Data Classification: Overview

Yuh-Jye Lee

National Taiwan University of Science and Technology

TIGP bioinformatics program, March 07, 2014

Mathematical Background You Will Need in the Class

- Mutli-Variable Calculus
  - What is the *gradient* of a differentiable function?
  - What is the *Hessian* of a twice differentiable function?
- Linear Algebra
  - How to compute the distance between two parallel hyperplanes in R^n?
  - Eigenvalue, positive definite matrix, inner product, projection matrix etc.
- Probability
  - Random variables, probability distributions, conditional probability, Bayes′ rule, expected value, variance etc.
- Statistics
  - Testing hypothesis, confidence interval etc.

Software Packages & Datasets

- MLC++
  - Machine learning library in C++
  - http://www.sgi.com/tech/mlc/
- WEKA
  - http://www.cs.waikato.ac.nz/ml/weka/
- Statlib
  - Data, software and news from the statistics community
  - http://lib.stat.cmu.edu
- GALIB
  - MIT GALib in C++
  - http://lancet.mit.edu/ga
- Delve
  - Data for Evaluating Learning in Valid Experiments
  - http://www.cs.utoronto.ca/~delve
- UCI
  - Machine Learning Data Repository UC Irvine
  - http://www.ics.uci.edu/~mlearn/MLRepository.html
- UCI KDD Archive
  - http://kdd.ics.uci.edu/summary.data.application.html

References in this Lecture (and will be very useful in others)

- Trevor Hastie, Rob Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning. Springer, 2001*
- Vladimir N. Vapnik. *The Nature of Statistical Learning Theory. Springer, Second Edition, 1995*.
- Chris Burges. *A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2 (2):121-167, 1998.*
- Ian H. Witten, and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques.Elsevier, Second Edition, 2005*
  WEKA is developed by them.

Major conferences in ML

- ICML (International Conference on Machine Learning)
- ECML (European Conference on Machine Learning)
- UAI (Uncertainty in Artificial Intelligence)
- NIPS (Neural Information Processing Systems)
- COLT (Computational Learning Theory)
- IJCAI (International Joint Conference on Artificial Intelligence)
- …

What is Learning All about?

- Get knowledge of by study, experience, or be taught
- Become aware by information or from observation
- Commit to memory
- Be informed of or receive instruction

A Possible Definition of Learning

- Things learn when they change their behavior in a way that makes them *perform* better in the future.
- Have your shoes *learned* the shape of your foot ?
- In learning the purpose is the learner's, whereas in training it is the teacher's.

Learning & Adaptation

- Machine Learning: 機器學習?
  - Machine → Automatic
  - Learning → Performance is improved
- "Modification of a behavioral tendency by expertise." (Webster 1984)
- "A learning machine, broadly defined is any device whose actions are influenced by past experiences." (Nilsson 1965)
- "Any change in a system that allows it to perform better the second time on repetition of the same task or on another task drawn from the same population." (Simon 1983)
- "An improvement in information processing ability that results from information processing activity." (Tanimoto 1990)

Applications of ML

- Learning to recognize spoken words
  - SPHINX (Lee 1989)
- Learning to drive an autonomous vehicle
  - ALVINN (Pomerleau 1989)
- Learning to pick patterns of terrorist action
- Learning to classify celestial objects
  - (Fayyad et al 1995)
- Learning to play chess
  - Learning to play go game (Shih, 1989)
  - Learning to play world-class backgammon (TD-GAMMON, Tesauro 1992)
- Designing the morphology and control structure of electro-mechanical artifacts
  - GOLEM (Lipton, Pollock 2000)
- IBM Watson Wins Jeopardy, Humans Rally Back (2011)

Types of learning problems

- A rough (and somewhat outdated) classification of learning problems:
  - Supervised learning, where we get a set of training inputs and outputs
    - classification, regression
  - Unsupervised learning, where we are interested in capturing inherent organization in the data
    - clustering, density estimation
  - Semi-supervised learning, in practice, labeled data are very limited but a lot of unlabeled data
  - Reinforcement learning, where we only get feedback in the form of how well we are doing (not what we should be doing)

Learning a Class from Examples

- Suppose we want to learn a class (concept) C
  - example: “sports car”
  - given a collection of cars, have people label them as sports car (positive example) or non-sports car (negative example)
  - task: find a description that is shared by all of the positive examples and none of the negative examples
  - Once we have this definition for C, we can
    - predict – given a new unseen car, predict whether or not it is a sports car
    - describe/compress – understand what people expect in a car

Choosing an Input Representation

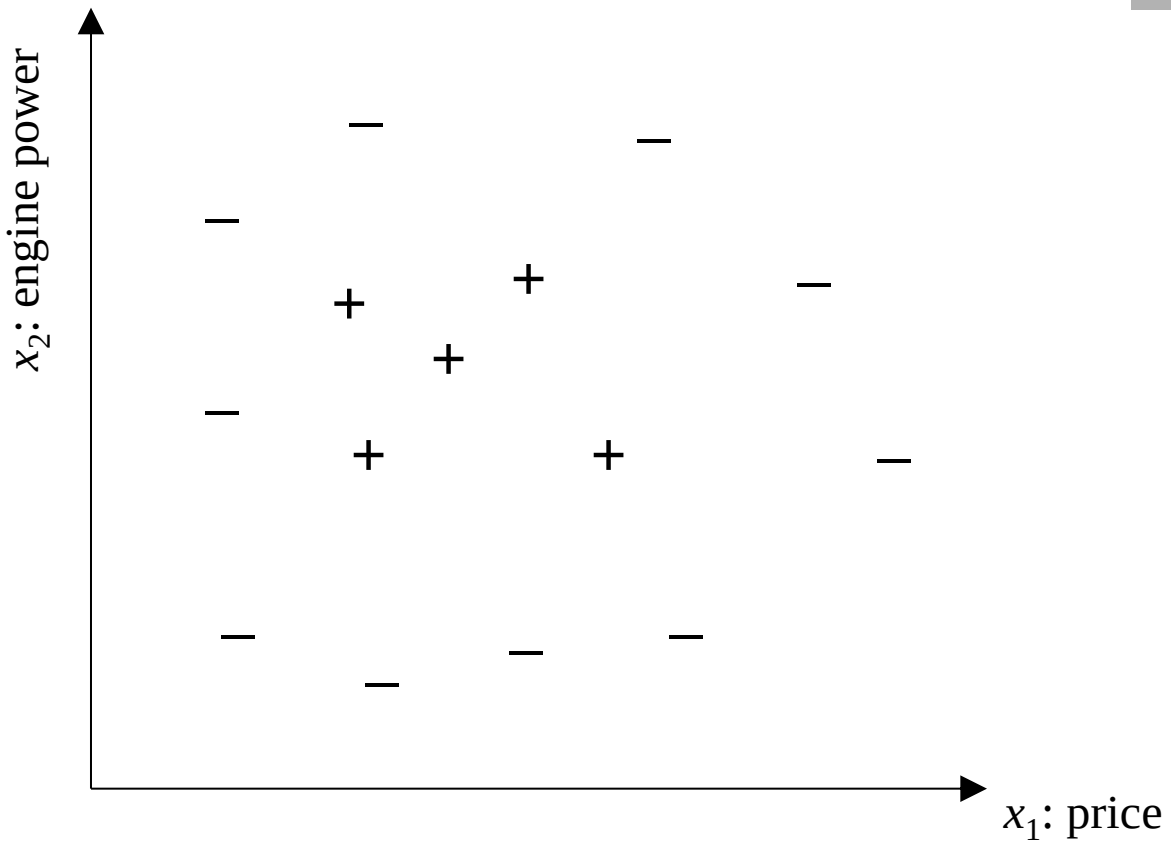
- Suppose that of all the features describing cars, we choose price and engine power. Choosing just two features
  - makes things simpler
  - allows us to ignore irrelevant attributes
- Let
  - x_1 represent the price (in USD)
  - x_2 represent the engine volume (in cm^3)
- Then each car is represented

