

Học có giám sát

Supervised learning

Nội dung

- Dẫn nhập
- Giới thiệu về học máy
- Một số bộ phân lớp phổ biến
 - K-NN
 - Decision Tree
 - Support Vector Machine (SVM)
- Thực hành: Phân loại ảnh

Bài toán: Phát hiện biển số xe

- Mục đích: Xác định vị trí của biển số trong ảnh



- Giải pháp
 - Trích chọn đặc trưng (features extraction): HOG, SIFT,...
 - Lựa chọn mô hình phân lớp
 - Học các tham số của mô hình
 - Sử dụng cửa sổ trượt (Sliding windows)

Training data

- 10% total of images

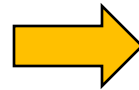
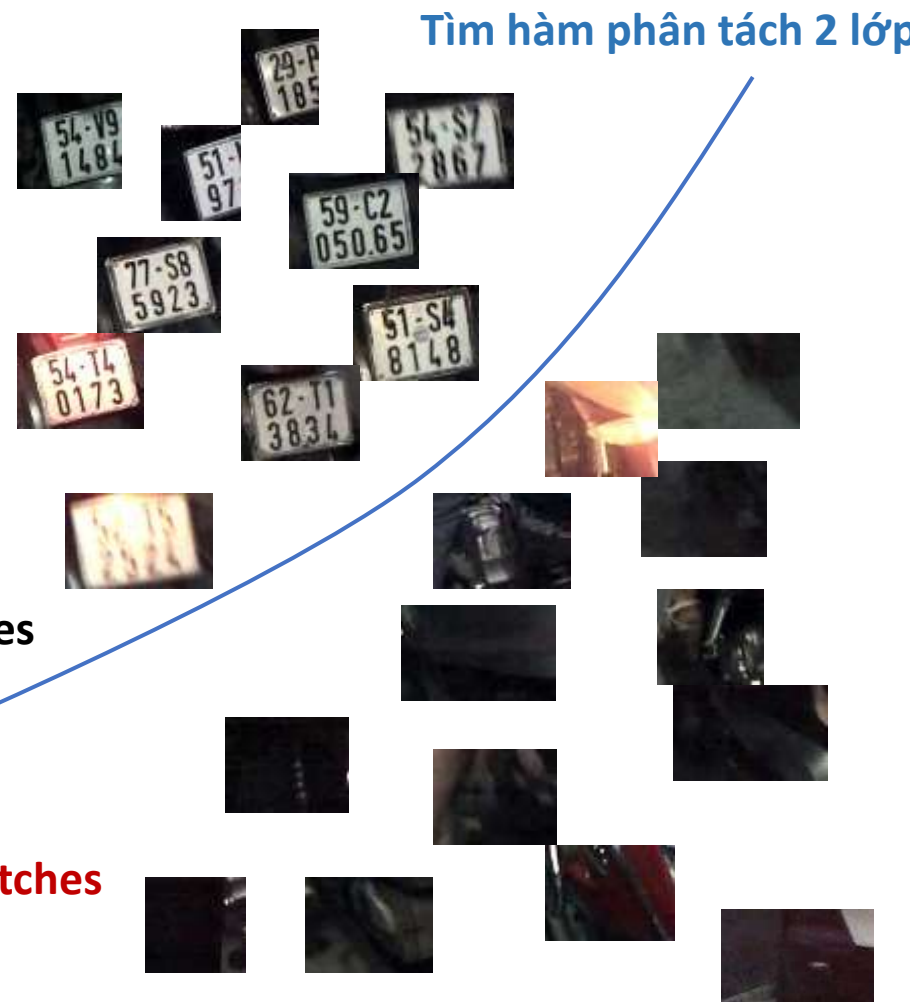
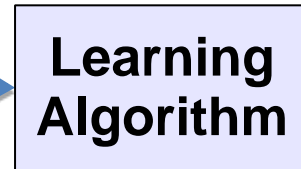
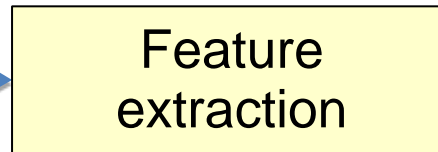


Plate patches

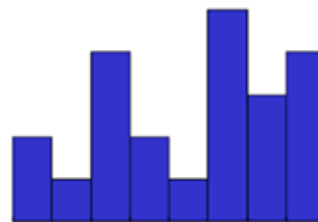
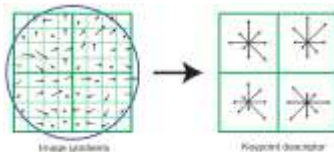
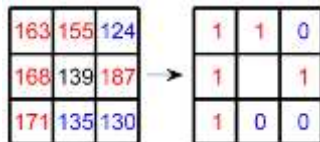
Random patches



Feature Extraction



Feature vectors

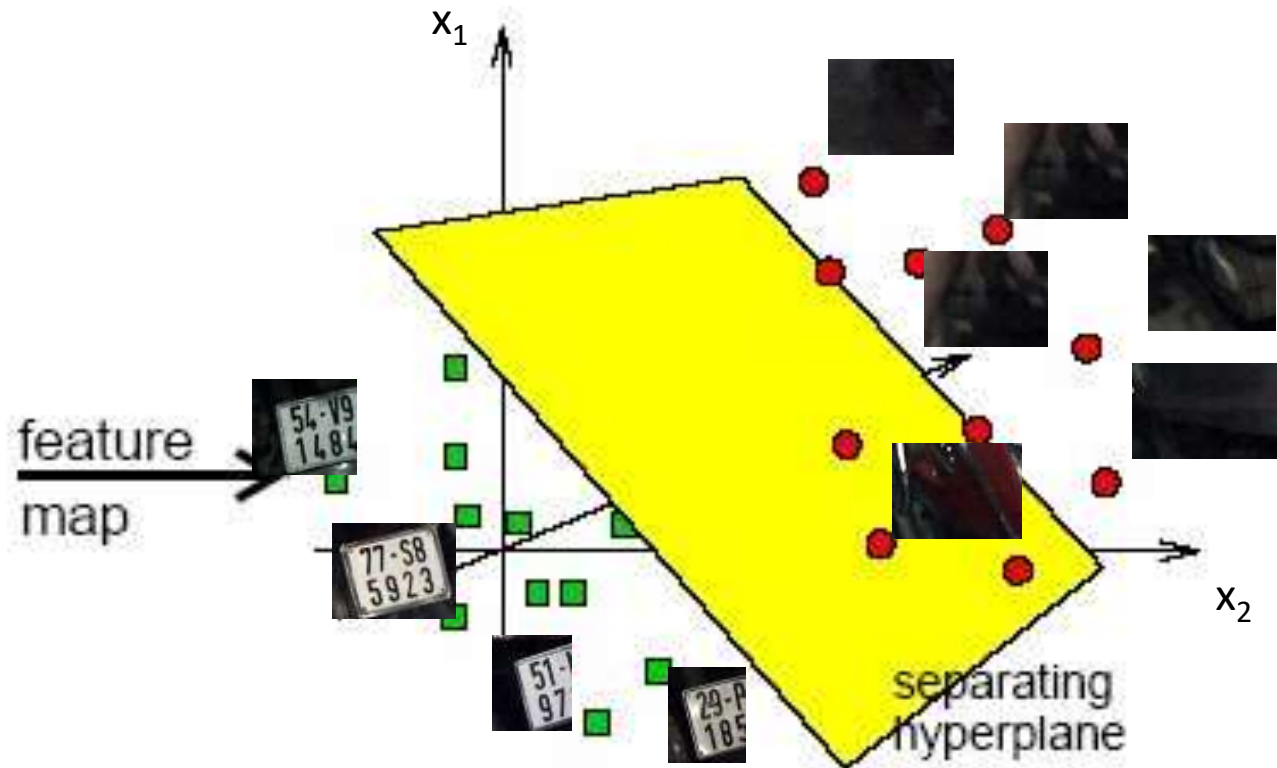


VD: HoG (Histogram
of Gradient)

- Hand-designed Features (e.g., SIFT, SURF ...)
- Parameters:
Hog Cell Size,

Learning a classifier model

- SVM (Support Vector Machine)



Given a feature $x \rightarrow \text{sign}(f(x, w))$ $\begin{cases} = 1 \rightarrow \text{Positive} \\ = -1 \text{ Otherwise} \end{cases}$

Best result

- 10 % for Training; 90% for testing



- Evaluation results:
 - Missed detection: 2/50 (4%)
 - False Alarm Rate (Wrong detection): 2/50 (4%)

Worse result

- Only update HogCellSize parameter
 - HogCellSize = 32 (= 8 for the best result)



- Evaluation results
 - Plate region is small size
 - Missing rate >> ; False alarm Rate >>

Giới thiệu về Học máy

Machine Learning

- Formal definition: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E " - *Tom M. Mitchell*
- Another definition: "The goal of machine learning is to program computers to use **example data** or past experience to solve a given problem." — *Introduction to Machine Learning, 2nd Edition, MIT Press*

Machine learning = Improve the performance at a task with experiences

- Task (problem) **T**
- Experience (data) **E**
- Performance (evaluation) **P**

• Classification

From: cheapsales@buystufffromme.com
To: ang@cs.stanford.edu
Subject: Buy now!

