Supervised Learning
Lecture 4: Advanced Approaches

Tho Quan
qttho@hcmut.edu.vn

Agenda

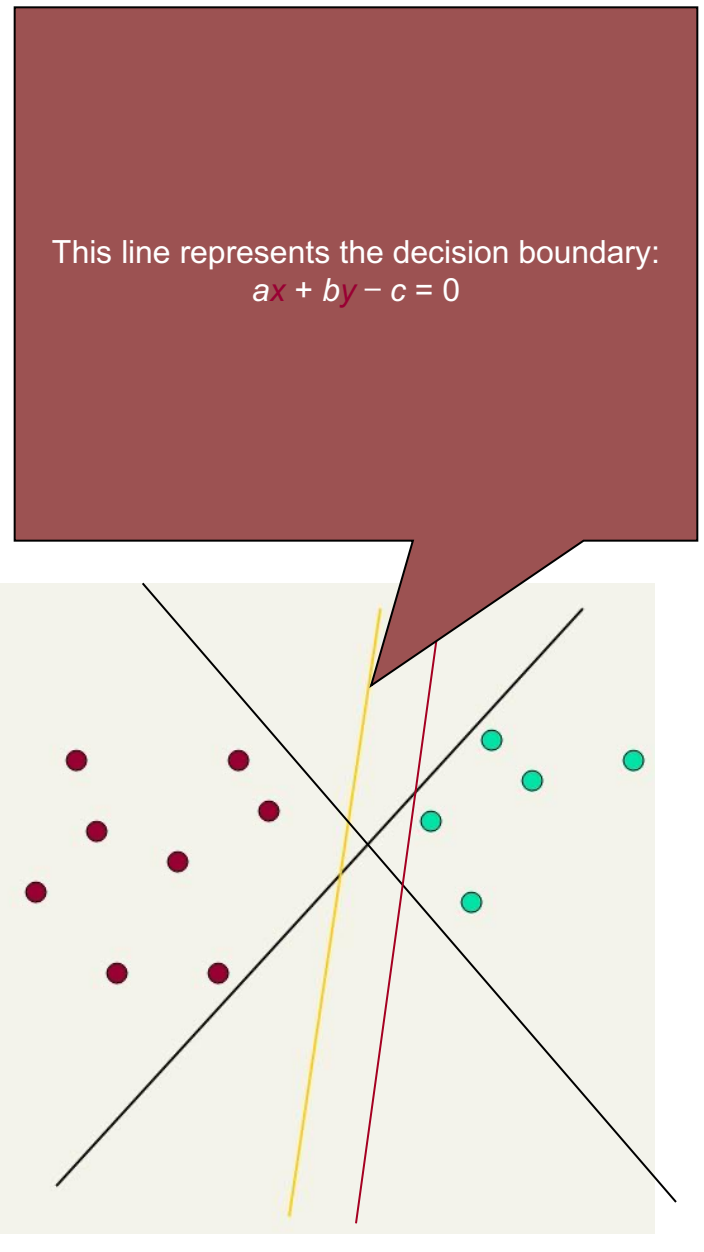
- SVM and Kernel Method
- Perceptron and ANN
- Feature Selection

Text classification: Up until now and today

- Previously: 3 algorithms for text classification
  - Vector space classification using centroids and hyperplanes that split them
    - Simple, linear discriminant classifier; perhaps too simple
      - (or maybe not*)
    - K Nearest Neighbor classification
      - Simple, expensive at test time, high variance, non-linear
  - Naive Bayes classifier
- Today
  - SVMs
  - Some empirical evaluation and comparison