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

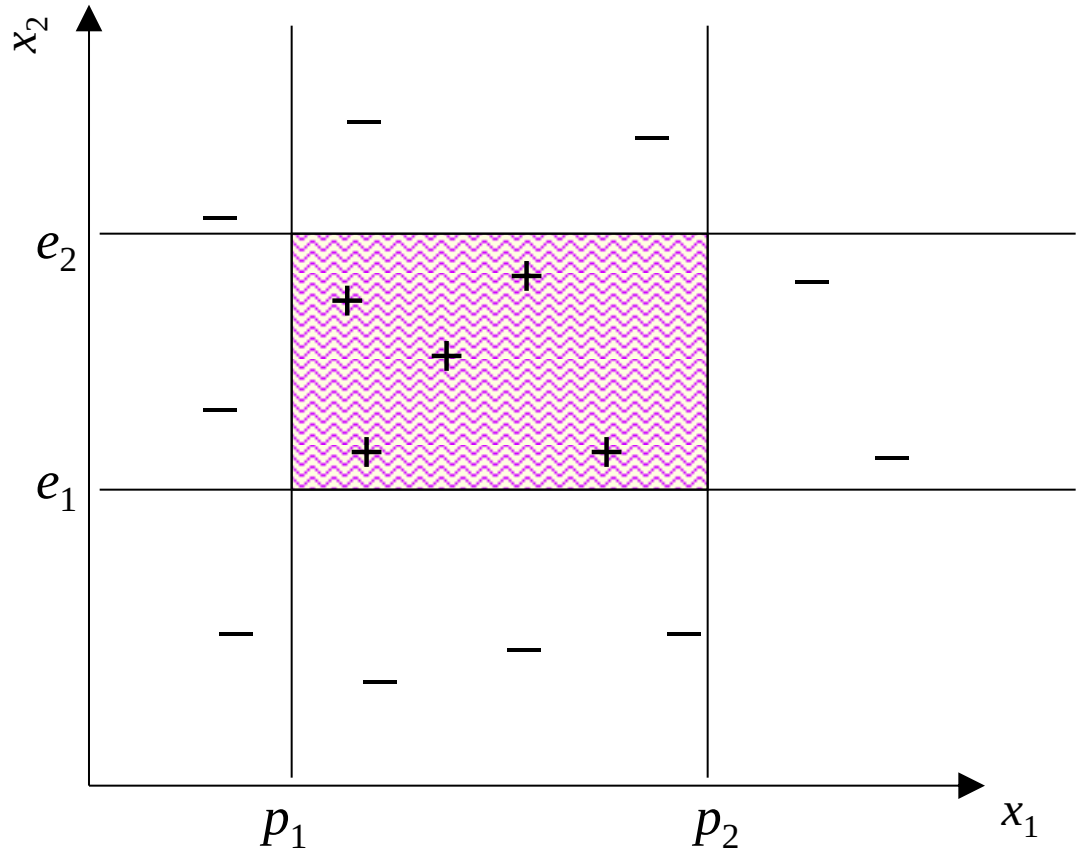
- and its label y denotes its type $y = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is a positive example} \\ -1 & \text{if } \mathbf{x} \text{ is a negative example} \end{cases}$
- each example is represented by the pair $(\mathbf{x}, y)$
- and a training set containing N examples is represented by

$$\mathcal{X} = \{\mathbf{x}^t, y^t\}_{t=1}^{N}$$

Plotting the Training Data



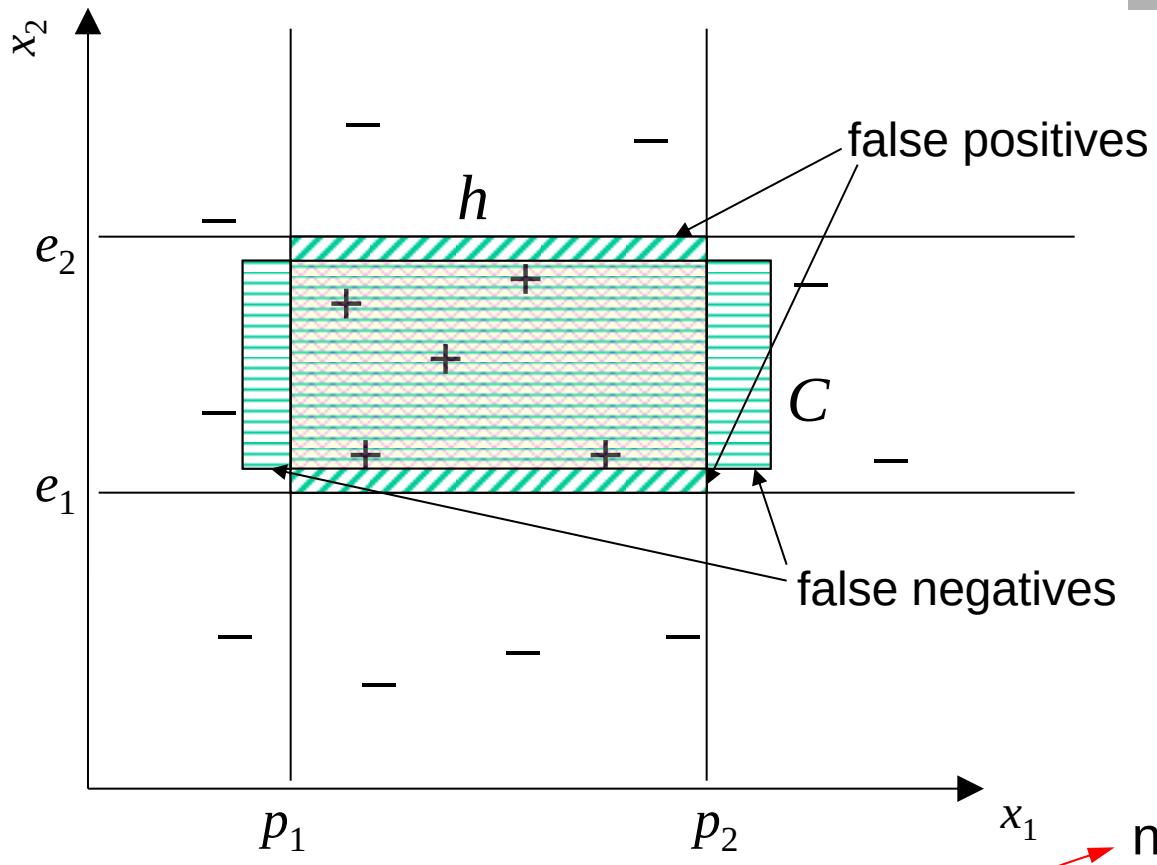
Hypothesis Class



suppose that we think that for a car to be a sports car, its price and its engine power should be in a certain range:

$$(p_1 \leq \text{price} \leq p_2) \text{ AND } (e_1 \leq \text{engine} \leq e_2)$$

Concept Class



no training errors

suppose that the actual class is C

task: find $h \in \mathcal{H}$ that is **consistent** with $\mathcal{X}$

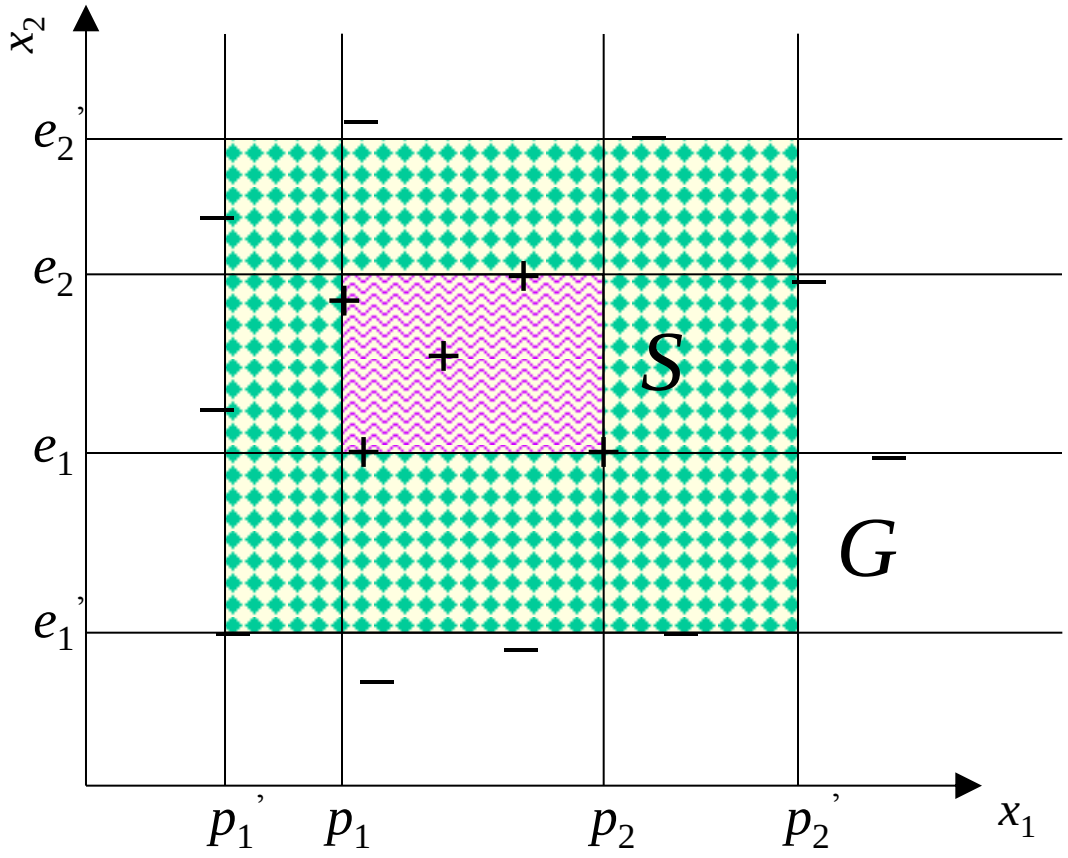
Choosing a Hypothesis

- Empirical Error: proportion of training instances where predictions of h do not match the training set

$$E(h|X) = \frac{1}{N}\sum_{t=1}^{N} 1(h(\mathbf{x}^t) \neq y^t)$$

- Each (p_1, p_2, e_1, e_2) defines a hypothesis $h \in \mathcal{H}$
- We need to find the best one…

