Pattern Classification (III)

杜俊

jundu@ustc.edu.cn

Outline

- Bayesian Decision Theory
  - How to make the optimal decision?
  - Maximum *a posterior* (MAP) decision rule

- Generative Models
  - Joint distribution of observation and label sequences
  - Model estimation: MLE, Bayesian learning, discriminative training

- Discriminative Models
  - Model the posterior probability directly (discriminant function)
  - Logistic regression, support vector machine, neural network

Plug-in MAP Decision Rule

$$\mathrm{C}_P = \arg\max_{\mathrm{C}_i} p(\mathrm{C}_i \mid X) = \arg\max_{\mathrm{C}_i} P(\mathrm{C}_i) \cdot p(X \mid \mathrm{C}_i)$$

$$\approx \arg\max_{\mathrm{C}_i} \overline{P}_{\Gamma_i}(\mathrm{C}_i) \cdot \overline{p}_{\Lambda_i}(X \mid \mathrm{C}_i)$$

Generative VS. Discriminative Models

- Generative Models
  - Joint distribution of observation and label sequences

$$p(X, \mathrm{C}_i) = P(\mathrm{C}_i) \cdot p(X \mid \mathrm{C}_i)$$

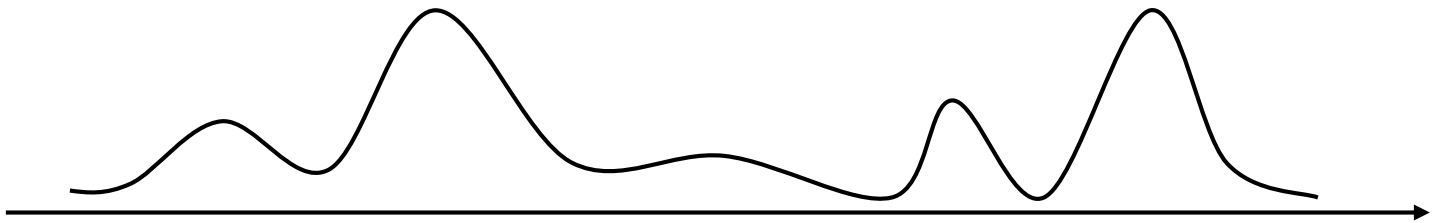
- Discriminative Models
  - Model the posterior probability directly (discriminant function)
  - Model the boundaries of data sets

$$\mathrm{g}_i(X) \Rightarrow p(\mathrm{C}_i \mid X)$$

Useful Models (I)

- Choose a proper model based on the nature of observation data

- Some useful statistical models for a variety of data types:
  - Gaussian (Normal) distribution
    ➔ Uni-modal continuous feature scalars
  - Multivariate Gaussian (Normal) distribution
    ➔ Uni-modal continuous feature vectors
  - Gaussian Mixture models (GMM)
    ➔ Multi-modal continuous feature scalars/vectors



Useful Models (II)

- Markov chain model: discrete sequential data
  - N-gram model in language modeling

- Hidden Markov model (HMM): ideal for various kinds of sequential observation data; provides better modeling capability than simple Markov chain model :
  - Model speech signals for recognition (speech recognition)
  - Model sign/gesture for recognition (sign language recognition)
  - Model biological data (DNA & protein sequence): profile HMM
  - …

Useful Models (III)

- Markov random field: multi-dimensional spatial data
  - Model image data: e.g., used for OCR, etc
  - HMM is a special case of Markov random field

- Graphical models (a.k.a., Bayesian networks, Belief networks)
  - Widely used in machine learning, data mining
  - High-dimensional data (discrete or continuous)
  - Model a very complex stochastic process
  - Automatically learn dependency from data
  - HMM is also a special case of graphical model

- Discriminant functions
  - Linear regression
  - Logistic regression
  - Support vector machine
  - Neural network

Discriminant Functions (I)

- Instead of designing a classifier based on probability distribution , we can build an ad-hoc classifier based on some discriminant functions to model class boundary info directly.
  - A set of discriminant functions $g_i(\mathbf{x}; \theta_i)$ for each class C_i
  - $g_i(\mathbf{x}; \theta_i)$ has a pre-defined function form with unknown parameters θ_i which should be estimated from training data
  - For an unknown pattern with feature vector $\mathbf{x}$, the decision is

$$C_{\mathbf{x}} = \arg\max_i g_i(\mathbf{x}; \theta_i)$$

Discriminant Functions (II)