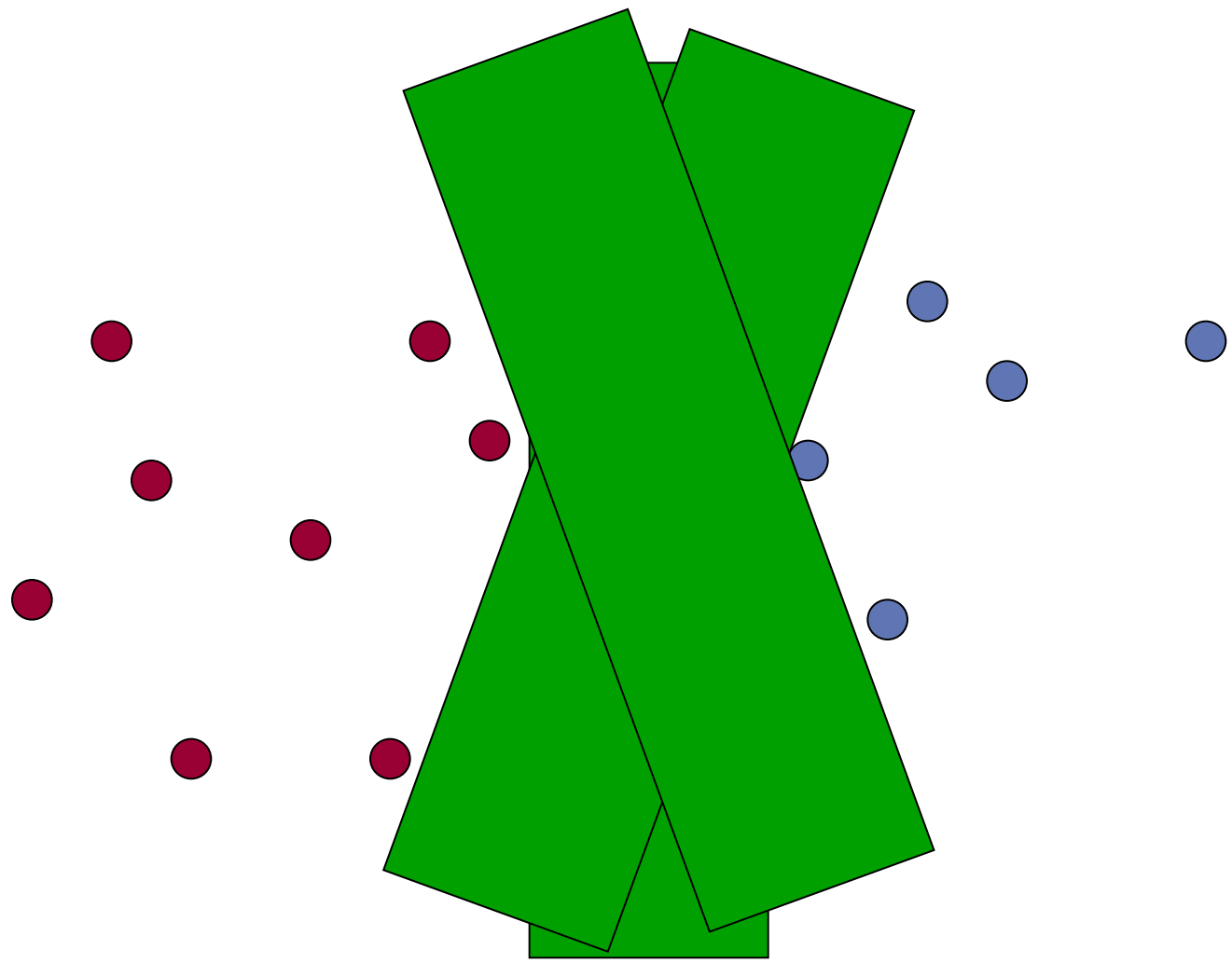
Linear classifiers: Which Hyperplane?