Hypothesis Choice



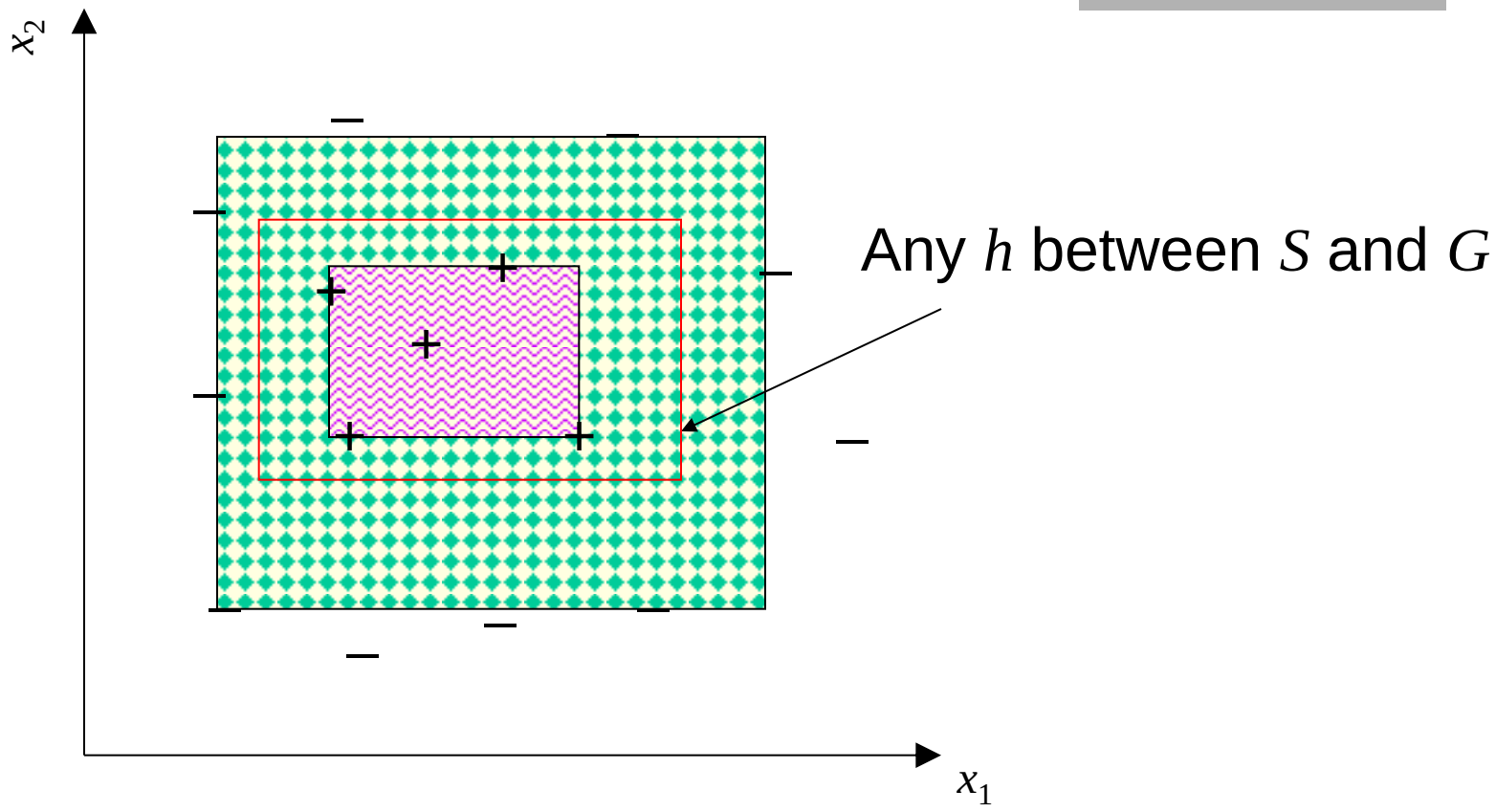
Most specific?

Most general?

Most specific hypothesis S

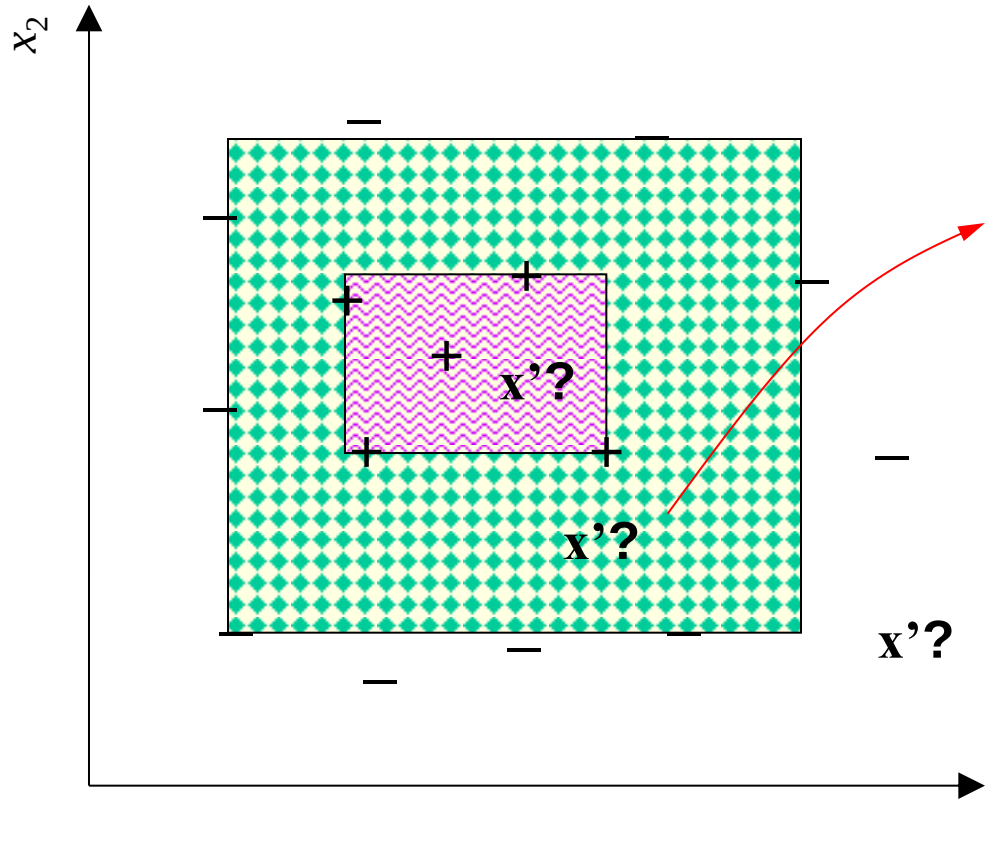
Most general hypothesis G

Consistent Hypothesis



G and S define the boundaries of the Version Space.
The set of hypotheses more general than S and more specific than G forms the **Version Space**, the set of consistent hypotheses

Now what?



- *Using the average of S and G or just rejecting it to experts?*

How do we make prediction for a new **x'**?

Binary Classification Problem

Learn a Classifier from the Training Set

Given a training dataset

$$S = \{(x^i, y_i) \big| x^i \in R^n, y_i \in \{-1, 1\}, i = 1, \ldots, m\}$$

$$x^i \in A_+ \Leftrightarrow y_i = 1 \;\; \& \quad x^i \in A_- \Leftrightarrow y_i = -1$$

Main goal: **Predict the unseen class label for new data**

(I) Find a function $f : R^n \rightarrow R$ by learning from data

$$f(x) > 0 \Rightarrow x \in A_+ \;\; and \quad f(x) < 0 \Rightarrow x \in A_-$$

(II) Estimate the *posteriori probability* of label

$$Pr(y = 1|x) > Pr(y = -1|x) \Rightarrow x \in A_+$$

Naïve Bayes for Classification Problem
Good for Binary as well as Multi-category

- Let each attribute be a random variable. What is the probability of the class given an instance?

$$Pr(Y = y | X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) = ?$$

- Naïve Bayes assumptions:
  - The importance of each attribute is equal
  - All attributes are *independent !*

$$Pr(Y = y | X_1 = x_1, X_2 = x_2, \ldots X_n = x_n)$$

$$= \frac{Pr(Y=y)}{Pr(X=x)} \prod_{j=1}^{n} Pr(X_j = x_j | Y = y)$$

The Weather Data Example

Ian H. Witten & Eibe Frank, Data Mining

Outlook	Temperature	Humidity	Windy	Play (Label)
Sunny	Hot	High	False	-1
Sunny	Hot	High	True	-1
Overcast	Hot	High	False	+1
Rainy	Mild	High	False	+1
Rainy	Cool	Normal	False	+1
Rainy	Cool	Normal	True	-1
Overcast	Cool	Normal	True	+1
Sunny	Mild	High	False	-1
Sunny	Cool	Normal	False	+1
Rainy	Mild	Normal	False	+1
Sunny	Mild	Normal	True	+1
Overcast	Mild	High	True	+1
Overcast	Hot	Normal	False	+1
Rainy	Mild	High	True	-1

Probabilities for the Weather Data

Using Frequencies to Approximate Probabilities

Outlook			Temp.			Humidity			Windy			Play	
Play	Yes	No		Yes	No		Yes	No		Yes	No	Yes	No
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	T	3/9	3/5		
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	F	6/9	2/5	9/14	5/14
Rainy	3/9	2/5	Cool	3/9	1/5								

$Pr(X_1 =' rainy' | Y = 1)$

$Pr(Y = 1)$

???