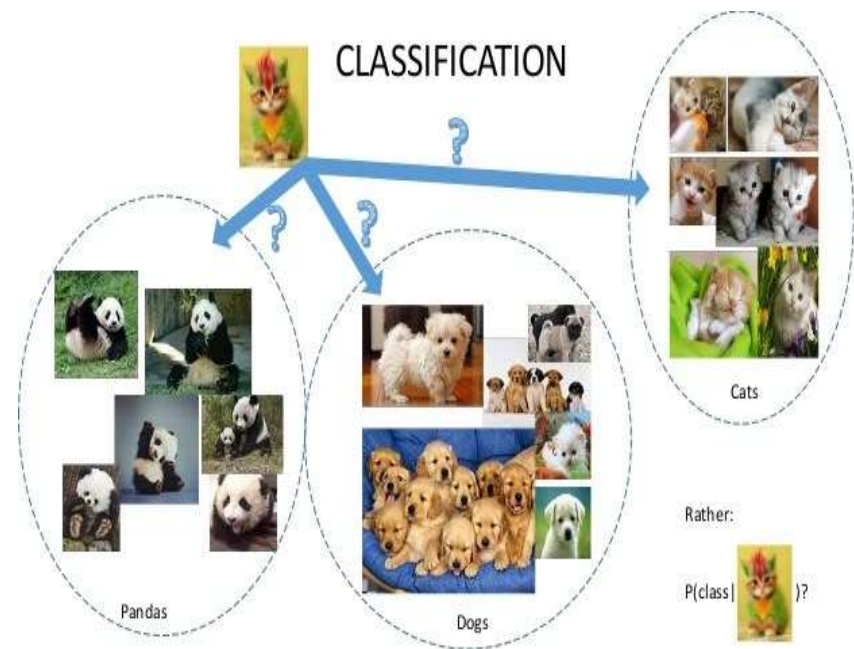
Deal of the week! Buy now!
Rolex watches - \$100
Medicine (any kind) - \$50
Also low cost M0rgages
available.

Spam

From: Alfred Ng
To: ang@cs.stanford.edu
Subject: Christmas dates?

Hey Andrew,
Was talking to Mom about plans
for Xmas. When do you get off
work. Meet Dec 22?
Alf

Non-spam



- Object detection

(Prof. H. Schneiderman)



Example training images
for each orientation



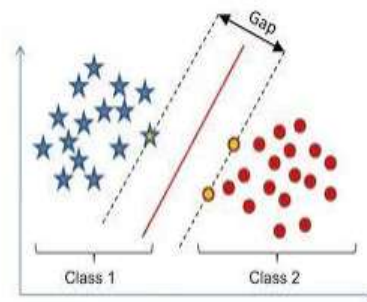
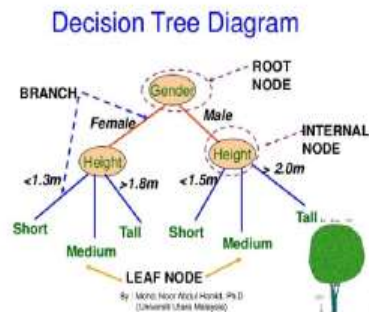
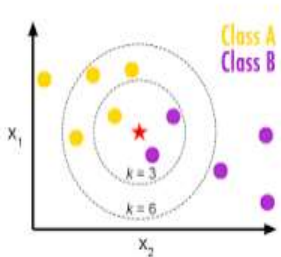
Giới thiệu về học máy

- Học có giám sát (Supervised Learning) vs. Học không có giám sát (Unsupervised Learning)
 - Supervised: Kết quả đầu ra của bài toán (phân lớp) được “định hướng” bởi nhãn của tập dữ liệu huấn luyện (Training)
 - Unsupervised: Nhãn được gán phụ thuộc vào kết quả phân cụm (cluster) của tập dữ liệu đầu vào

	<i>Supervised Learning</i>	<i>Unsupervised Learning</i>
<i>Discrete</i>	classification or categorization	clustering
<i>Continuous</i>	regression	dimensionality reduction

Học có giám sát

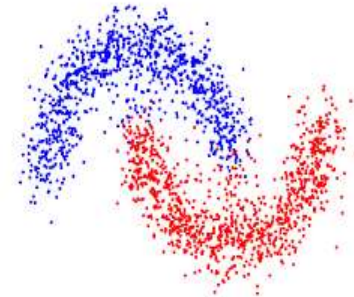
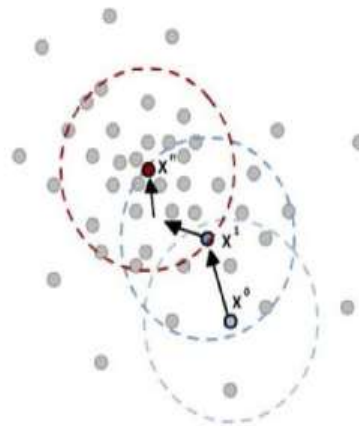
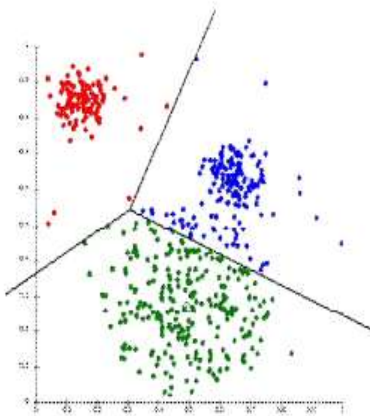
- Học có giám sát (Supervised learning)
 - Bài toán phân lớp đơn giản kNN và Hồi quy tuyến tính
 - Cây phân lớp (Decision Tree)
 - Support Vector Machines (SVM)
 - Naïve Bayes



$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Học không có giám sát

- K-means
- Mean-shift
- Phân cụm phổ (Spectral Clustering)



Machine learning frame-work

- Apply a prediction function to a feature representation of the image to get the desired output:

$f(\text{apple image}) = \text{"apple"}$

$f(\text{tomato image}) = \text{"tomato"}$

$f(\text{cow image}) = \text{"cow"}$

Machine learning framework

$$f(\mathbf{x}) = y$$

Prediction function Image feature Output (label)

Training: Given a *training set* of labeled examples:

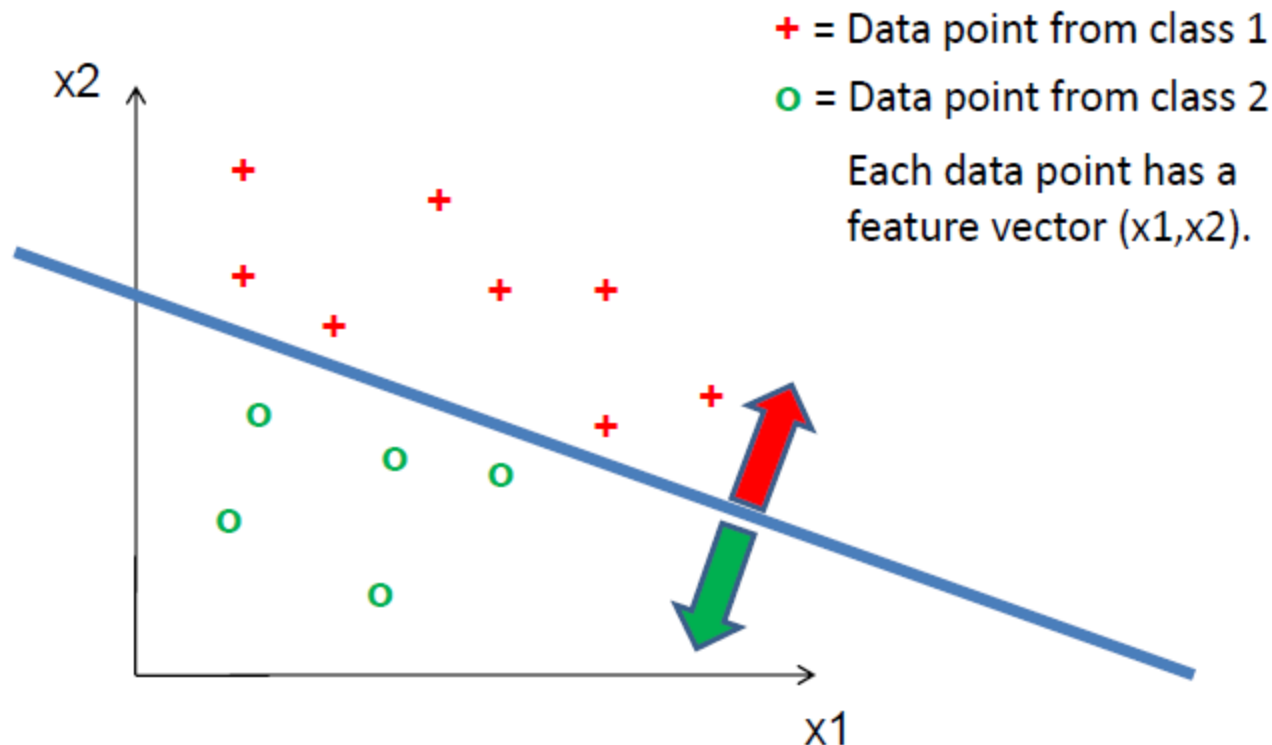
$$\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$$

Estimate the prediction function f by minimizing the prediction error on the training set.

