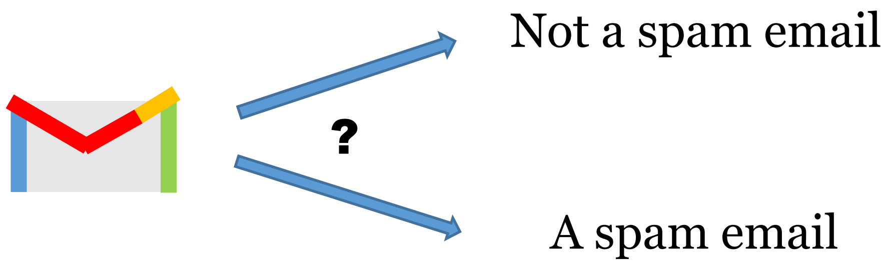
AI Lab for Wireless Communications

Week3 – Decoding with support vector machine

Speaker: Kuan-Yu Lin

Classification Problem

- Purpose: Labeling data to class
- Toy example: spam emails
- A popular machine learning algorithm: support vector machine



Decoding (Classification)

- Decoding can be think as a classification problem.
- Mapping the estimated message to transmitted message
- $\hat{m}$ compare with message m

$\underline{d}$=(0010011) **?**

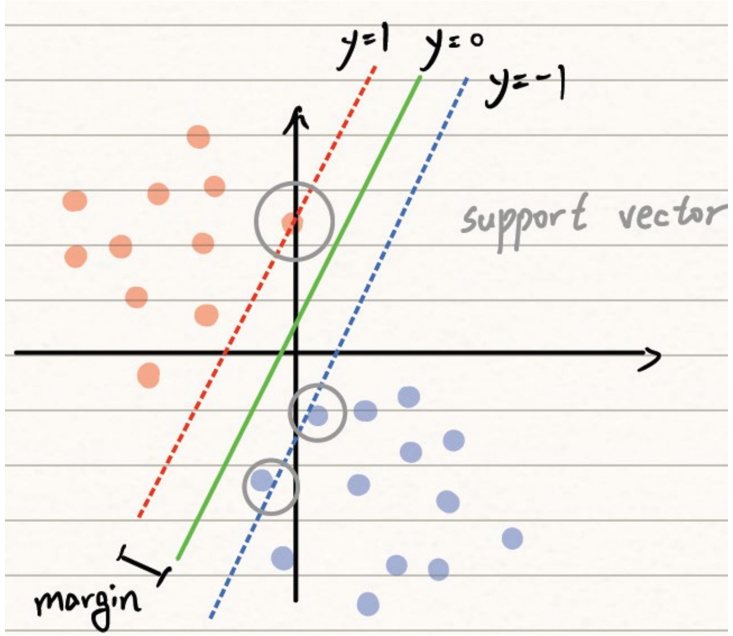
m_0=(0000000) class 0

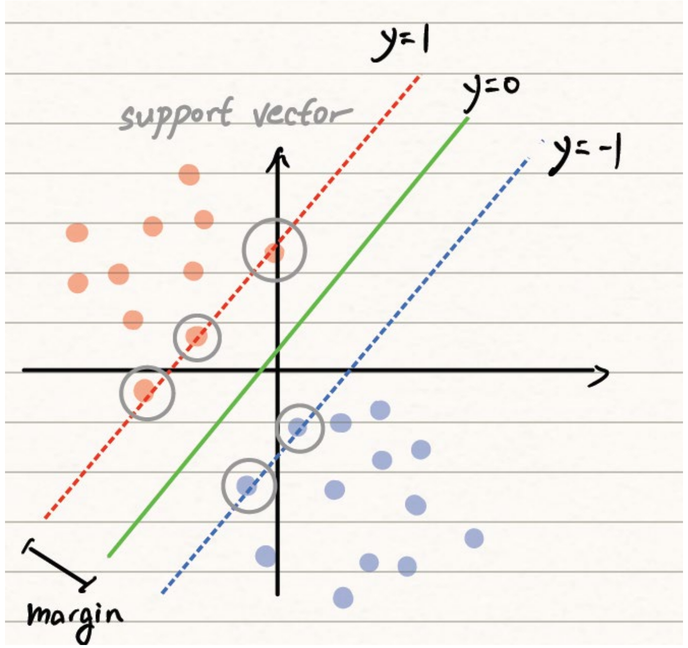
⋮ ⋮

m_{15} = (1111111) class 15

Support Vector Machine

- A binary classification method
- Maximum margin to get the best classification
- $\begin{cases} y=1,\ y=-1 \implies \text{margin boundary} \\ y=0 \qquad\quad\ \ \implies \text{decision boundary} \end{cases}$





Support Vector Machine – Training Part

- Step 1. Choose the data and label to train decision boundary
- Step 2. Set our model and let the kernel be linear as example
- Step 3. Train the model
- Step 4. Get w, w_0, support vector

- clf = SVC(), clf.fit(), clf.dual_coef_, clf.intercept_, clf.support_vectors_

Support Vector Machine – Predict Part

- Step1. Input a testing data to the discriminant function g(.)

- Step2.
  - If $g(x) \geq 0$, then x is assign to class 0
  - If $g(x) < 0$, then x is assign to class 1

How to achieve optimal?

- Information of SVM classification
  - Decision boundary: $y = w^T x + w_0$
  - Margin boundary: $w^T x + w_0 = 1 \;\; or \;\; -1$
  - Margin = $\frac{1}{\|w\|}$