- Lots of possible solutions for *a, b, c.*
- Some methods find a separating hyperplane, but not the optimal one [according to some criterion of expected goodness]
  - E.g., perceptron
- Support Vector Machine (SVM) finds an optimal* solution.
  - Maximizes the distance between the hyperplane and the "difficult points" close to decision boundary
  - One intuition: if there are no points near the decision surface, then there are no very uncertain classification decisions

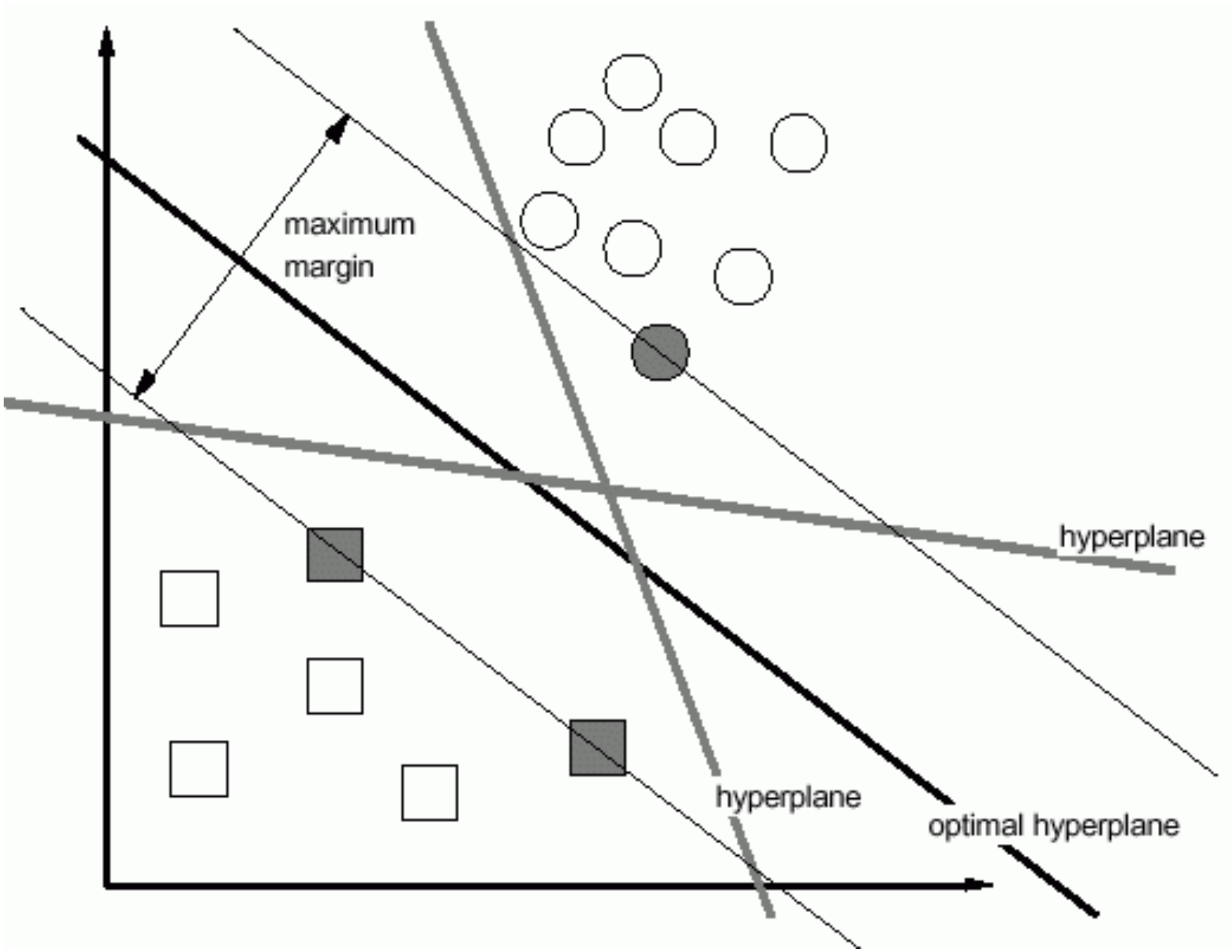


Another intuition

- If you have to place a fat separator between classes, you have less choices, and so the capacity of the model has been decreased