Testing: Apply f to a unseen *test example* \mathbf{x} and output the predicted value $y = f(\mathbf{x})$ to *classify* \mathbf{x} .

Với một bộ phân lớp

- Cho một tập các feature vectors và các nhãn tương ứng \rightarrow xây dựng hàm dự đoán nhãn với tập features đưa vào



Dữ liệu để xây dựng mô hình

Training
Images



- Train classifier

Validation
Images



- Measure error
- Tune model hyperparameters

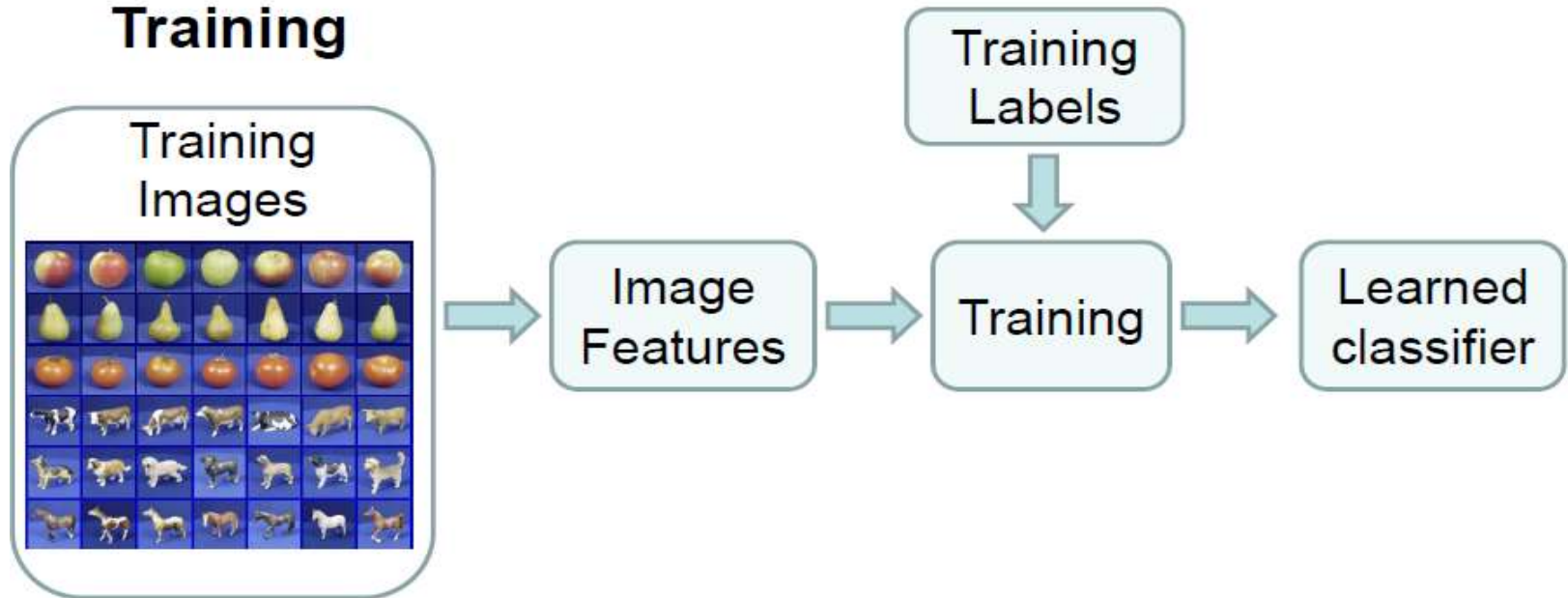
Testing
Images



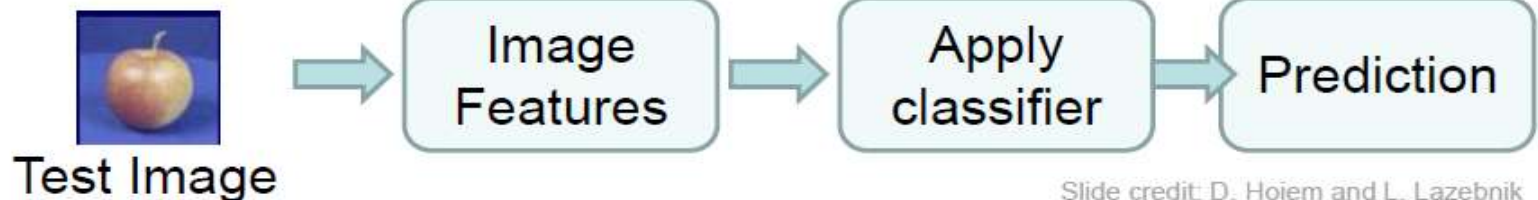
- Secret labels
- Measure error

Các bước thực hiện

Training



Testing



Vấn đề giải quyết

- Thuật toán học để sao cho mô hình học là tốt nhất với các dữ liệu mới → tính tổng quát hóa



Training set (labels known)



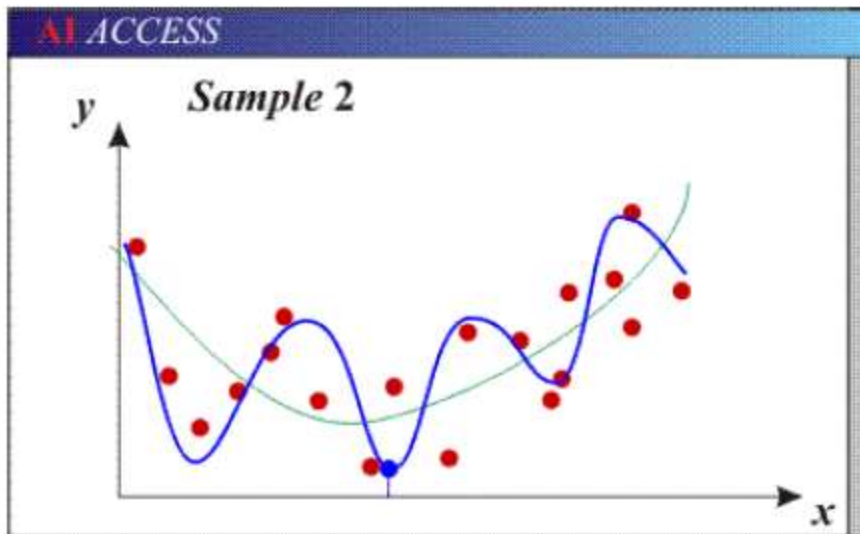
Test set (labels unknown)

Một số vấn đề với mô hình đã học

- **Bias:** how much the average model over all training sets differs from the true model.
 - Error due to inaccurate assumptions/simplifications made by the model.
- **Variance:** how much models estimated from different training sets differ from each other.
- **Underfitting:** model is too “simple” to represent all the relevant class characteristics
 - High bias (few degrees of freedom) and low variance
 - High training error and high test error
- **Overfitting:** model is too “complex” and fits irrelevant characteristics (noise) in the data
 - Low bias (many degrees of freedom) and high variance
 - Low training error and high test error

Minh họa lỗi về over-fitting

- **Overfitting:** model is too “complex” and fits irrelevant characteristics (noise) in the data
 - Low bias (many degrees of freedom) and high variance
 - Low training error and high test error

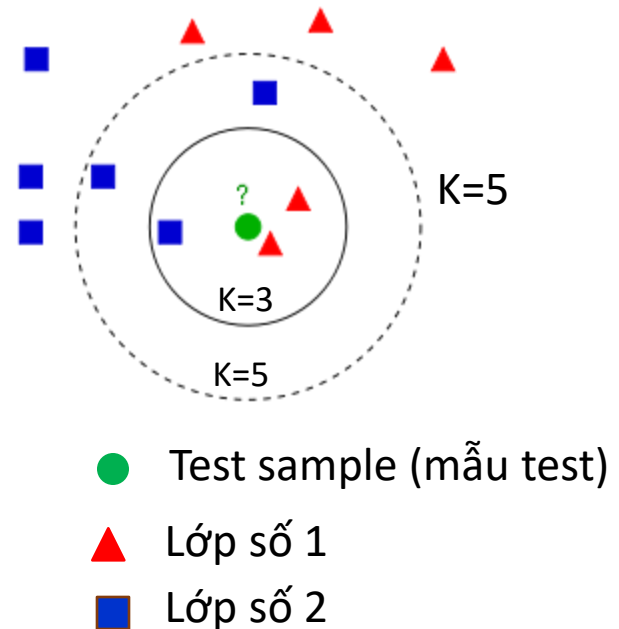


Một số bộ phân lớp thường gặp

- **K-nearest neighbor**
- **Boosted Decision Trees**
- **SVM**
- Neural networks (+CNNs)
- Naïve Bayes
- Bayesian network
- Logistic regression
- Randomized Forests
- Restricted Boltzmann Machines
- ...