- Finding the optimal (w, w_0) to maximize the margin under the following condition

$$y_n(w^T x + w_0) \geq 1 \;\text{ for } n = 1, 2, \dots, N$$

where $y_n \in \{-1,1\}$ is the class label for x_n

How to achieve optimal?

- Formulate the problem as follow

$$\text{minimize } J(w, w_0) = \frac{1}{2}||w||^2$$

$$\text{subject to } y_n(w^T x + w_0) \geq 1 \text{ for } n = 1, 2, \dots, N$$

It is a constrained optimization problem!

Constrained optimization problem

- The Lagrangian function is:

$$J(w, w_0, \alpha) = \frac{1}{2} w^T w - \sum_{n=1}^{N} \alpha_n [y_n(w^T x_n + w_0) - 1]$$

where α_n is the Lagrangian coefficient

- Condition of optimality

$$\frac{\partial J(w,w_0,\alpha)}{\partial w} = 0 \quad \Longrightarrow \quad w = \sum_{n=1}^{N} \alpha_n y_n x_n$$

$$\frac{\partial J(w,w_0,\alpha)}{\partial w_0} = 0 \quad \Longrightarrow \quad \sum_{n=1}^{N} \alpha_n y_n = 0$$

Constrained optimization problem

- Lagrangian dual problem:

Maximize $J(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N} \alpha_m \alpha_n y_m y_n x_m^T x_n$

Subject to $\sum_{n=1}^{N} y_n \alpha_n = 0$

$\alpha_n \geq 0$ for n=1,2,…,N

- Judge support vector by KKT condition

(complementary slackness)

if $\hat{\alpha}_n = 0, \quad y_n(w^T x + w_0) > 1$

if $\hat{\alpha}_n \neq 0, \quad y_n(w^T x + w_0) = 0 \quad \rightarrow \quad$ *support vectors*

Constrained optimization problem

- For optimal Lagrangian multipliers, optimal weight vector is given by

$$\hat{w} = \sum_{n=1}^{N_s} \hat{\alpha}_n \, y_n x_n$$

where N_s is the number of support vectors

http://scikit-learn.org.cn/view/83.html#1.4.7%20%E6%95%B0%E5%AD%A6%E5%85%AC%E5%BC%8F