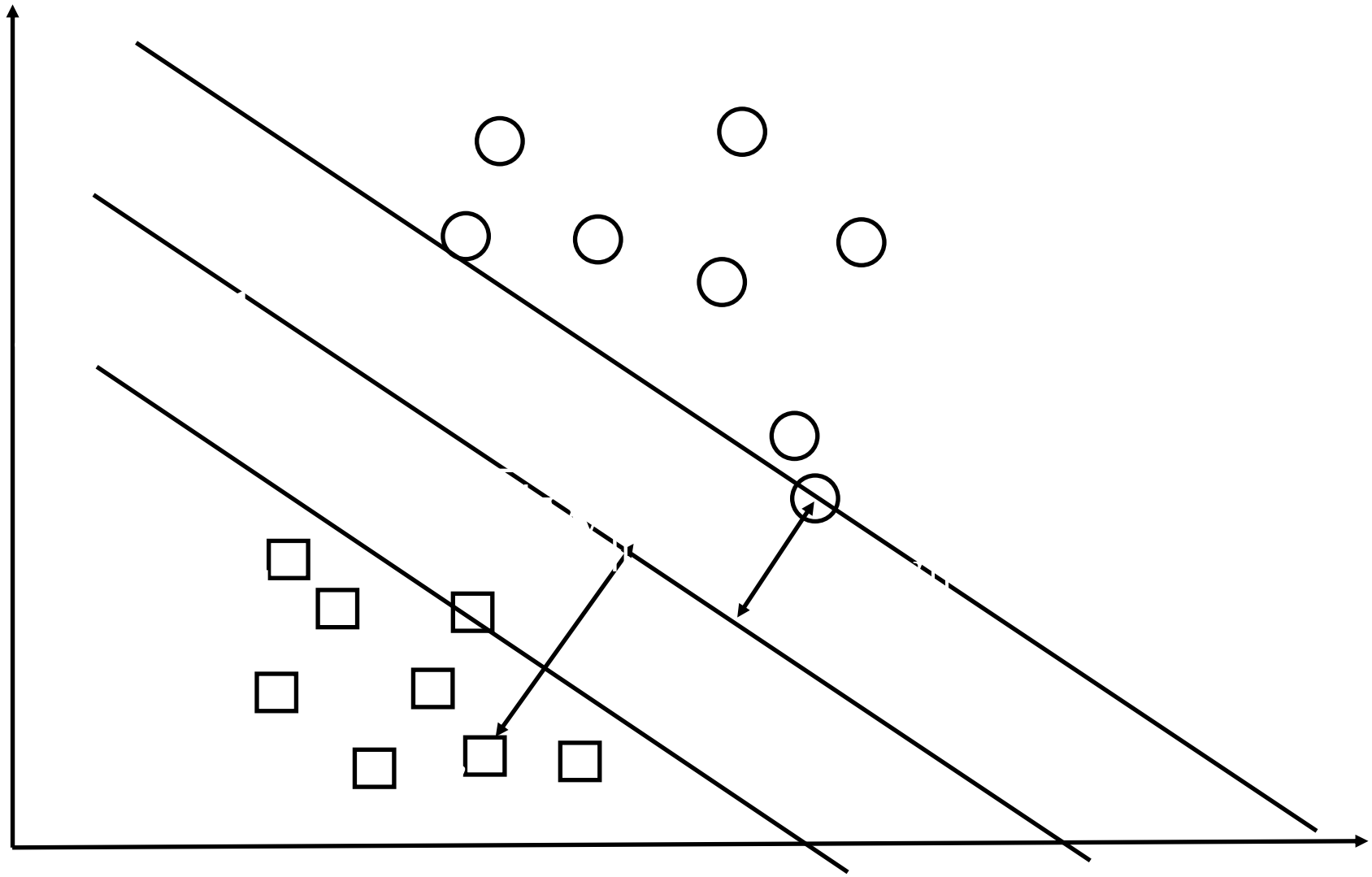


Maximization of margin



Among all discriminating hyperplanes there is one that is clearly better.