- Examples
  - Linear discriminant function:

$$g(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = \langle \mathbf{w}, \mathbf{x} \rangle + w_0$$

  - Quadratic discrimiant function: (2nd order)

$$g(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{W}\mathbf{x} + \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = \langle \mathbf{x} \otimes \mathbf{x}, \mathrm{vec}(\mathbf{W}) \rangle + \langle \mathbf{w}, \mathbf{x} \rangle + w_0$$

  - Polynomial discriminant function: (N-th order)

  - Neural network: (arbitrary nonlinear function)

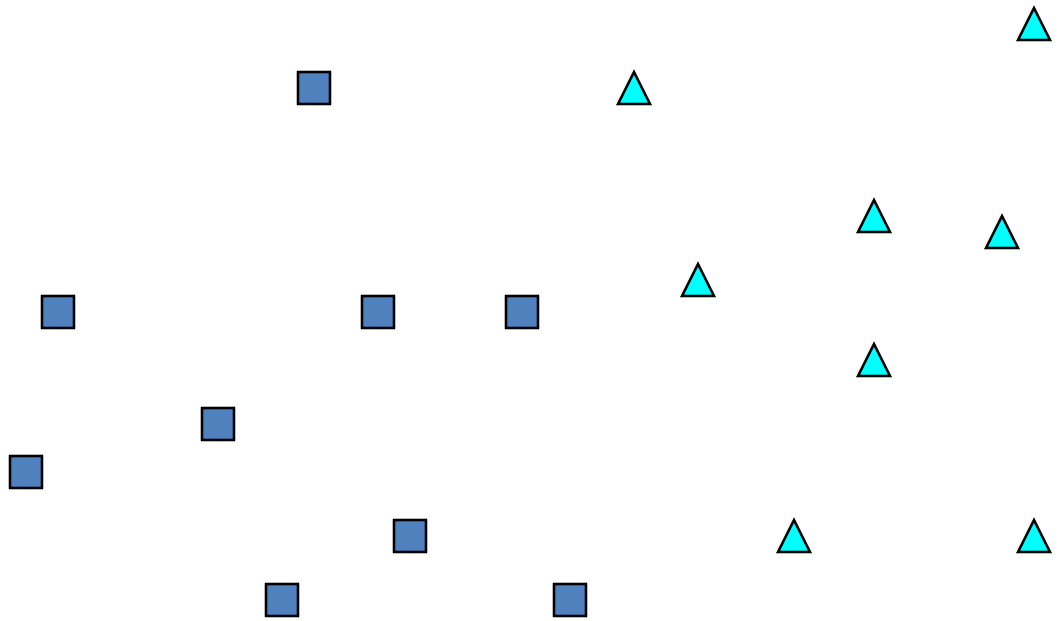
  - Optimal MAP classifier is a special case when choosing discriminant functions as the class posterior probabilities

Discriminant Functions (III)

- Unknown parameters of discriminant functions are estimated based on optimizing an objective function via some gradient descent methods:
  - Linear regression: Achieving a good mapping
  - Logistic regression: Minimizing empirical classification errors
  - Support vector machine (SVM): Maximizing separation margin
  - Neural network: MMSE or cross-entropy minimization

Linear Regression

- Find a good mapping from **x** to y



Label: +1

Label: -1

Linear Regression

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \mathbf{x}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_N^{\mathrm{T}} \end{bmatrix} \qquad \mathbf{y} = \mathbf{X}\mathbf{w} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} +1 \\ -1 \\ \vdots \\ +1 \end{bmatrix}$$

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \left\| \mathbf{X}\mathbf{w} - \mathbf{y} \right\|_2^2$$

$$\mathbf{w}^* = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

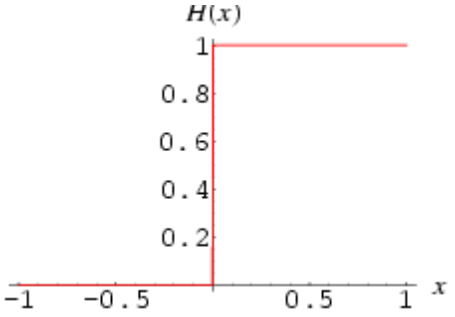
Linear regression does NOT work well for classification