Discriminant function

- For testing sample x, the discriminant function g(.) is

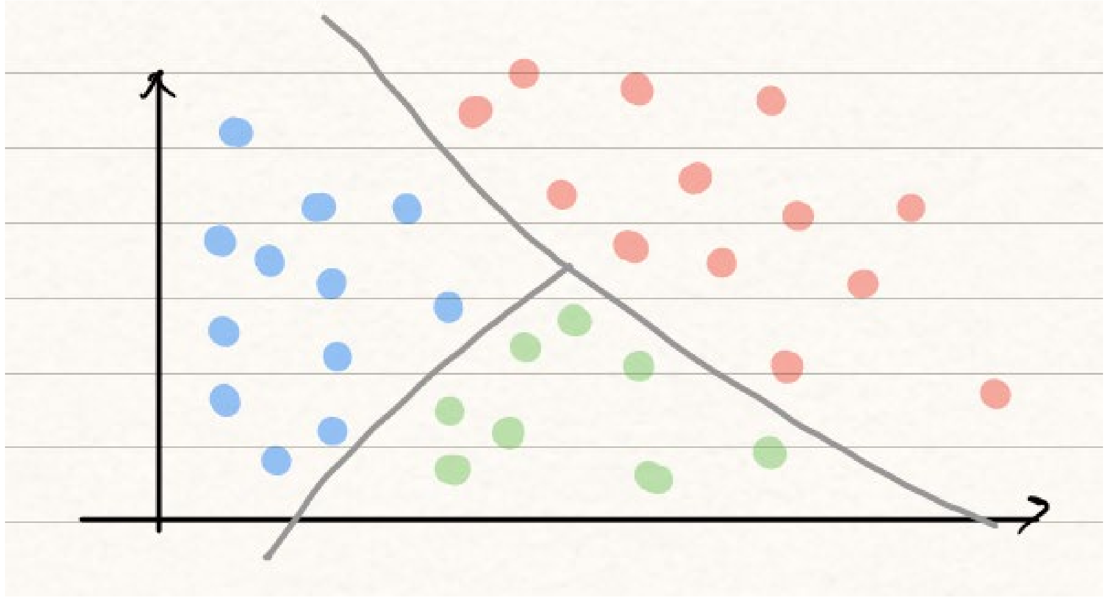
$$g(x) = \hat{w}^T x + \hat{w}_0 = \sum_{n=1}^{N_s} \hat{\alpha}_n \, y_n x_n^T x_n + \hat{w}_0$$