Maximization of margin



$g(\mathbf{X})=\mathbf{W}^{\mathrm{T}}\mathbf{X}+W_0$ is the discriminant function, $g(\mathbf{X})/\|\mathbf{W}\|$ is the distance and the best discriminating hyperplane should maximize the distance between the $g(\mathbf{X})=0$ plane and the data samples.

Linear Support Vector Machine (SVM)

- Hyperplane

  $\mathbf{w}^T \mathbf{x} + b = 0$

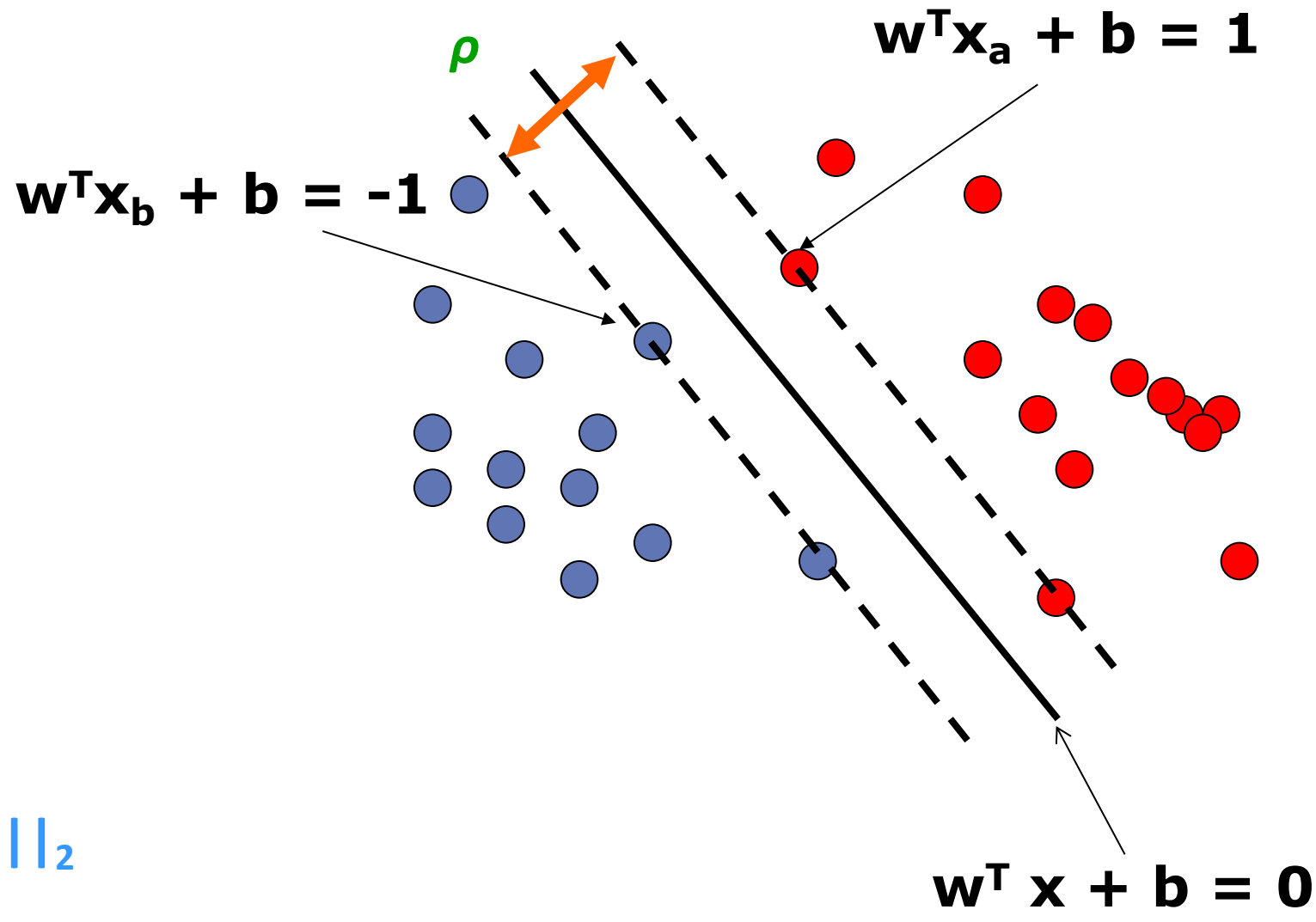
- Extra scale constraint:

