risk*

$$c\left(\left(x_1, y_1, f(x_1)\right), \ldots, \left(x_m, y_m, f(x_m)\right)\right) + \Omega\left(\|f\|_{\mathcal{H}}\right) \tag{4.4}$$

admits a representation of the form

$$f(x) = \sum_{i=1}^{m} \alpha_i k(x_i, x). \tag{4.5}$$

Kernelized Ridge Regression

$$\boldsymbol{w}^* = \operatorname{argmin} \sum_{i=1}^{n} \left(y_i - \boldsymbol{x}_i^T \boldsymbol{w}\right)^2 + \lambda \sum_{j=1}^{p} w_j^2$$

$$\boldsymbol{w}^* = \left(XX^T + \lambda \boldsymbol{I}\right)^{-1} X\boldsymbol{y}$$

- **Woodbury matrix identity**

$$(\mathbf{P}^{-1} + \mathbf{B}^T\mathbf{R}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{R}^{-1} = \mathbf{P}\mathbf{B}^T(\mathbf{B}\mathbf{P}\mathbf{B}^T + \mathbf{R})^{-1} \qquad (158)$$

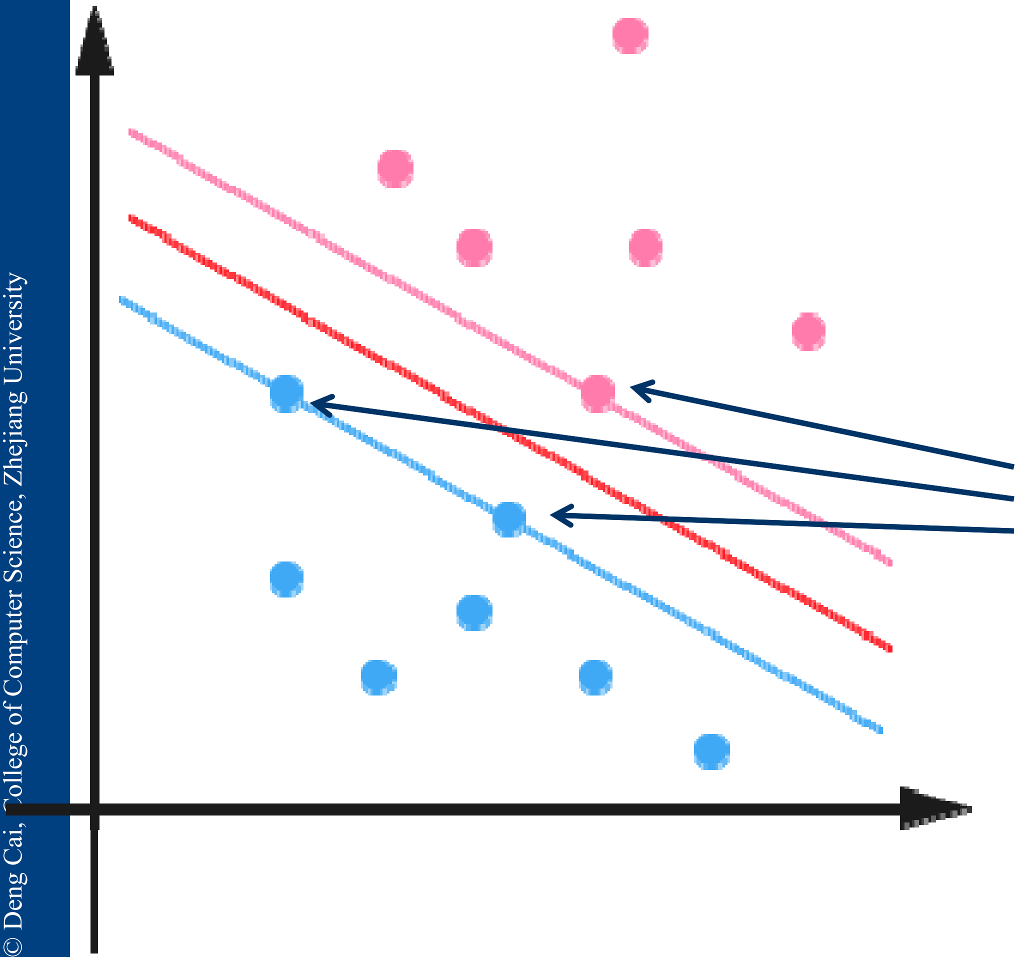
$$\boldsymbol{w}^* = X\left(X^T X + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{y}$$

$$\boldsymbol{\alpha} \triangleq \left(X^T X + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{y}$$

$$\boldsymbol{w}^* = X\boldsymbol{\alpha} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i$$

$$\hat{f}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i^T \boldsymbol{x} = \sum_{i=1}^{n} \alpha_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}) \qquad \kappa(\boldsymbol{x}_i, \boldsymbol{x}) = \boldsymbol{x}_i^T \boldsymbol{x}$$

Support Vector Machine



Hyper plane of maximum margin is *supported* by those points (vectors) on the margin. Those are called Support Vectors. Non-support vectors can move freely without affecting the position of the hyperplane as long as they don't exceed the margin.

Support Vector Machine

- The final classifier is

$$\operatorname{sgn}\left(\boldsymbol{w}^T\boldsymbol{x} + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + b\right)$$

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i$$

- Note: for non-support vectors, the corresponding α_i is zero.

Kernels

- Let $\kappa(\boldsymbol{x}, \boldsymbol{x}') \geq 0$ be some measure of similarity between objects $\boldsymbol{x}, \boldsymbol{x}' \in \chi$, where χ is some abstract space; we will call κ a kernel function.
  - Typically the function is symmetric, and non-negative

- Examples
  - Linear kernels $\kappa(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{x}'$
  - Polynomial kernels $\kappa(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T \boldsymbol{x}' + 1)^d$
  - RBF kernels $\kappa(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2}\right)$

The advantages of kernel methods

- Non-linear classifiers
  - The kernel → Nonlinearity of the learned function.

- The samples can not be represented as feature vectors
  - But we can get the similarity of two samples

  - String kernels
  - Graph kernels