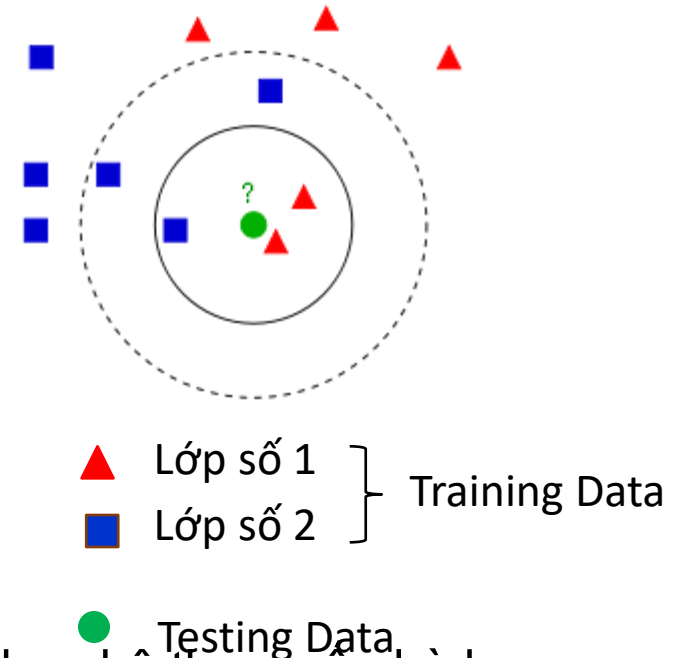
K-NN

- K-NN : K-Nearest Neighbors
 - Là một thuật toán Phân lớp (Classification)/ Hồi quy (Regression)
 - Là một trong số Simplest Algorithms
- Ý tưởng:
 - Bài toán phân lớp: Gán nhãn dựa trên số lượng voting (bầu) trong số K kết quả trả về gần nhất
 - Bài toán hồi quy: Gán nhãn (predict) cho tín hiệu liên tục
 - Chuyển đổi (phân lớp , hồi quy)
- Khó khăn:
 - Chọn công thức tính khoảng cách?
 - Chọn giá trị K?



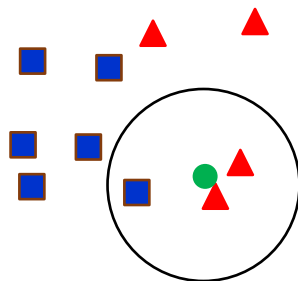
Training và Testing Data

- Training Data
 - Dữ liệu huấn luyện đã biết trước nhãn
 - Thường mất công chuẩn bị dữ liệu
- Testing Data
 - Dữ liệu sẽ được thuật toán gán nhãn
- Ground-truth Data
 - Dữ liệu Test đã có nhãn đúng đắn để đánh giá kết quả gán nhãn
- Validation Data
 - Dữ liệu đã có nhãn được sử dụng để chọn bộ tham số phù hợp
- Lưu ý:
 - Tỷ lệ giữa các nhãn trong training ảnh hưởng đến việc “huấn luyện”
 - Tỷ lệ Training/Testing data ảnh hưởng đến kết quả thuật toán

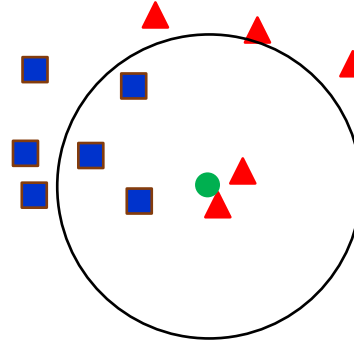


Training và Testing Data

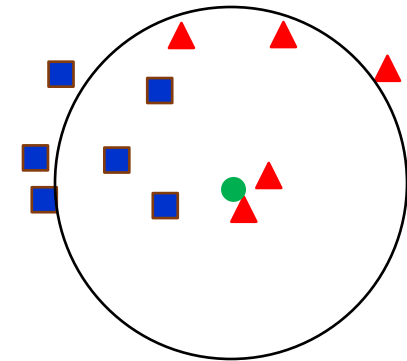
- Tỷ lệ thông thường (training/testing/validation): 70% : 10% : 20%
- Dùng validation data để chọn tham số K
 - Chiến lược: Thử lần lượt từng giá trị của K, đánh giá độ chính xác của thuật toán trên tập Validation, dừng khi accuracy đạt ngưỡng



K=3



K=5



K=7

- Chuẩn bị bộ dữ liệu thường mất thời gian + công sức

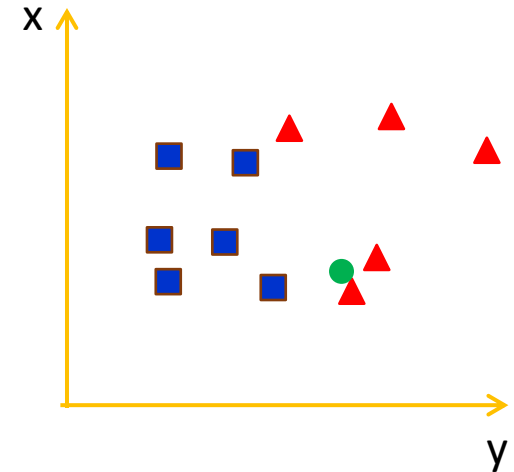
Feature vectors

- Vector đặc trưng:
 - Mô tả đặc điểm của đối tượng
 - Trích chọn vector đặc trưng (Feature extraction)
- Cần phát biểu được bài toán dưới dạng:
Ma trận $A = [\text{Quan sát} \times \text{Feature Vectors}]$

$\underbrace{\hspace{10em}}_{\text{Từ Dataset (m Quan sát)}} \quad \underbrace{\hspace{10em}}_{\text{Từ Feature extraction (n chiều)}}$

$$A = [m \text{ hàng} \times n \text{ cột}]$$

- Các vấn đề liên quan:
 - Chuẩn hóa (Normalization)
 - Lựa chọn đặc trưng (Feature Selection)
 - Biểu diễn đặc trưng (Feature Space): Linear, Non-Linear Space



Distances

- Là khoảng cách giữa hai feature vectors
- Khoảng cách Euclidean:

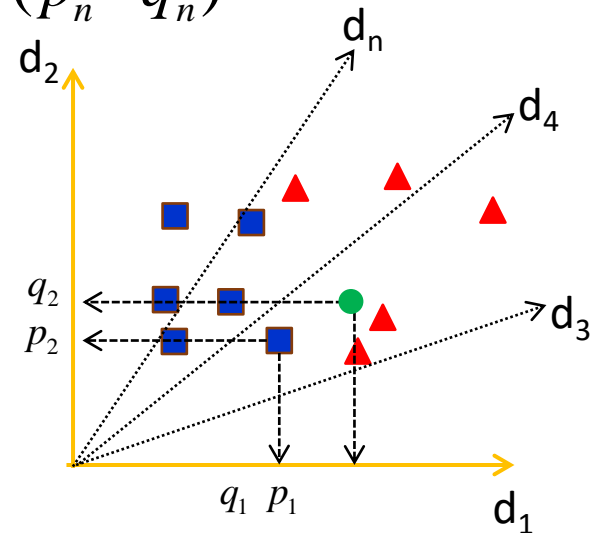
$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}$$

- Khoảng cách Mahalanobis:

$$d(\vec{p}, \vec{q}) = \sqrt{(\vec{p} - \vec{q})^T S^{-1} (\vec{p} - \vec{q})}$$

S^{-1} : is covariance matrix

- Other distances:
 - Absolute distance



Áp dụng đối với K-NN

- Xây dựng ma trận quan sát từ dữ liệu huấn luyện

$$A = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^D \\ x_2^1 & x_2^2 & \dots & x_2^D \\ \vdots & \vdots & \ddots & \vdots \\ x_N^1 & x_N^2 & \dots & x_N^D \end{bmatrix}$$

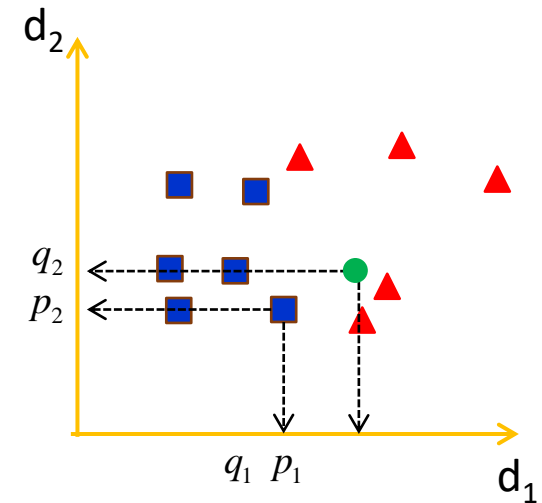
$x_N^1 = [x_N^1, x_N^2, \dots, x_N^D]$

$$Y = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

Nhãn : [0 , 1]