  $\min_{i=1,\ldots,n} |\mathbf{w}^T \mathbf{x}_i + b| = 1$

- This implies:

  $\mathbf{w}^T(\mathbf{x}_a - \mathbf{x}_b) = 2$

  $\rho = ||\mathbf{x}_a - \mathbf{x}_b||_2 = 2/||\mathbf{w}||_2$



Solving the Optimization Problem

Find $\mathbf{w}$ and b such that
$\mathbf{\Phi(w)} = \frac{1}{2}\,\mathbf{w}^T\mathbf{w}$ is minimized;
and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i\,(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

- This is now optimizing a *quadratic* function subject to *linear* constraints
- Quadratic optimization problems are a well-known class of mathematical programming problem, and many (intricate) algorithms exist for solving them (with many special ones built for SVMs)
- The solution involves constructing a *dual problem* where a *Lagrange multiplier* α_i is associated with every constraint in the primary problem:

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q(\alpha)} = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

The Optimization Problem Solution

- The solution has the form:

$$\mathbf{w} = \Sigma \alpha_i y_i \mathbf{x_i} \qquad b = y_k - \mathbf{w}^T \mathbf{x_k} \text{ for any } \mathbf{x_k} \text{ such that } \alpha_k \neq 0$$

- Each non-zero α_i indicates that corresponding $\mathbf{x_i}$ is a support vector.
- Then the classifying function will have the form:

$$f(\mathbf{x}) = \Sigma \alpha_i y_i \mathbf{x_i}^T \mathbf{x} + b$$

- Notice that it relies on an *inner product* between the test point **x** and the support vectors $\mathbf{x_i}$
  - We will return to this later.