Logistic Regression

- Counting errors in training samples

$$(\mathbf{x}_i, y_i) \Rightarrow \begin{cases} g_i = -y_i \mathbf{w}^{\mathrm{T}} \mathbf{x}_i < 0 & \text{Correct Classification} \\ g_i = -y_i \mathbf{w}^{\mathrm{T}} \mathbf{x}_i > 0 & \text{Wrong Classification} \end{cases}$$

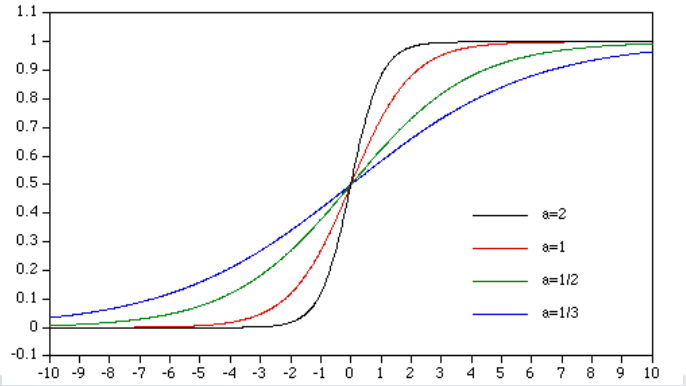
$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_i H(g_i) = \arg\min_{\mathbf{w}} \sum_i H(-y_i \mathbf{w}^{\mathrm{T}} \mathbf{x}_i)$$



$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_i l(g_i) = \arg\min_{\mathbf{w}} \sum_i l(-y_i \mathbf{w}^{\mathrm{T}} \mathbf{x}_i)$$