Likelihood of the two classes:

$$Pr(Y = 1 | sunny, cool, high, T) \propto \frac{2}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14}$$

$$Pr(Y = -1 | sunny, cool, high, T) \propto \frac{3}{5} \cdot \frac{1}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} \cdot \frac{5}{14}$$

The Zero-frequency Problem

- What if an attribute value does NOT occur with a class value?
  - The *posterior* probability will all be zero! No matter how likely the other attribute values are!
  - Laplace estimator will fix "zero-frequency" $\frac{k+\lambda}{n+a\lambda}$

Q: Roll a dice 8 times. The outcomes are as : 2, 5, 6, 2, 1, 5, 3, 6. What is the probability for showing 4.

$$P(X=4)=\frac{0+\lambda}{8+6\lambda},\quad P(X=5)=\frac{2+\lambda}{8+6\lambda}$$

Binary Classification Problem

Learn a Classifier from the Training Set

Given a training dataset

$$S = \{(x^i, y_i) \big| x^i \in R^n, y_i \in \{-1, 1\}, i = 1, \ldots, m\}$$

$$x^i \in A_+ \Leftrightarrow y_i = 1 \;\; \& \quad x^i \in A_- \Leftrightarrow y_i = -1$$

Main goal: Predict the unseen class label for new data

(I) Find a function $f : R^n \rightarrow R$ by learning from data

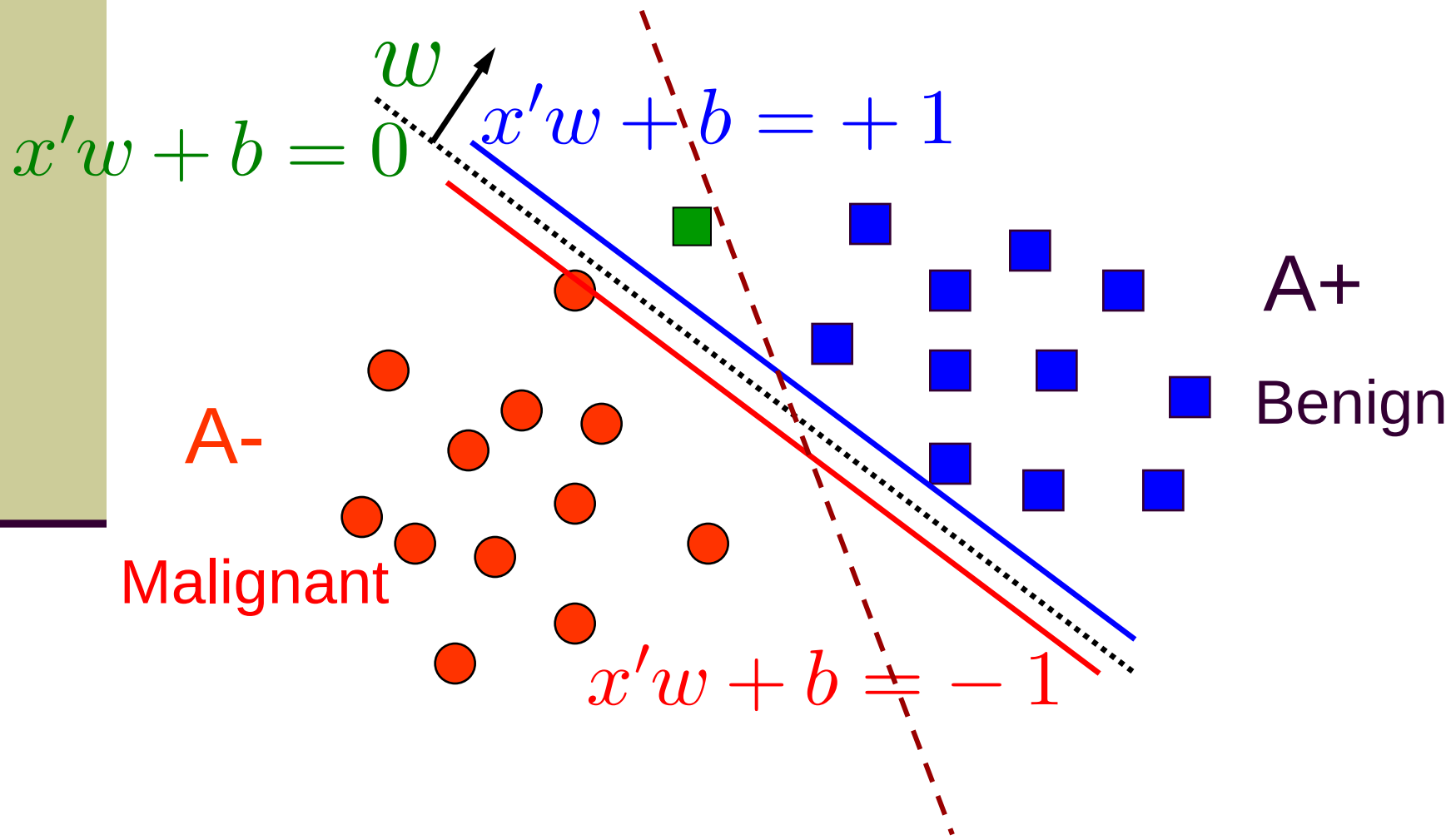
$$f(x) > 0 \Rightarrow x \in A_+ \;\; and \quad f(x) < 0 \Rightarrow x \in A_-$$

(II) Estimate the *posteriori probability* of label

$$Pr(y = 1|x) > Pr(y = -1|x) \Rightarrow x \in A_+$$

Binary Classification Problem

Linearly Separable Case



Linear Learning Machines

- Simplest case: the decision function is a hyperplane in input space.
- The Perceptron Algorithm: Rosenblatt, 1956
  - An *on-line* and *mistake-driven* procedure
  - Update the weight vector and *bias* when there is a *misclassified point*
  - Converge when problem is *linearly separable*

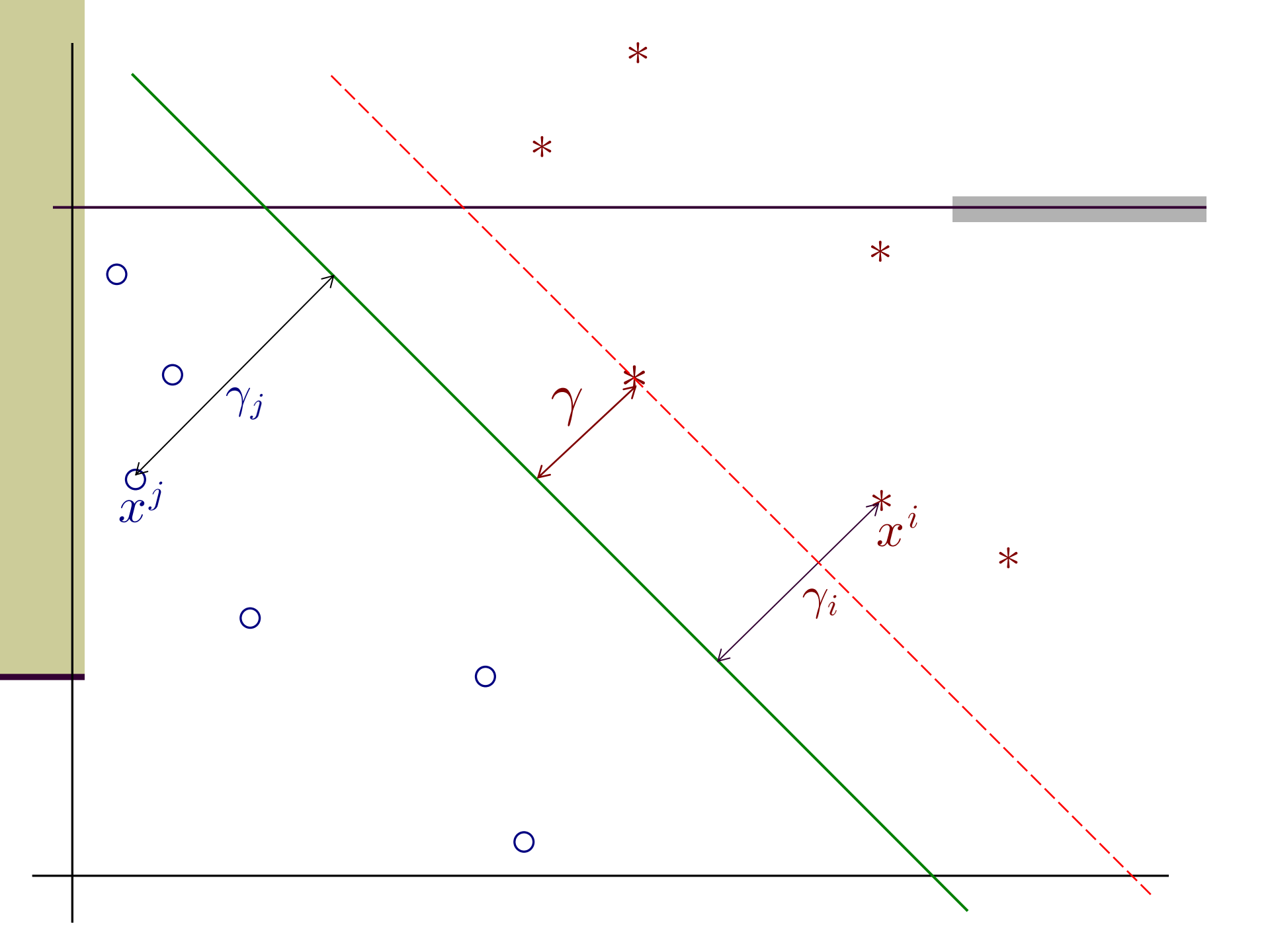
Basic Notations

Inner product: $x, w \in R^n, \ \langle x \cdot w \rangle = \sum_{i=1}^{n} x_i w_i$