- Tính khoảng cách Euclidean cho mỗi mẫu test s

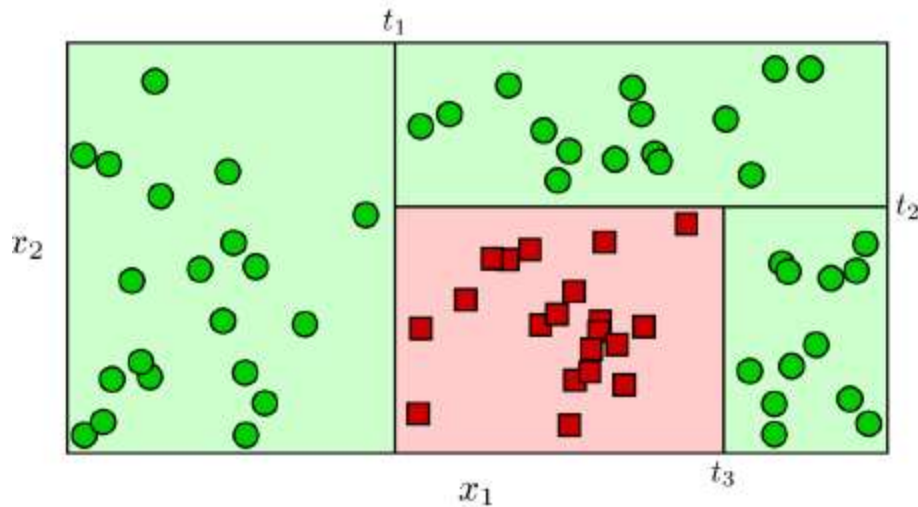
$$D = \begin{bmatrix} d(x_1, s) & y_1 \\ d(x_2, s) & y_2 \\ \vdots & \vdots \\ d(x_N, s) & y_N \end{bmatrix}$$



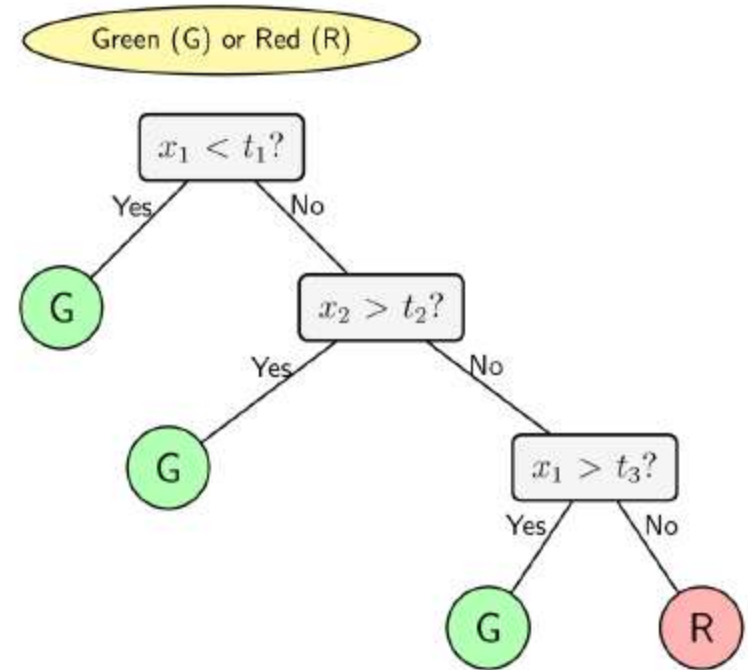
- Sắp xếp hàng của D theo thứ tự giảm dần của $d(x_i, s)$

- Lấy K nhãn đầu tiên ($y_1 \dots y_K$) và gán nhãn nào số lượng được vote nhiều nhất

Thuật toán Decision tree

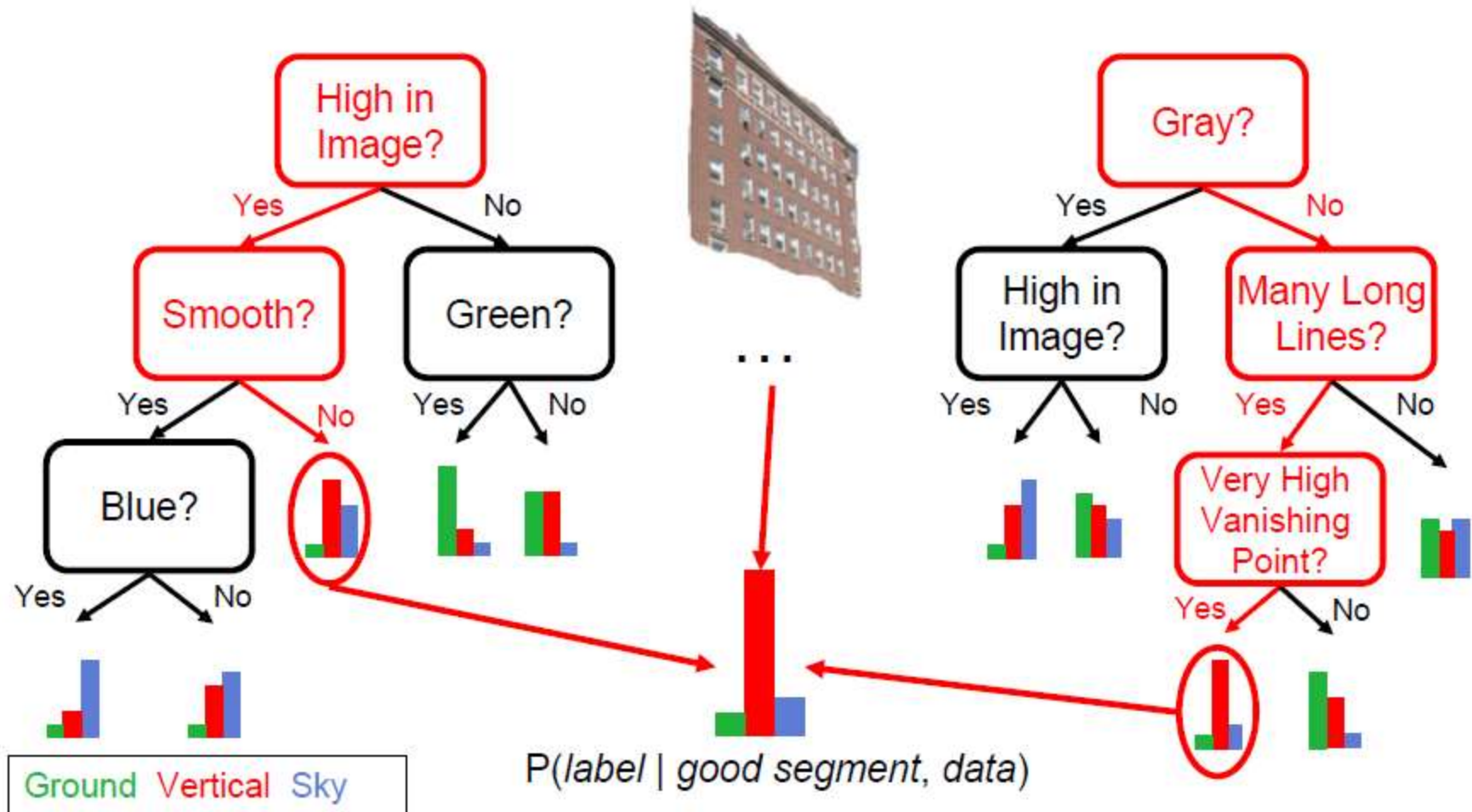


(a)



(b)