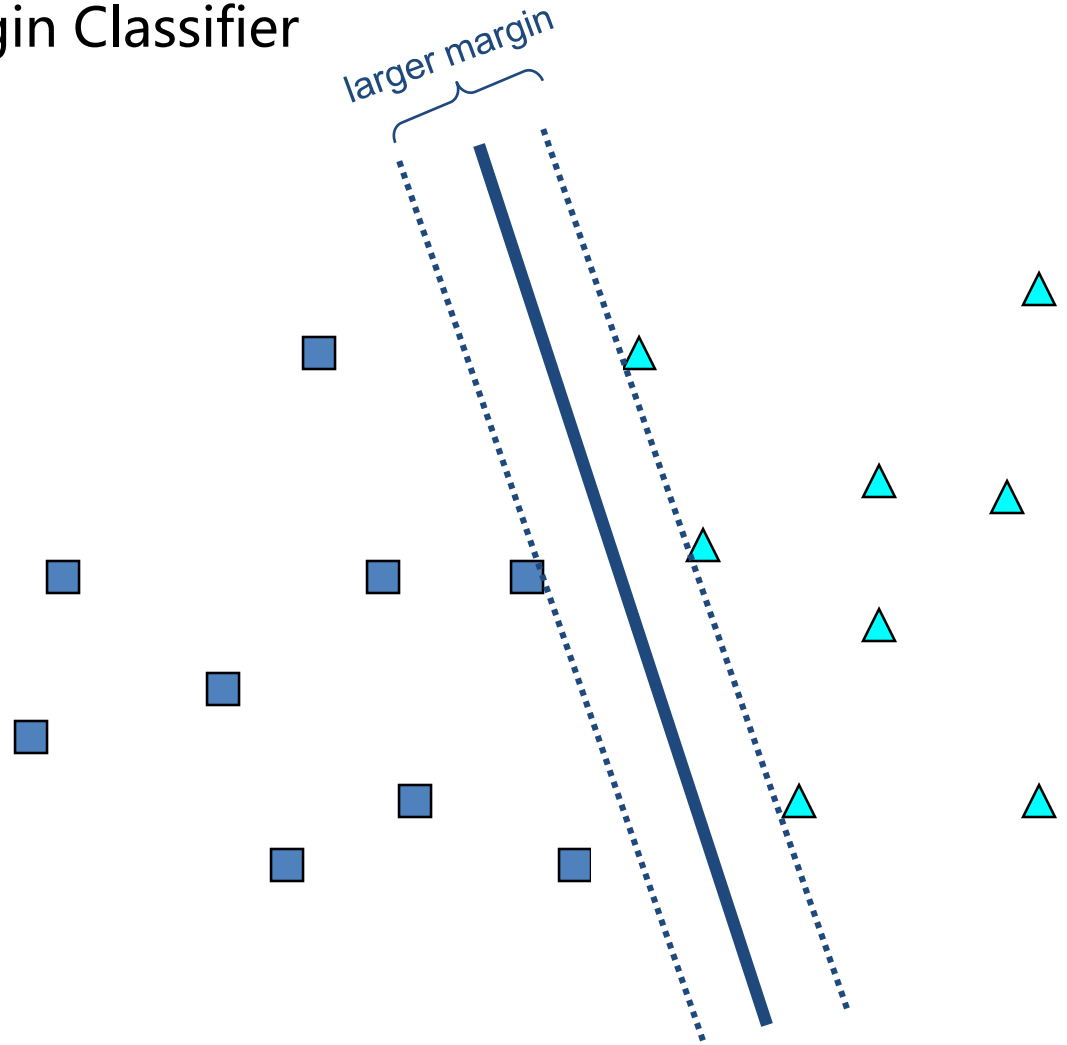
$$l(x) = \frac{1}{1 + e^{-sx}}$$

logistic sigmoid function



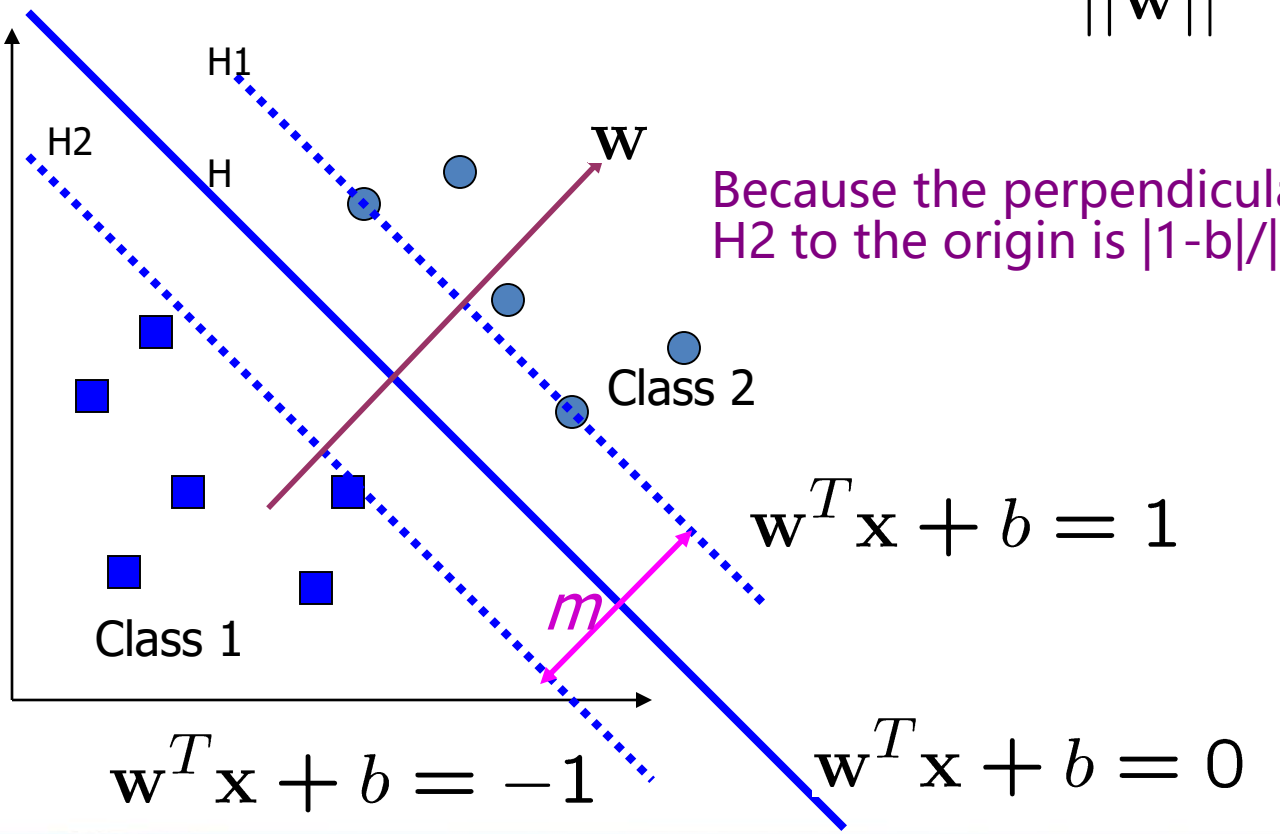
Support Vector Machine (I)