Norm: 1-norm: $||x||_1 = \sum_{i=1}^{n} |x_i|$

2-norm: $||x||_2 = (\sum_{i=1}^{n} x_i^2)^{\frac{1}{2}}$

∞-norm: $||x||_\infty = \max_{1 \le i \le n} |x_i|$


γ_j
γ
x^j
x^i
γ_i

The Perceptron Algorithm
Rosenblatt, 1956

Given a linearly separable training set S and learning rate $\eta > 0$ and the initial weight vector, bias: $w^0 = 0, \quad b_0 = 0$

and let $R = \max_{1 \leq i \leq \ell} ||x^i||, \quad k = 0.$

The Perceptron Algorithm (Primal Form)

Repeat: $for\ \ i = 1\ \ to\ \ \ell$

$$if\ \ y_i(\langle w^k \cdot x^i \rangle + b_k) \leq 0\ \ then$$

$$w^{k+1} \leftarrow w^k + \eta y_i x^i$$

$$b_{k+1} \leftarrow b_k + \eta y_i R^2$$

$$k \leftarrow k + 1$$

$$end\ \ if$$

$$end\ \ for$$

until no mistakes made within the for loop return:

$k,\ \ (w^k, b_k)$. What is k ?

$$y_i\left(\left\langle w^{k+1} \cdot x^i \right\rangle + b_{k+1}\right) > y_i\left(\left\langle w^k \cdot x^i \right\rangle + b_k\right) \ ?$$

$$w^{k+1} \leftarrow w^k + \eta y_i x^i \ \ and \quad b_{k+1} \leftarrow b_k + \eta y_i R^2$$

$$y_i\left(\left\langle w^{k+1} \cdot x^i \right\rangle + b_{k+1}\right) = y_i\left(\left\langle (w^k + \eta y_i x^i) \cdot x^i \right\rangle + b_k + \eta y_i R^2\right)$$

$$= y_i\left(\left\langle w^k \cdot x^i \right\rangle + b_k\right) + y_i\left(\eta y_i(\left\langle x^i \cdot x^i \right\rangle + R^2)\right)$$

$$= y_i\left(\left\langle w^k \cdot x^i \right\rangle + b_k\right) + \eta(\left\langle x^i \cdot x^i \right\rangle + R^2)$$

The Perceptron Algorithm

(STOP in Finite Steps)

Theorem 2.3 (Novikoff)

Let S be a non-trivial training set, and let

$$R = \max_{1 \leq i \leq \ell} ||x^i||.$$

Suppose that there exists a vector w_{opt} $such$ $that$ $||w_{opt}|| = 1$ and $y_i(\langle w_{opt} \cdot x^i \rangle + b_{opt}) > \gamma$ for $1 \leq i \leq \ell$. Then the number of mistakes made by the on-line perceptron algorithm on S is at most $(\frac{2R}{\gamma})^2$.

The Perceptron Algorithm (Dual Form)

$$w = \sum_{i=1}^{\ell} \alpha_i y_i x^i$$

Given a linearly separable training set S and $\alpha = 0, \quad \alpha \in R^l$

$b = 0, R = \max_{1 \le i \le \ell} ||x_i||$

Repeat: $for \;\; i = 1 \;\; to \;\; \ell$

$if \;\; y_i(\sum_{j=1}^{l} \alpha_j y_j \langle x^j \cdot x^i \rangle + b) \leq 0 \;\; then$

$\alpha_i \leftarrow \alpha_i + 1; \quad b \leftarrow b + y_i R^2$

$end \;\; if$

$end \;\; for$

until no mistakes made within the for loop return: (α, b)

What We Got in the Dual Form Perceptron Algorithm?

- The number of updates equals: $\sum_{i=1}^{\ell} \alpha_i = ||\alpha||_1 \leq (\frac{2R}{\gamma})^2$
- $\alpha_i > 0$ implies that the training point (x^i, y_i) has been misclassified in the training process at least once.
- $\alpha_i = 0$ implies that removing the training point (x^i, y_i) will not affect the final results
- The training data only appear in the algorithm through the entries of the Gram matrix, $G \in R^{\ell \times \ell}$ which is defined below:

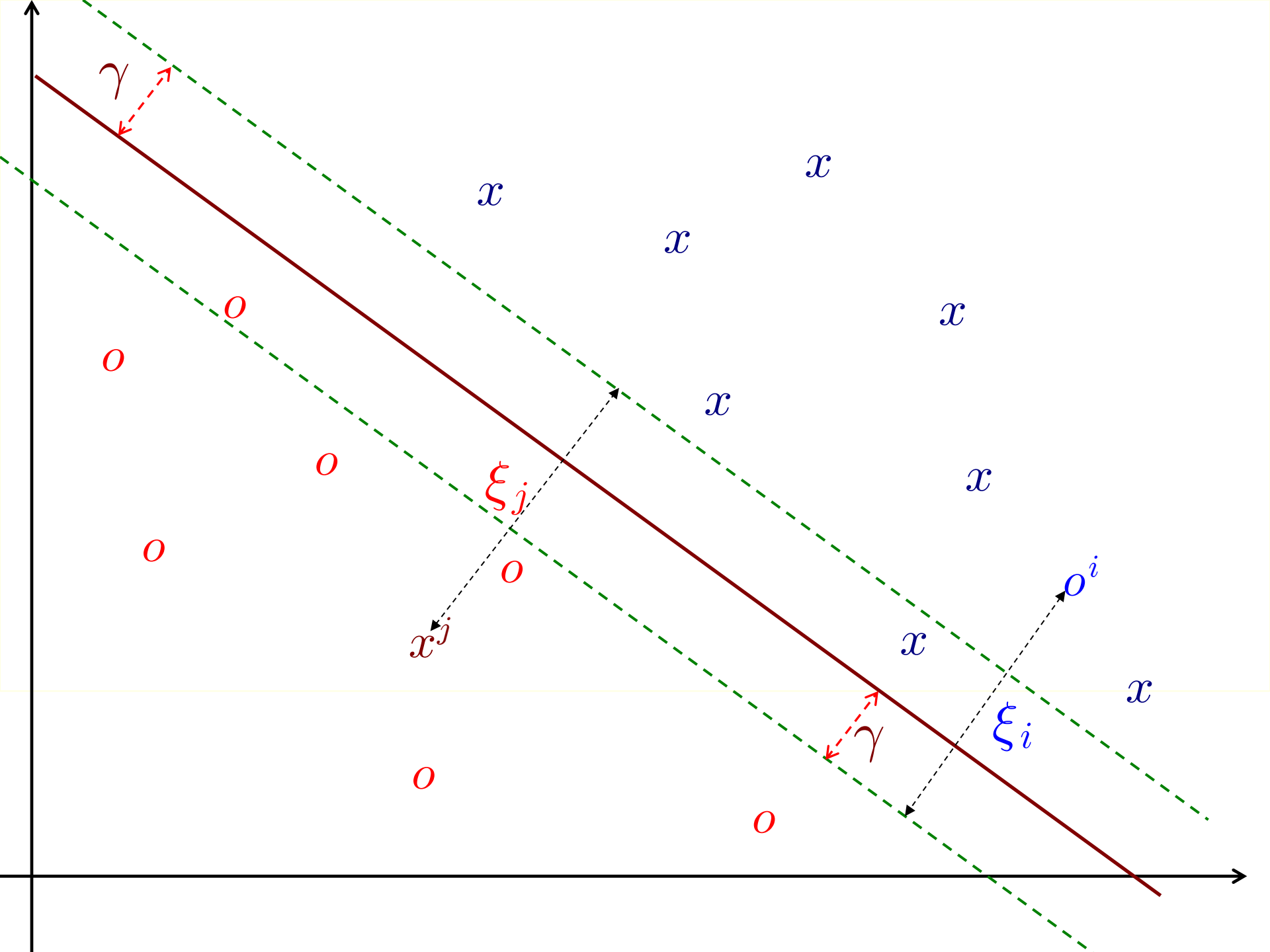
$$G_{ij} = \left\langle x^i, x^j \right\rangle$$

The Margin Slack Variable of (x^i, y_i) with respect to $\langle w, x \rangle + b = 0$

For a fixed value $\gamma > 0$ called the *target margin*, we define the *margin slack variable* of training point (x^i, y_i) with respect to the hyperplane $\langle w, x \rangle + b = 0$ and $\gamma > 0$ as

$$\xi_i = \max(0, \ \gamma - y_i(\langle w, x^i \rangle + b)).$$

- If $\xi_i > \gamma \Leftrightarrow y_i(\langle w, x^i \rangle + b) < 0$ then x^i is *misclassified* by the hyperplane $\langle w, x \rangle + b = 0$


γ
x
x
x
x
x
x
x
o
o
o
o
o
o
o
ξ_j
x^j
o^i
ξ_i
γ

Bound of Mistakes of a *for loop* for the Perceptron Algorithm

Theorem 2.7 (Freund & Schapir)

Let S be a non-trivial training set with no duplicate examples, with $\|x^i\| \leq R$. Let (w, b) be any hyperplane with $\|w\| = 1$, and $\gamma > 0$ and define