Thuật toán Boosted Decision Tree

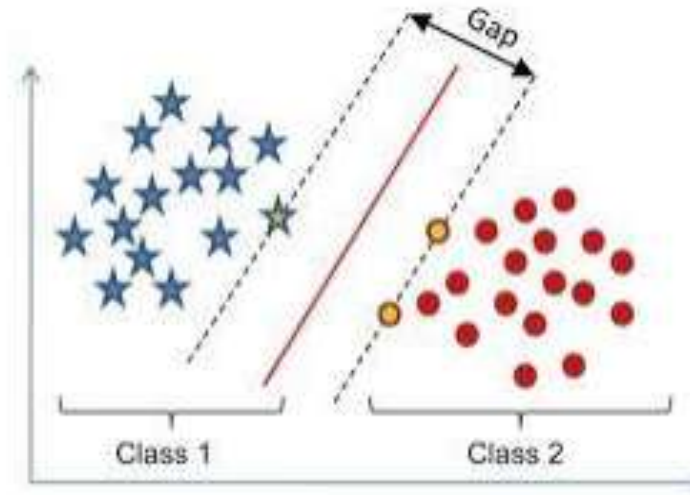


Thuật toán Boosted Decision Tree

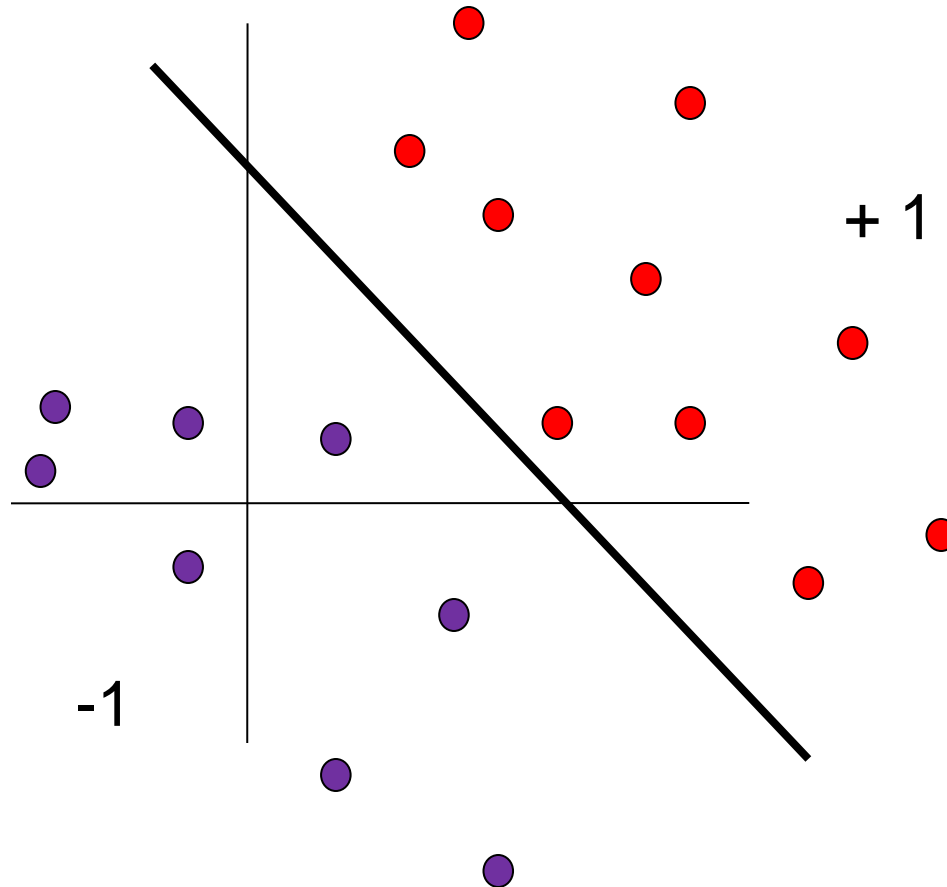
- Flexible: can deal with both continuous and categorical variables
- How to control bias/variance trade-off
 - Size of trees
 - Number of trees
- Boosting trees often works best with a small number of well-designed features

Support Vector Machine

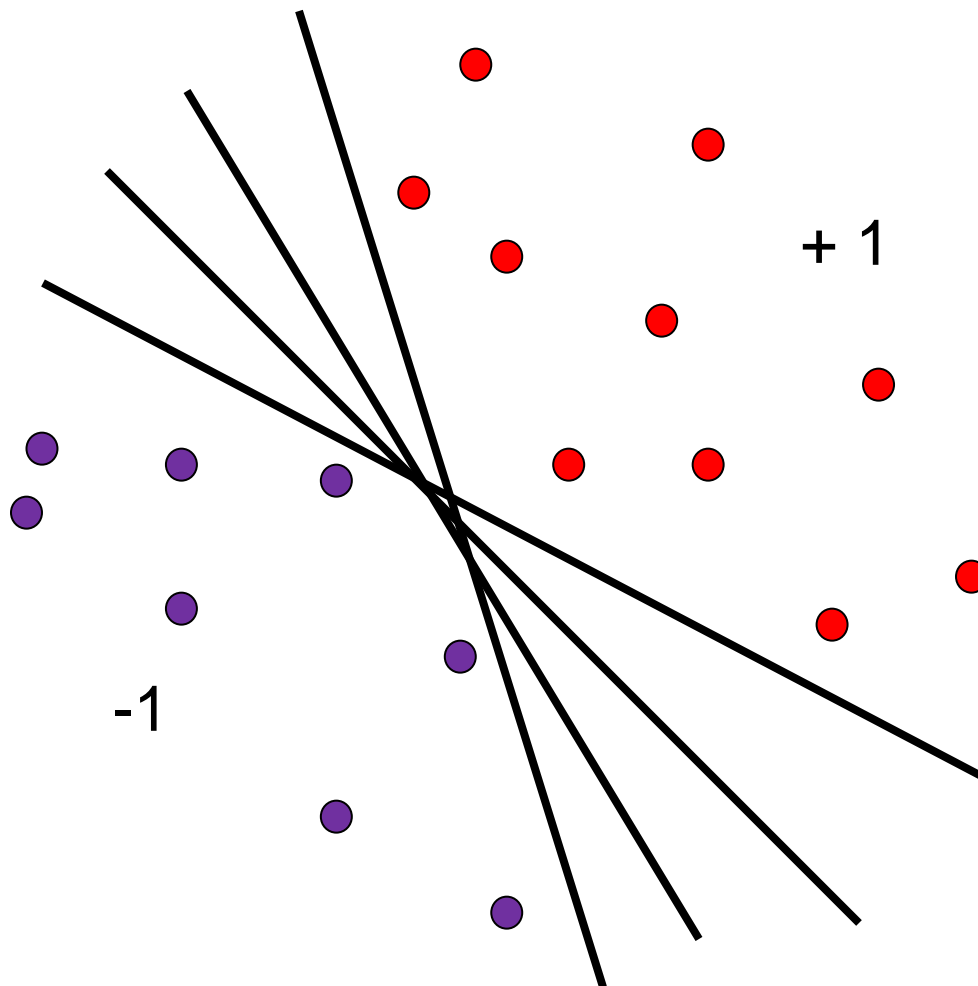
- Máy vec-tơ hỗ trợ (SVM)



Phân loại tuyến tính



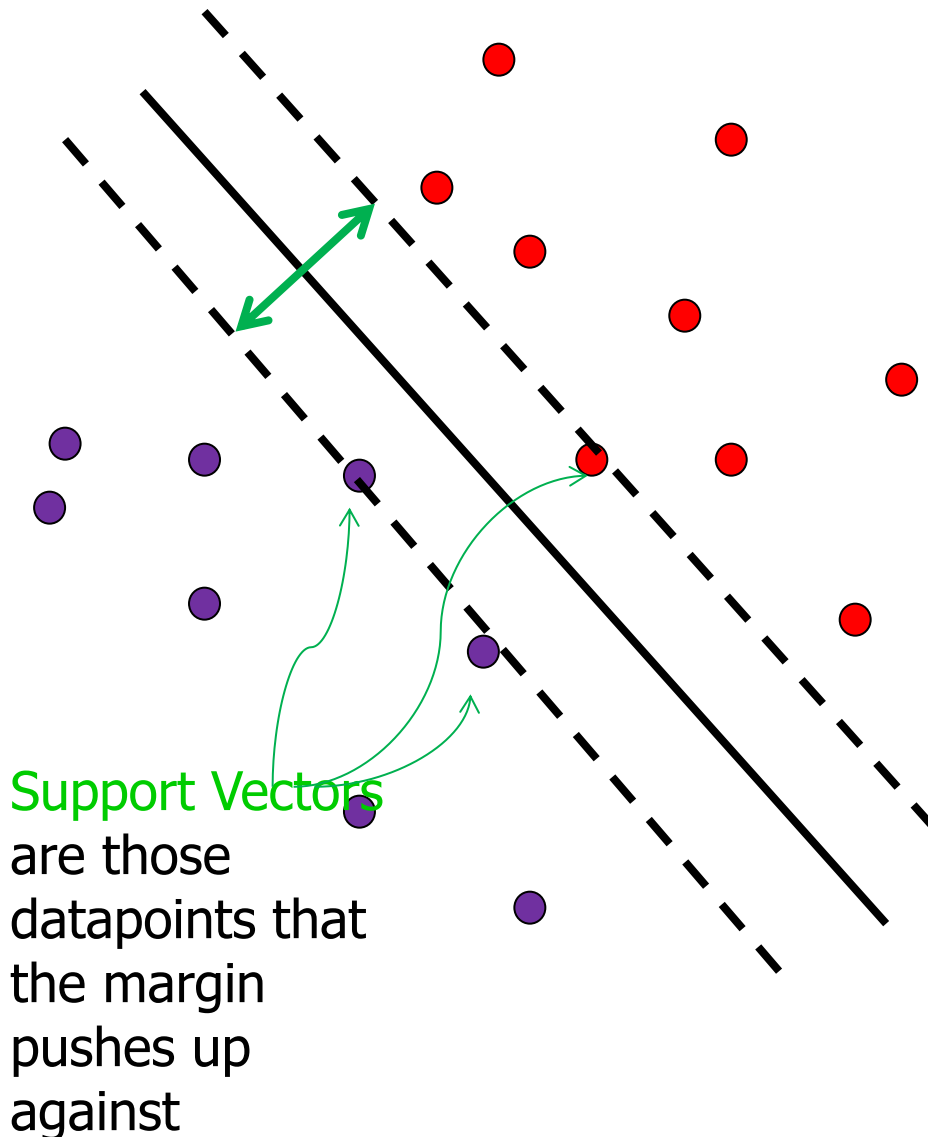
Phân loại tuyến tính



- Tìm hàm tuyến tính phân chia mẫu dương (positive samples) và mẫu âm (negative samples)