- Maximum Margin Classifier



Support Vector Machine (II)

- The decision boundary H should be as far away from the data of both classes as possible
- We should maximize the margin: $m = \frac{2}{||\mathbf{w}||}$



Because the perpendicular distance from H1, H2 to the origin is |1-b|/||w|| and |-1-b|/||w||

Support Vector Machine (III)

- The decision boundary can be found by solving the following constrained optimization problem:

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2 \quad \|\mathbf{w}\|^2 = \mathbf{w}^{\mathrm{T}}\mathbf{w} \quad \text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

- Convert to its dual problem (using KKT condition):

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad \text{subject to } \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

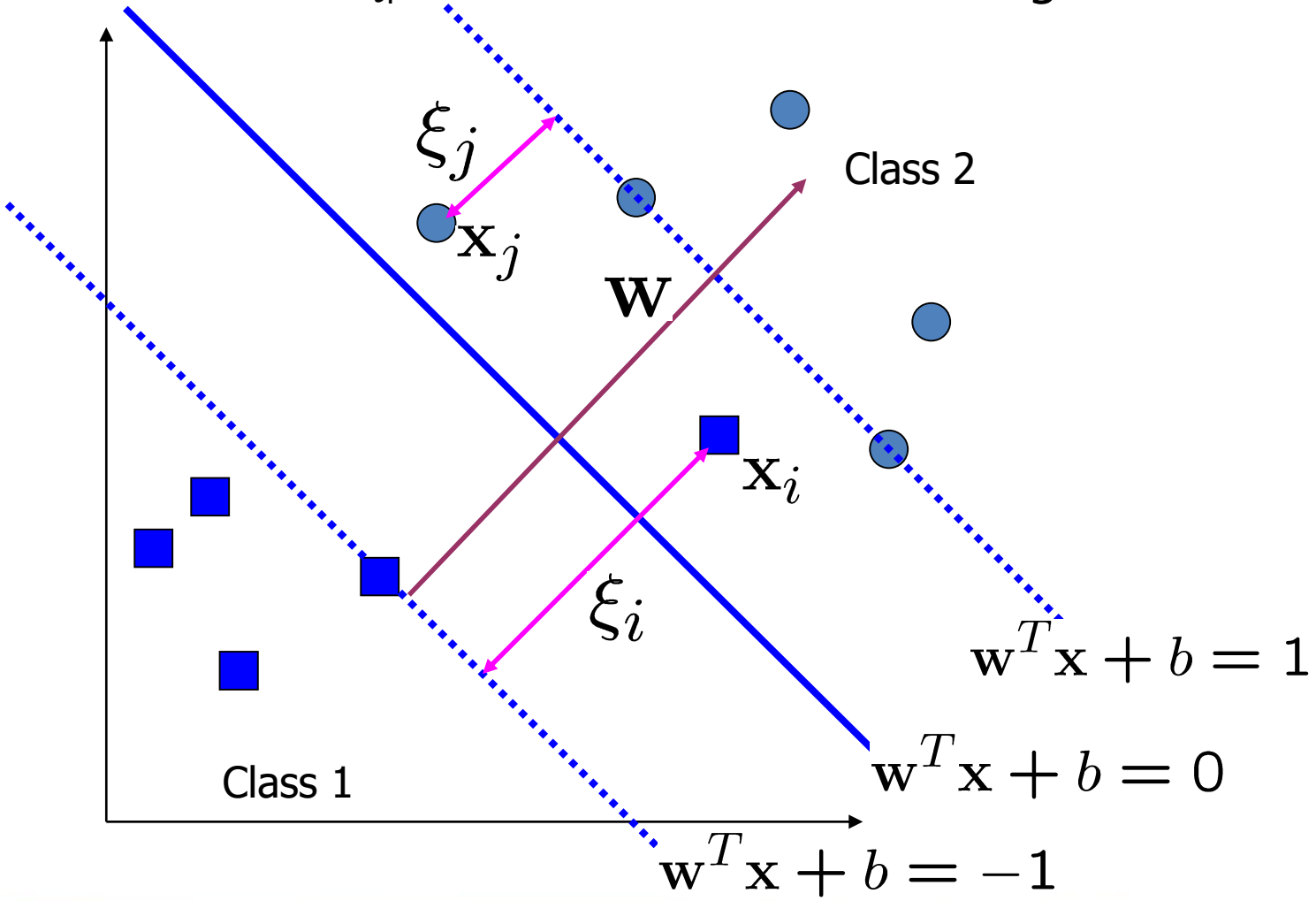
- A standard quadratic programming (QP) problem