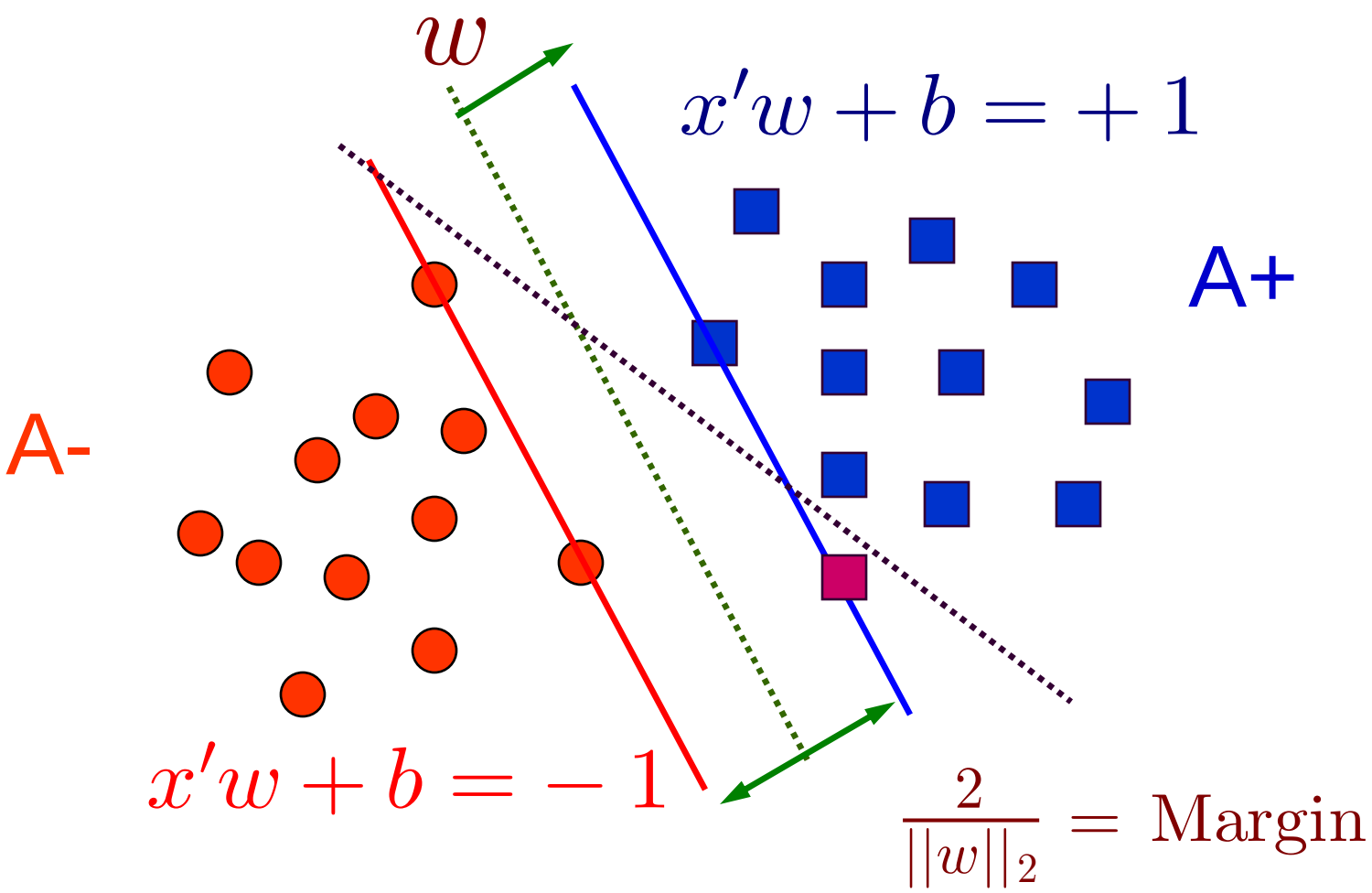
$$D = \sqrt{\sum_{i=1}^{\ell} \xi_i^2}, \quad \xi_i = \max(0, \ \gamma - y_i(\langle w \cdot x \rangle + b)).$$

Then the number of mistakes in the first execution of the *for loop* of the Perceptron Alg. on S is bounded by

$$\left(\frac{2(R+D)}{\gamma}\right)^2.$$

Support Vector Machines

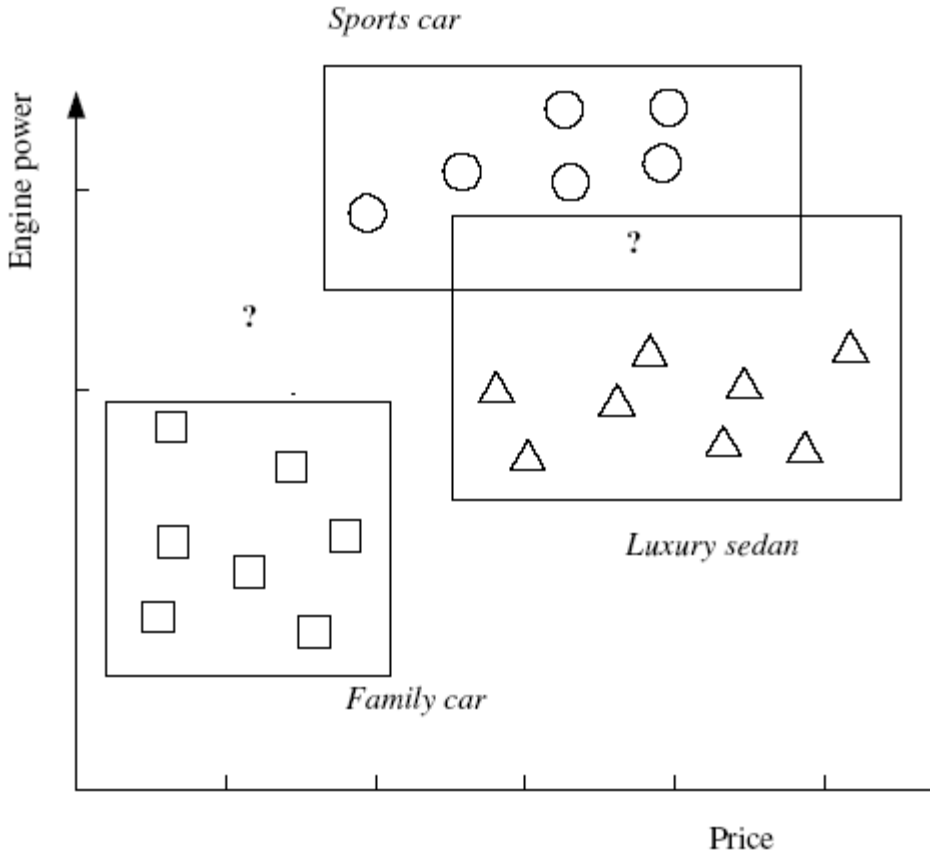
Maximizing the Margin between Bounding Planes



Why We Maximize the Margin? (Based on Statistical Learning Theory)

- The Structural Risk Minimization (SRM):
  - The expected risk will be less than or equal to empirical risk (training error)+ VC (error) bound
- $\|w\|_2 \propto VC\ \ bound$
- $\min VC\ \ bound \Leftrightarrow \min \frac{1}{2}\|w\|_2^2 \Leftrightarrow \max Margin$

Learning Multiple Classes



- K-class classification
- $\Rightarrow$ K two-class problems (one against all)
- $\Rightarrow$ could introduce *doubt*
- $\Rightarrow$ could have unbalance data

Regression

- Supervised learning where the output is not a classification (e.g. 0/1, true/false, yes/no), but the output is a real number.
  - $\mathcal{X} = \{\mathbf{x}^t, y^t\}_{t=1}^{N}, y^t \in \mathbf{R}$

Regression

- Suppose that the true function is
  $y^t = f(\mathbf{x}^t) + \varepsilon$ where ε is random noise
- Suppose that we learn $g(x)$ as our model. The empirical error on the training set is

$$\frac{1}{N}\sum_{t=1}^{N} L(y^t, g(\mathbf{x}^t))$$

⇒ Because y^t and $g(\mathbf{x}^t)$ are numeric, it makes sense for L to be the distance between them.

⇒ Common distance measures:

- mean squared error

$$\frac{1}{N}\sum_{t=1}^{N} (y^t - g(\mathbf{x}^t))^2$$

- absolute value of difference
- etc.