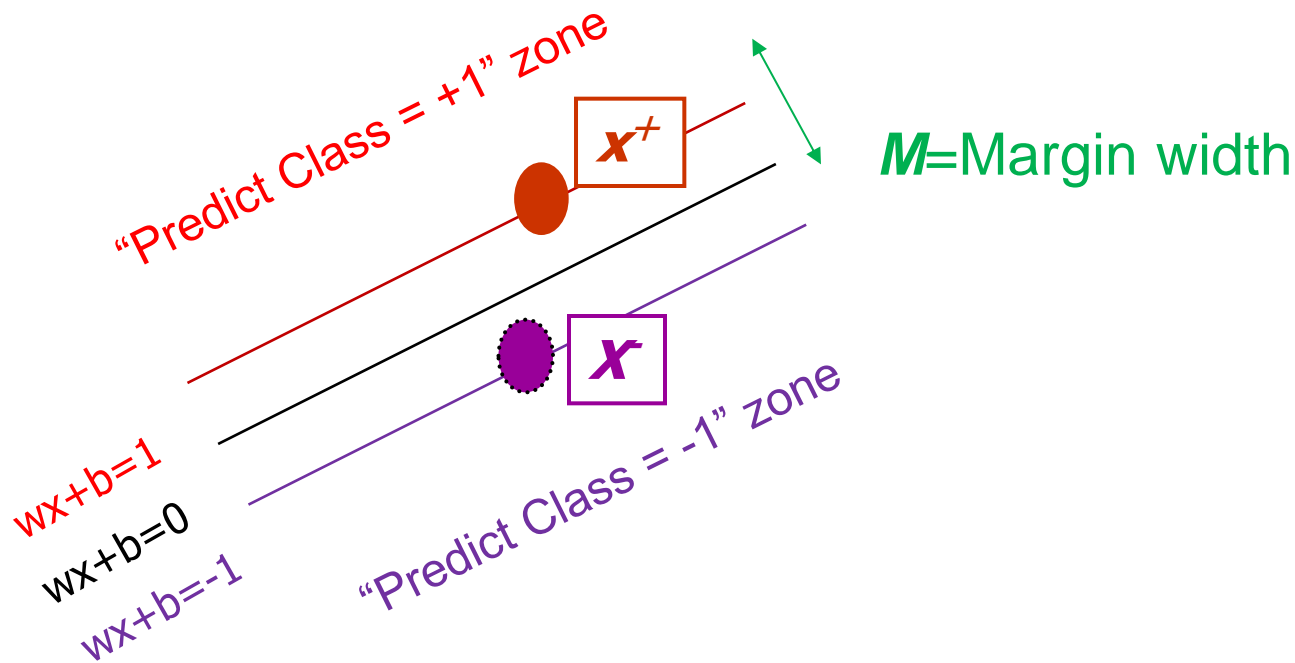
Hàm nào tốt nhất?

Support Vector Machines (SVMs)



- Là bộ phân loại dựa trên xây dựng **siêu phẳng** phân cách tối ưu (*đường thẳng trong trường hợp 2d*)
- Cực đại **lề (margin)** giữa các mẫu positives và các mẫu negatives

Mô hình của SVM tuyến tính



What we know:

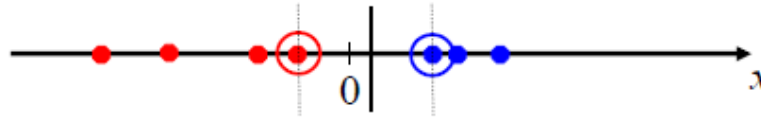
- $w \cdot x^+ + b = +1$
- $w \cdot x^- + b = -1$

$$\Rightarrow w \cdot (x^+ - x^-) = 2$$

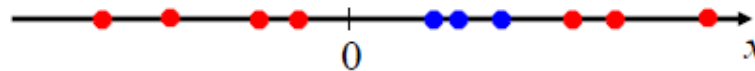
$$M = \frac{(x^+ - x^-) \cdot w}{|w|} = \frac{2}{|w|}$$

Bộ phân lớp Non-linear SVM

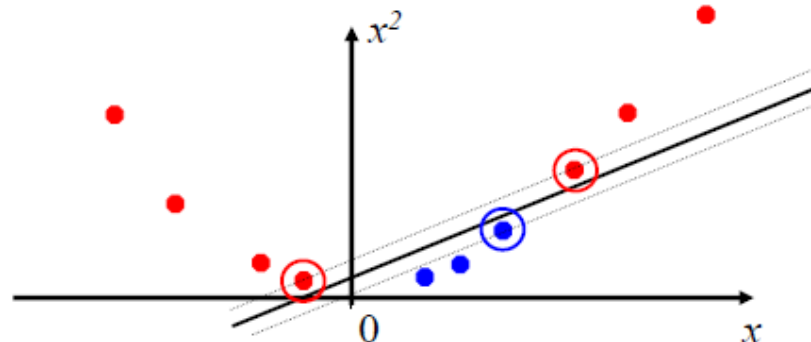
- Datasets that are linearly separable work out great:



- But what if the dataset is just too hard?

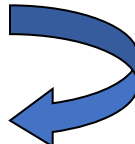


- We can map it to a higher-dimensional space:



Mô hình Toán

- ♦ Goal: 1) Correctly classify all training data

$$\begin{array}{ll}
 wx_i + b \geq 1 & \text{if } y_i = +1 \\
 wx_i + b \leq 1 & \text{if } y_i = -1 \\
 y_i(wx_i + b) \geq 1 & \text{for all } i
 \end{array}$$


2) Maximize the Margin
same as minimize

$$M = \frac{2}{|w|}$$

$$\frac{1}{2} w^t w$$

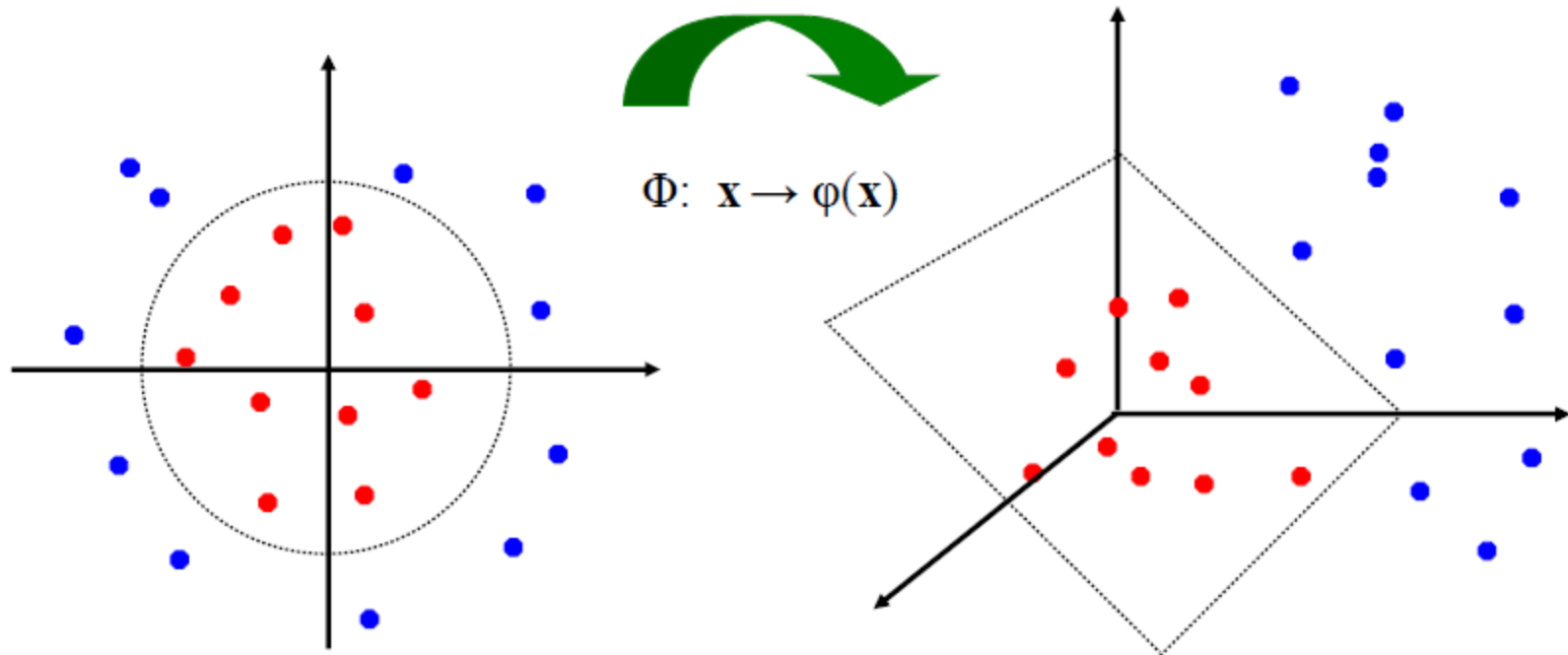
- ♦ **Giải bài toán tối ưu để tìm w and b**