$$\mathbf{w}^* = \sum_i \alpha_i y_i \mathbf{x}_i$$

- Support vectors: $\alpha_i \neq 0$

Support Vector Machine (IV)

- We allow "error" ξ_i in classification → soft-margin SVM



Support Vector Machine (V)

- Soft-margin SVM can be formulated as:

$$w^* = \min_{w,x_i} \left[\frac{1}{2} \| w \|^2 + C \cdot \sum_i x_i \right]$$

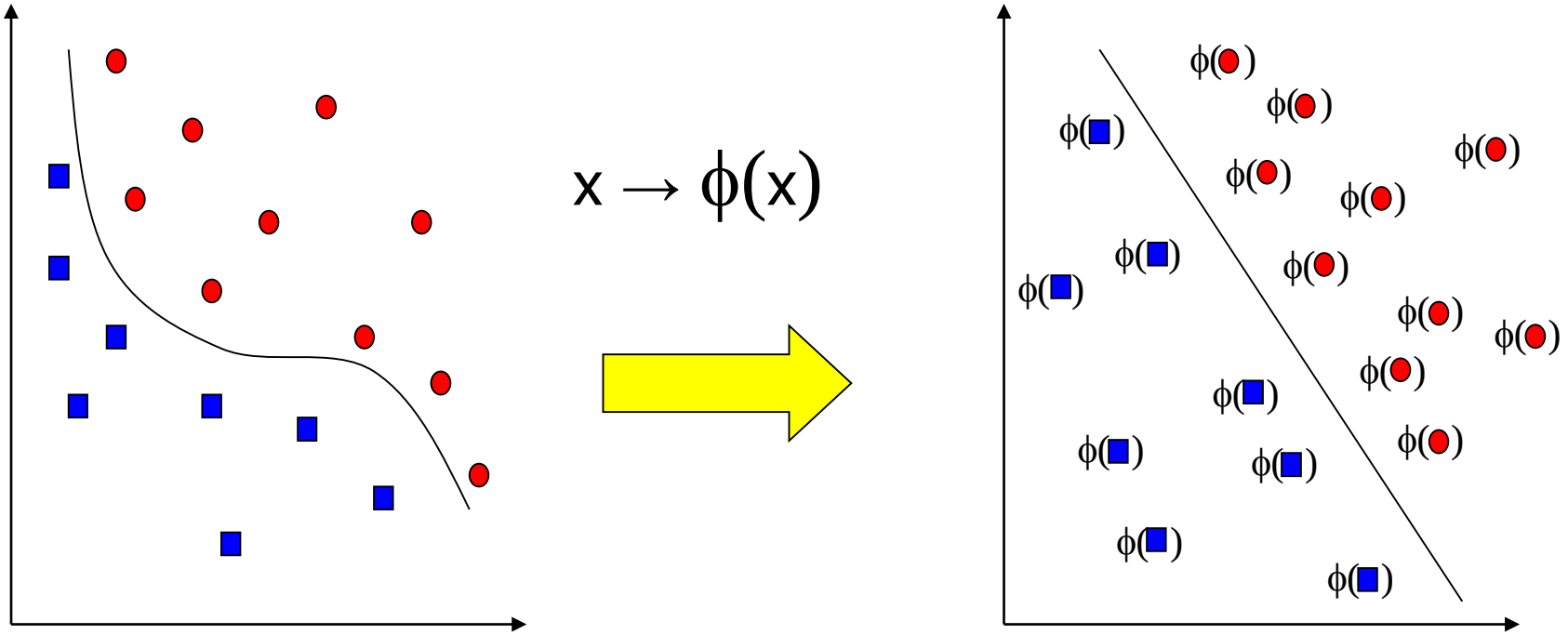
subject to

$$y_i(x_i w^T + b) > 1 - x_i \quad x_i > 0 \quad (" i)$$