Example: Linear Regression

- Assume $g(x)$ is linear

$$g(x) = w_1 x_1 + \cdots + w_d x_d + w_0 = \sum_{i=1}^{d} w_i x_i + w_0$$

and we want to minimize the mean squared error

$$\frac{1}{N} \sum_{t=1}^{N} (y^t - g(x^t))^2$$

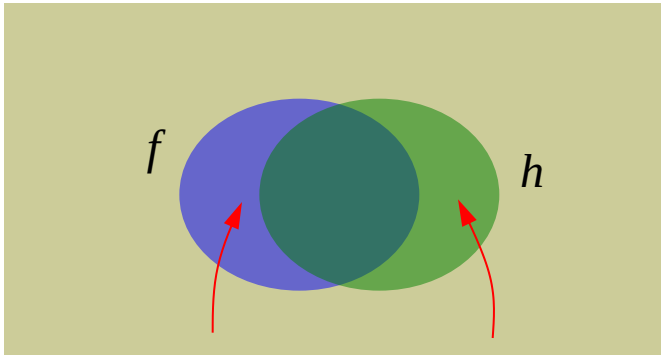
- We can solve this for the w_i that minimizes the error

Issues

- Hypothesis space must be flexible enough to represent concept
- Making sure that the gap of S and G sets do not get too large
- Assumes no noise!
  - inconsistently labeled examples will cause the version space to **collapse**
  - there have been extensions to handle this…

Why We can Learn from Data in the Noise-Free Case

- **Assumption:** Examples are generated according to a probability distribution $p(\mathbf{x})$ and labeled according to an unknown function $y = f(\mathbf{x})$
- **Learning Algorithm:** The learning algorithm is given a set of N examples, and it outputs a hypothesis $h \in \mathcal{H}$ that is **consistent** with those examples (correctly labels all of them).
- **Goal:** h should have a low error rate on new examples from the same distribution $p(\mathbf{x})$.



$$\text{error}(h, f) = p[f(\mathbf{x}) \neq h(\mathbf{x})]$$

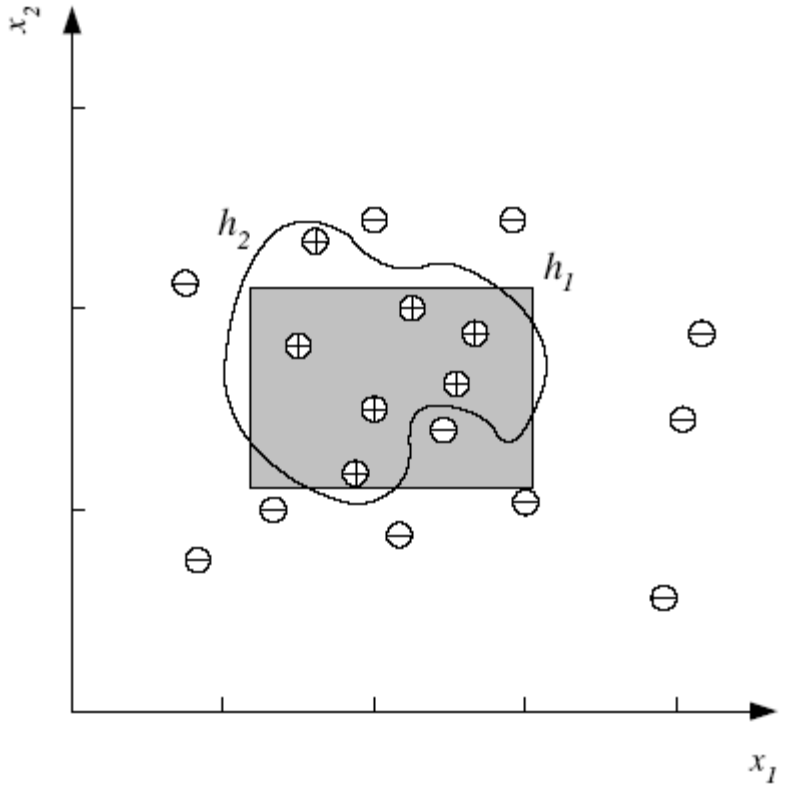
Signal or Noise

- Noise: unwanted anomaly in the data
- Another reason we can't always have a perfect hypothesis
  - error in sensor readings for input
  - teacher noise: error in labeling the data
  - additional attributes which we have not taken into account. These are called **hidden** or **latent** because they are unobserved.
- The Signal and the Noise

  Why So Many Predictions Fail, but Some Don't?

  Nate Silver

When there is noise…



- There may not have a simple boundary between the positive and negative instances
- Zero (training) misclassification error may not be possible

Probably Approximately Correct Learning
pac Model

- Key assumption:

  Training and testing data are generated *i.i.d.* according to a *fixed but unknown* distribution D

- Evaluate the "quality" of a hypothesis (classifier) $h \in H$ should take the *unknown* distribution D into account (*i.e.* "average error" or "expected error" made by the $h \in H$)

- We call such measure risk functional and denote it as $err_{D}(h) = D\{(x, y) \in X \times \{1, -1\} | \ h(x) \neq y\}$

Generalization Error of pac Model

- Let $S = \{(x^1, y_1), \ldots, (x^l, y_l)\}$ be a set of l training examples chosen i.i.d. according to D
- Treat the generalization error $err_D(h_S)$ as a *r.v.* depending on the random selection of S
- Find a bound of the trail of the distribution of *r.v.* $err_D(h_S)$ in the form $\varepsilon = \varepsilon(l, H, \delta)$
- $\varepsilon = \varepsilon(l, H, \delta)$ is a function of l, H and δ,where $1 - \delta$ is the confidence level of the error bound which is given by learner

Probably Approximately Correct

- We assert:

$$Pr(\{ err_{D}(h_S) > \varepsilon = \varepsilon(l, H, \delta)\}) < \delta$$

or

$$Pr(\{ err_{D}(h_S) \leq \varepsilon = \varepsilon(l, H, \delta)\}) \geq 1 - \delta$$

- The error made by the hypothesis h_s will be less then the error bound $\varepsilon(l, H, \delta)$ that is not depend on the unknown distribution D

PAC *vs.* Poll (民意調查)

- There are 1265 samples were drawn via simple random sampling. The error is less than $\pm 2.76\%$ with 95% confident level.

$$Pr(\{ err_{D}(h_S) \leq \varepsilon = \varepsilon(l, H, \delta)\}) \geq 1 - \delta$$

$$l = 1265, \;\; \varepsilon(l, H, \delta) = 0.0276, \;\; \delta = 0.05$$

Find the Hypothesis with Minimum Expected Risk?

- Let $S = \{(x^1, y_1), \ldots, (x^l, y_l)\} \subseteq X \times \{-1, 1\}$ be the training examples chosen i.i.d. according to D with the probability density $p(x, y)$
- The expected misclassification error made by $h \in H$ is

$$R[h] = \int_{X \times \{-1,1\}} \frac{1}{2} |h(x) - y| dp(x, y)$$

- The *ideal* hypothesis h^*_{opt} should has the smallest expected risk $R[h^*_{opt}] \leq R[h], \quad \forall h \in H$

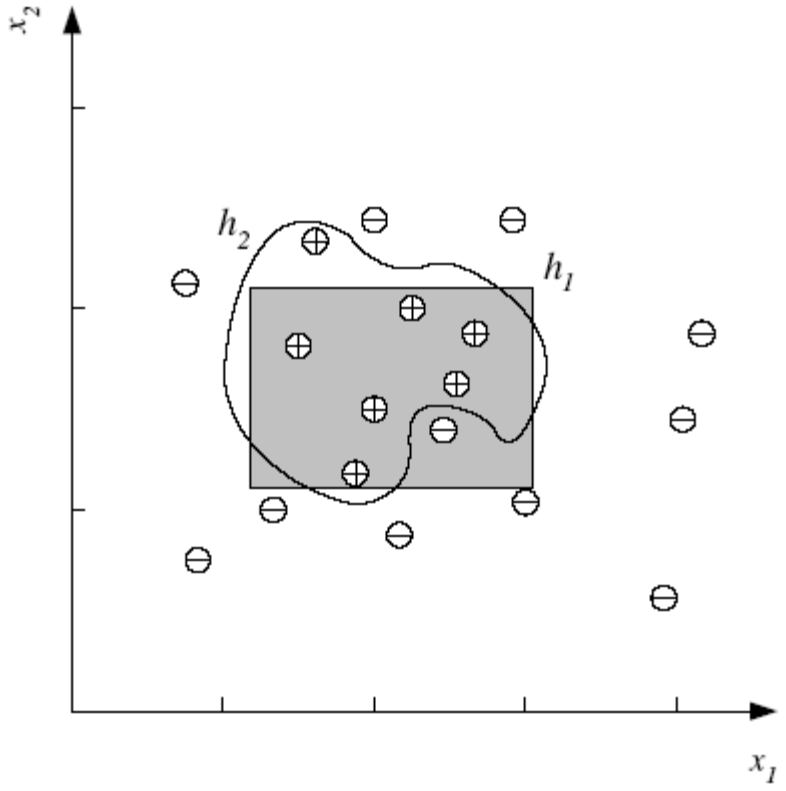
Unrealistic !!!