- ♦ Minimize $\Phi(w) = \frac{1}{2} w^t w$

subject to $y_i(wx_i + b) \geq 1 \quad \forall i$

Bộ phân lớp non-linear SVM

- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:



Thực hiện non-linear SVM

- *The kernel trick*: instead of explicitly computing the lifting transformation $\varphi(\mathbf{x})$, define a kernel function K such that

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$$

- This gives a nonlinear decision boundary in the original feature space:

$$\sum_i \alpha_i y_i \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}) + b = \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

Một vài dạng hàm Kernels

- Linear kernel:

$$K(x_i, x_j) = x_i^T x_j$$

- Histogram intersection kernel:

$$K(x_i, x_j) = \sum_k \min(x_i(k), x_j(k))$$

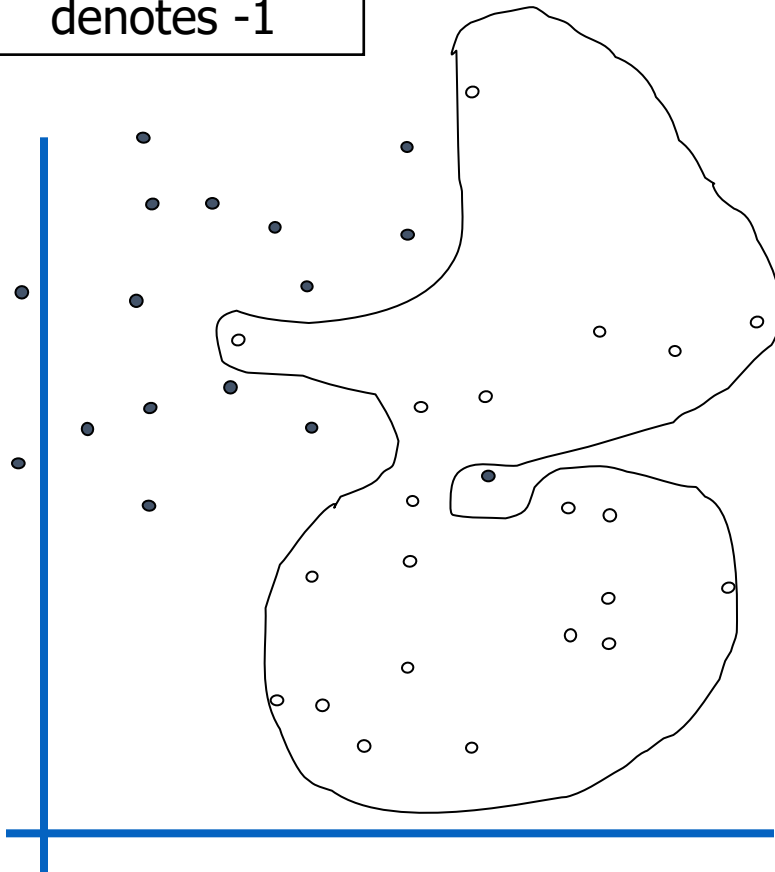
- Generalized Gaussian kernel:

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

Khi huấn luyện SVM, cần xác định các tham số tương ứng với mỗi loại kernel được sử dụng.

Khi dữ liệu có nhiễu

- denotes +1
- denotes -1



- ◆ Biên cứng (Hard Margin): Các điểm dữ liệu phải được phân lớp chính xác, không có sai số khi huấn luyện
- ◆ Điều gì xảy ra khi dữ liệu có nhiễu?
 - Giải pháp 1: use very powerful kernels

OVERFITTING!

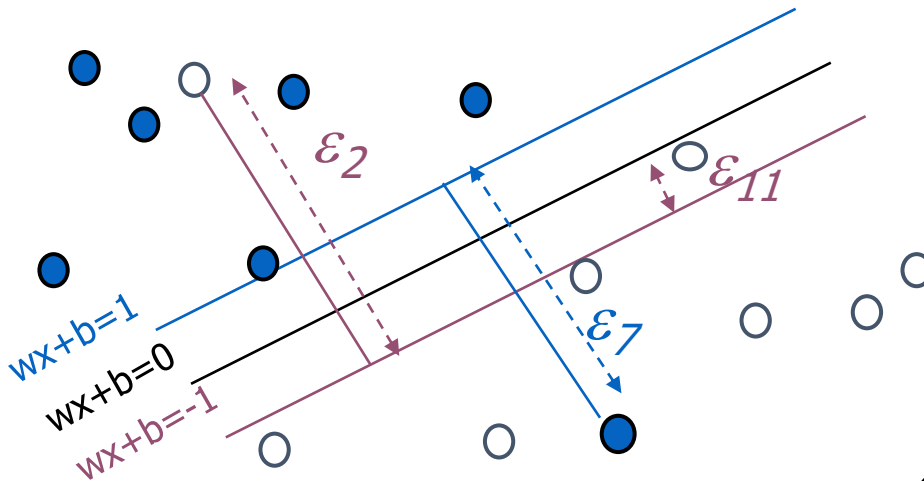
Phân lớp với biên mềm

Thêm biến giả ξ_i cho phép có sai số cho những điểm khó phân lớp hoặc những điểm là nhiễu.

Điều kiện tối ưu trở thành:

Minimize

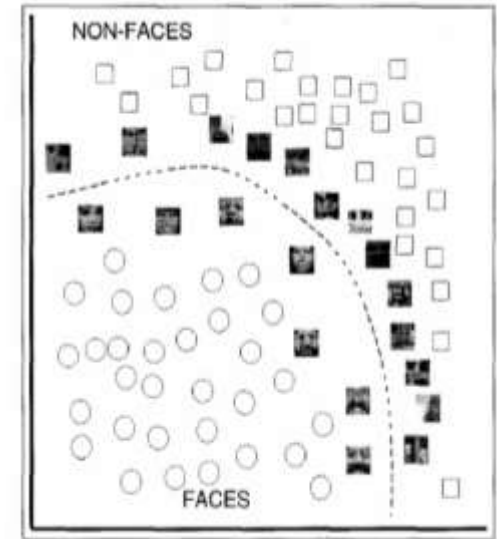
$$\frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{k=1}^R \varepsilon_k$$



C là tham số phải xác định khi huấn luyện SVM.

SVMs cho bài toán phân loại

1. Trích xuất véc tơ đặc trưng cho các mẫu huấn luyện.
2. Lựa chọn một hàm nhân
3. Tính ma trận khoảng cách hàm nhân *kernel matrix* giữa các mẫu huấn luyện
4. Sử dụng “kernel matrix” huấn luyện SVM: xác định véc tơ hỗ trợ và các tham số \mathbf{w} của mô hình
5. Phân loại mẫu dữ liệu mới: tính khoảng cách hàm nhân giữa mẫu mới và các véc tơ hỗ trợ, lắp \mathbf{w} vào tính toán và kiểm tra dấu của kết quả biểu thức đầu ra.



SVMs nhiều lớp

- Kết hợp nhiều SVM nhị phân
- **One vs. all**
 - Training: huấn luyện SVM cho từng lớp vs. phần còn lại
 - Testing: áp dụng từng SVM với mẫu test và gán nhãn cho lớp của SVM trả lại kết quả score cao nhất.
- **One vs. one**
 - Training: huấn luyện SVM cho từng cặp hai lớp
 - Testing: mỗi SVM “vote” một nhãn để gán cho mẫu test

SVMs: Ưu và nhược điểm

- Pros

- Sử dụng SVM hàm nhân rất linh hoạt và mạnh mẽ
- Thường số lượng véc tơ hỗ trợ khá thưa – hiệu quả đối với thời gian test
- Kết quả rất tốt trong thực tế, thậm chí đối với trường hợp tập huấn luyện bé

- Cons

- Không có SVM “trực tiếp” cho bài toán đa lớp, phải kết hợp các SVMs nhị phân
- Việc lựa chọn hàm nhân tốt nhất không đơn giản
- Độ phức tạp tính toán, bộ nhớ
 - Lúc huấn luyện, phải tính ma trận hàm nhân cho các cặp mẫu dữ liệu
 - Thời gian huấn luyện có thể rất lâu đối với các bài toán kích thước lớn

Khuyến nghị các bước sử dụng SVM cho người mới bắt đầu

- Chuyển dữ liệu về format sử dụng cho các gói công cụ SVM
 - Chuẩn hóa dữ liệu
 - [Thử với linear kernel nếu là bài toán đơn giản]
 - Thử với RBF kernel $K(\mathbf{x}, \mathbf{y}) = e^{-\gamma \|\mathbf{x} - \mathbf{y}\|^2}$
 - Dùng phương pháp kiểm tra chéo (cross-validation) để tìm các tham số C và γ tốt nhất
 - Dùng các giá trị C and γ tìm được để huấn luyện trên toàn bộ dữ liệu
 - Kiểm thử mô hình thu được (test).
-
- TK: <https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>

Thực hành

- Phân loại ảnh