- If g(x) ≥ 0, then x is assign to class 0

- If g(x) < 0, then x is assign to class 1

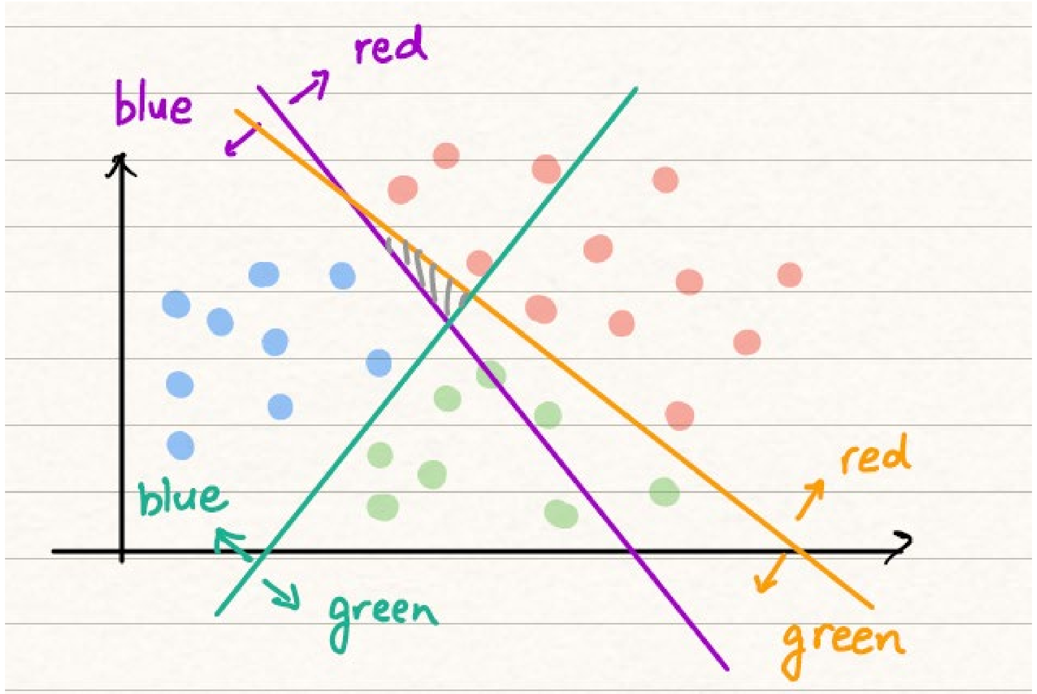
Multiclass Classification

- Support vector machine is a binary (two class) classifier.
- How to realize multiclass classification by binary classifier?



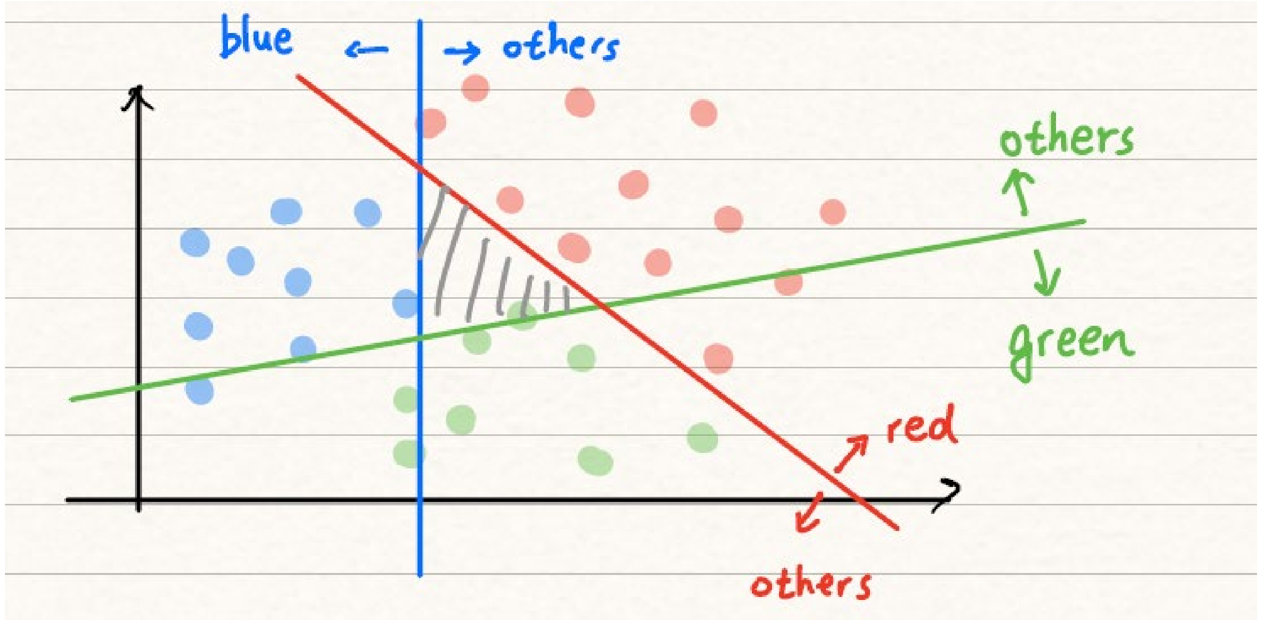
One-versus-one

- Choose two classes and find the decision boundary every time.
- Example: Three classes. C1 vs C2, C1 vs C3, C2 vs C3



One versus others

- Choose one class and set all others as another class. Find the decision boundary.



Which one is better ?

Implement o-v-o

- For the multiclass classification problem, we go through all cases of comparison and decide the data belongs to the most possible class by voting. (For the tie vote case, we select the class which has largest margin.)
- Step1. Training and predicting in every cases to get the votes.
- Step2. Check the number of the most votes class is one or not
- Step3.
  - If number of most votes is one, we get the most possible class.
  - If not, compare the margin and choose the class with largest margin.