Empirical Risk Minimization (ERM)

(D and $p(x, y)$ are not needed)

- Replace the expected risk over $p(x, y)$ by an average over the training example
- The empirical risk: $R_{emp}[h] = \frac{1}{l} \sum_{i=1}^{l} \frac{1}{2} |h(x^i) - y_i|$
- Find the hypothesis h^*_{emp} with the smallest empirical risk $R_{emp}[h^*_{emp}] \leq R_{emp}[h], \quad \forall h \in H$
- Only focusing on empirical risk will cause *overfitting*

When there is noise…



- There may not have a simple boundary between the positive and negative instances
- Zero (training) misclassification error may not be possible

VC Confidence (Vapnik and Chervonenkis)

(The Bound between $R_{emp}[h]$ & $R[h]$)

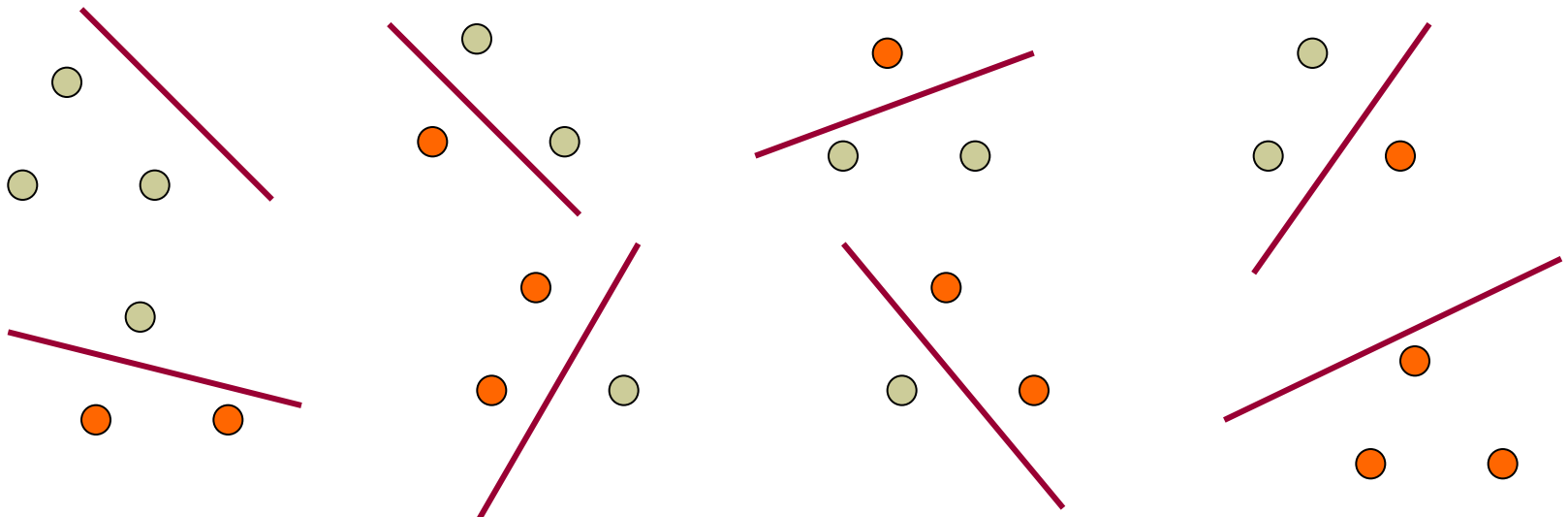
- The following inequality will be held with probability $1-\delta$

$$R[h] \leq R_{emp}[h] + \sqrt{\frac{v(log(2l/v)+1)-log(\delta/4)}{l}}$$

C. J. C. Burges, *A tutorial on support vector machines for pattern recognition*,
Data Mining and Knowledge Discovery 2 (2) (1998), p.121-167

Capacity (Complexity) of Hypothesis Space H :VC-dimension

- A given training set S is *shattered* by H if and only if for every labeling of S, $\exists\ h \in H$ consistent with this labeling
- Three (linear independent) points shattered by a hyperplanes in R^2



Shattering Points with Hyperplanes in R^n

Can you always shatter three points with a line in R^2?

Theorem: Consider some set of m points in R^n. Choose a point as origin. Then the m points can be shattered by oriented hyperplanes if and only if the position vectors of the rest points are linearly independent.

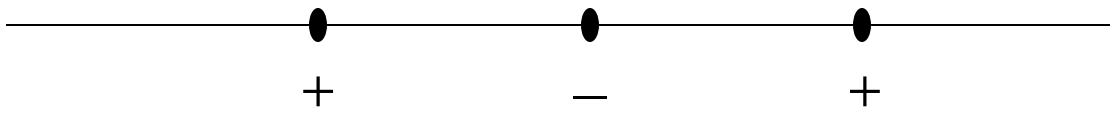
Definition of VC-dimension

(A Capacity Measure of Hypothesis Space H)

- The *Vapnik-Chervonenkis* dimension, $VC(H)$, of hypothesis space H defined over the input space X is the size of the (existent) largest finite subset of X shattered by H
- If arbitrary large finite set of X can be shattered by H, then $VC(H) \equiv \infty$
- Let $H = \{all \ \ hyperplanes \ \ in \ \ R^n\}$ then $VC(H) = n + 1$

Example I

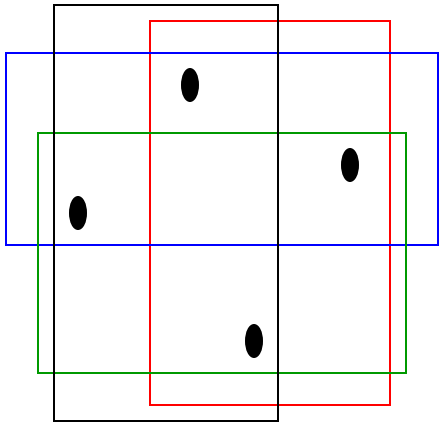
- $x \in \mathbf{R}$, $\mathcal{H}$ = interval on line
  - There exists two points that can be shattered
  - No set of three points can be shattered
  - $\mathrm{VC}(\mathcal{H}) = 2$



  - An example of three points (and a labeling) that cannot be shattered

Example II

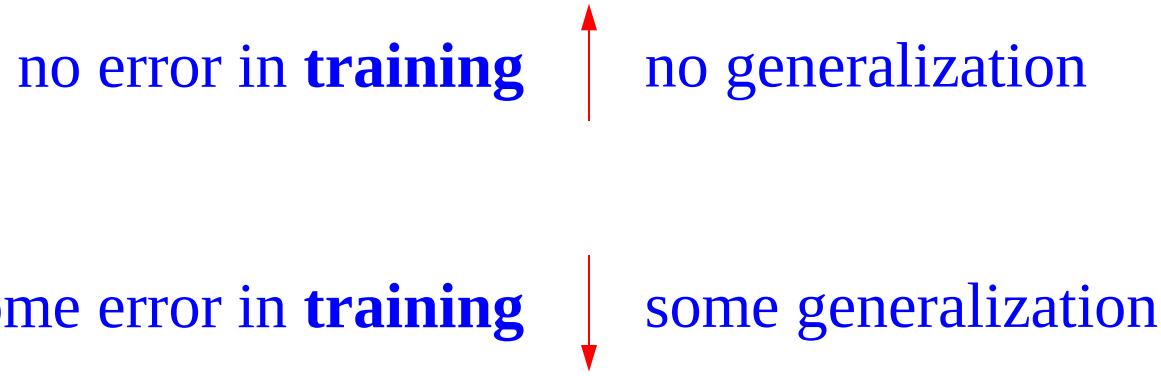
- $\mathbf{x} \in \mathbf{R} \times \mathbf{R}$, $\mathcal{H}$ = Axis parallel rectangles
  - There exist four points that can be shattered
  - No set of five points can be shattered
  - $\mathrm{VC}(\mathcal{H}) = 4$



- Hypotheses consistent with all ways of labeling three positive;
- Check that there hypothesis for all ways of labeling one, two or four points positive

Example III

- A lookup table has infinite VC dimension!



- A hypothesis space with low VC dimension

Comments

- VC dimension is distribution-free; it is independent of the probability distribution from which the instances are drawn
- In this sense, it gives us a worse case complexity (pessimistic)
  - In real life, the world is smoothly changing, instances close by most of the time have the same labels, no worry about *all possible labelings*
- However, this is still useful for providing bounds, such as the sample complexity of a hypothesis class.
- In general, we will see that there is a connection between the VC dimension (which we would like to minimize) and the error on the training set (empirical risk)

Something about Simple Models

- Easier to classify a new instance
- Easier to explain
- Fewer parameters, means it is easier to train. The sample complexity is lower.
- Lower variance. A small change in the training samples will not result in a wildly different hypothesis
- High bias. A simple model makes strong assumptions about the domain; great if we're right, a disaster if we are wrong.

  optimality?: *min* (variance + bias)
- May have better generalization performance, especially if there is noise.
- Occam's razor: simpler explanations are more plausible

Underfitting and Overfitting

- Matching the complexity of the hypothesis with the complexity of the target function
  - if the hypothesis is less complex than the function, we have **underfitting**. In this case, if we increase the complexity of the model, we will reduce both training error and validation error.
  - if the hypothesis is too complex, we may have **overfitting**. In this case, the validation error may go up even the training error goes down. For example, we fit the noise, rather than the target function.

Tradeoffs

- (Dietterich 2003)
- complexity/capacity of the hypothesis
- amount of training data
- generalization error on new examples

Take Home Remarks

- What is the hardest part of machine learning?
  - selecting attributes (representation)
  - deciding the hypothesis (assumption) space: big one or small one, that's the question!
- Training is relatively easy
  - DT, NN, SVM, (KNN), …
- The usual way of learning in real life

$\Rightarrow$ not supervised, not unsupervised, but semi-supervised, even with some taste of reinforcement learning

Take Home Remarks

- Learning == Search in Hypothesis Space
- Inductive Learning Hypothesis: Generalization is possible.
- If a machine performs well on most training data AND it is not too complex, it will probably do well on similar test data.
- Amazing fact: in many cases this can actually be proven. In other words, if our hypothesis space is not too complicated/flexible (has a low capacity in some formal sense), and if our training set is large enough then we can bound the probability of performing much worse on test data than on training data.
- The above statement is carefully formalized in 40 years of research in the area of learning theory.