- C is the regularization coefficient
- $\sum_i \xi_i$ is the upper bound of training classification errors

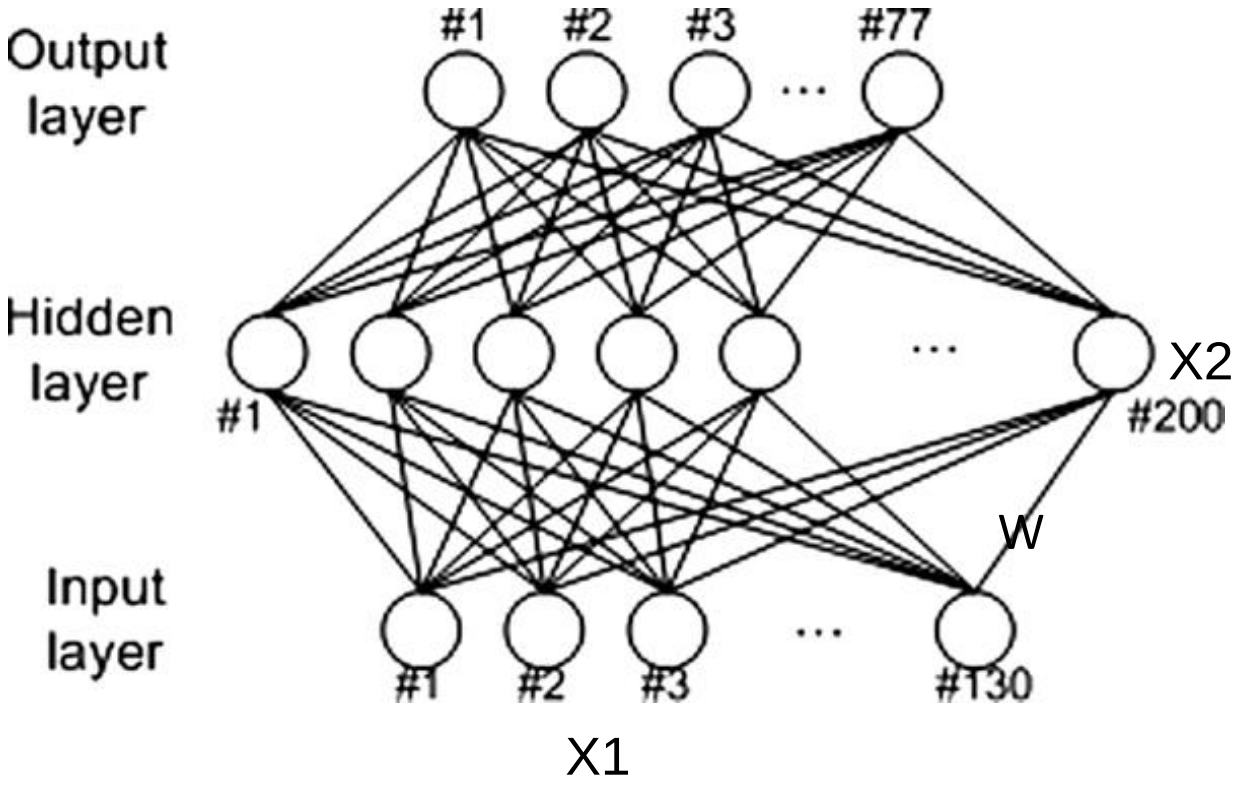
Support Vector Machine (VI)

- For nonlinear separation boundary
  - use a Kernel function



Neural Network

- Feed-forward multilayer perceptron (MLP)
- Error back-propagation (BP)



$$X_2 = l(X_1 W)$$

$$l(x) = \frac{1}{1 + e^{-sx}}$$