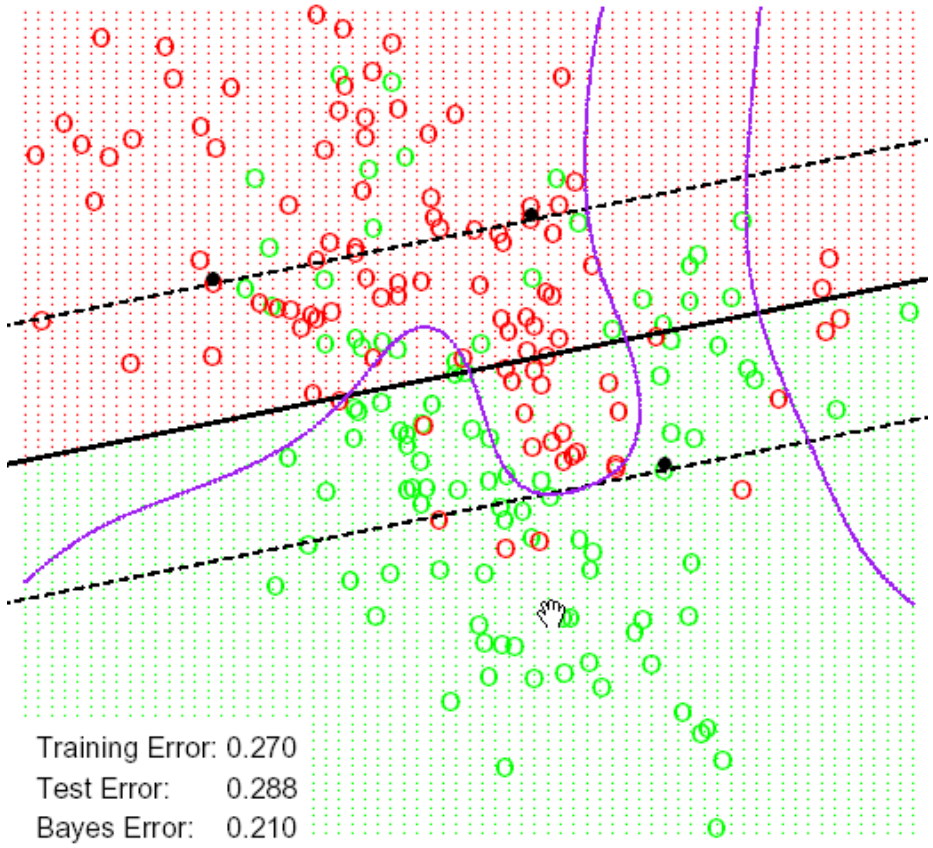
Linear SVMs: Summary

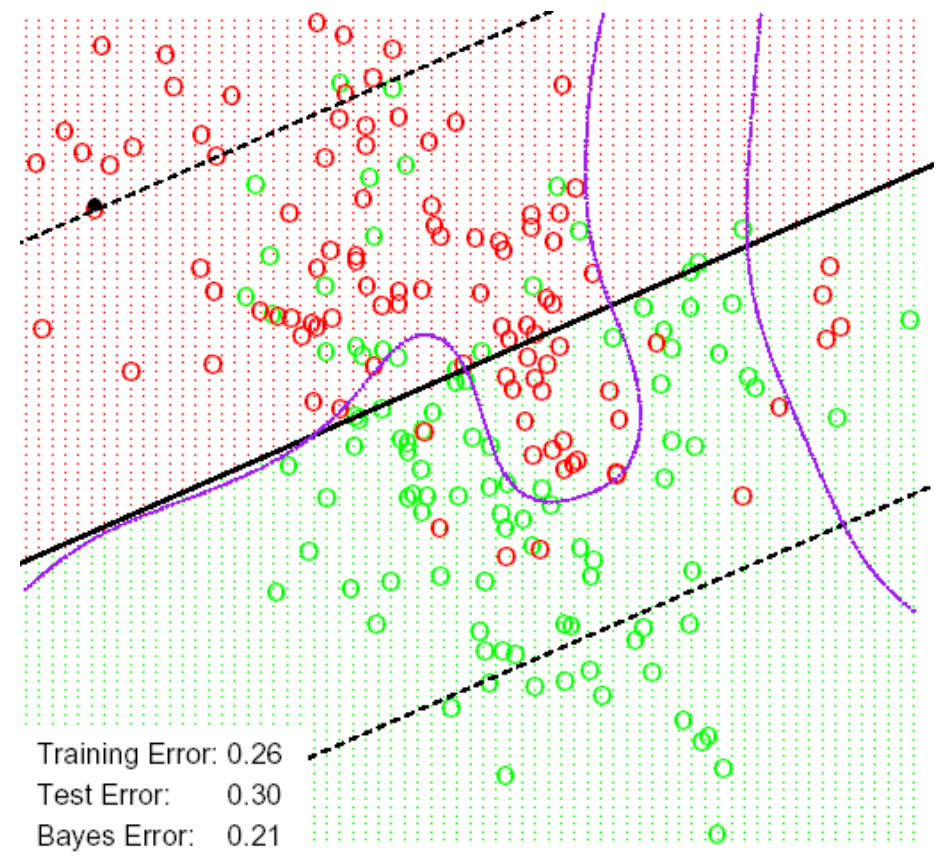
- The classifier is a *separating hyperplane.*
- The most "important" training points are the support vectors; they define the hyperplane.
- Quadratic optimization algorithms can identify which training points $\mathbf{x}_i$ are support vectors with non-zero Lagrangian multipliers α_i.
- Both in the dual formulation of the problem and in the solution, training points appear only inside inner products:

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \le \alpha_i \le C$ for all α_i

$$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x}_i^T\mathbf{x} + b$$

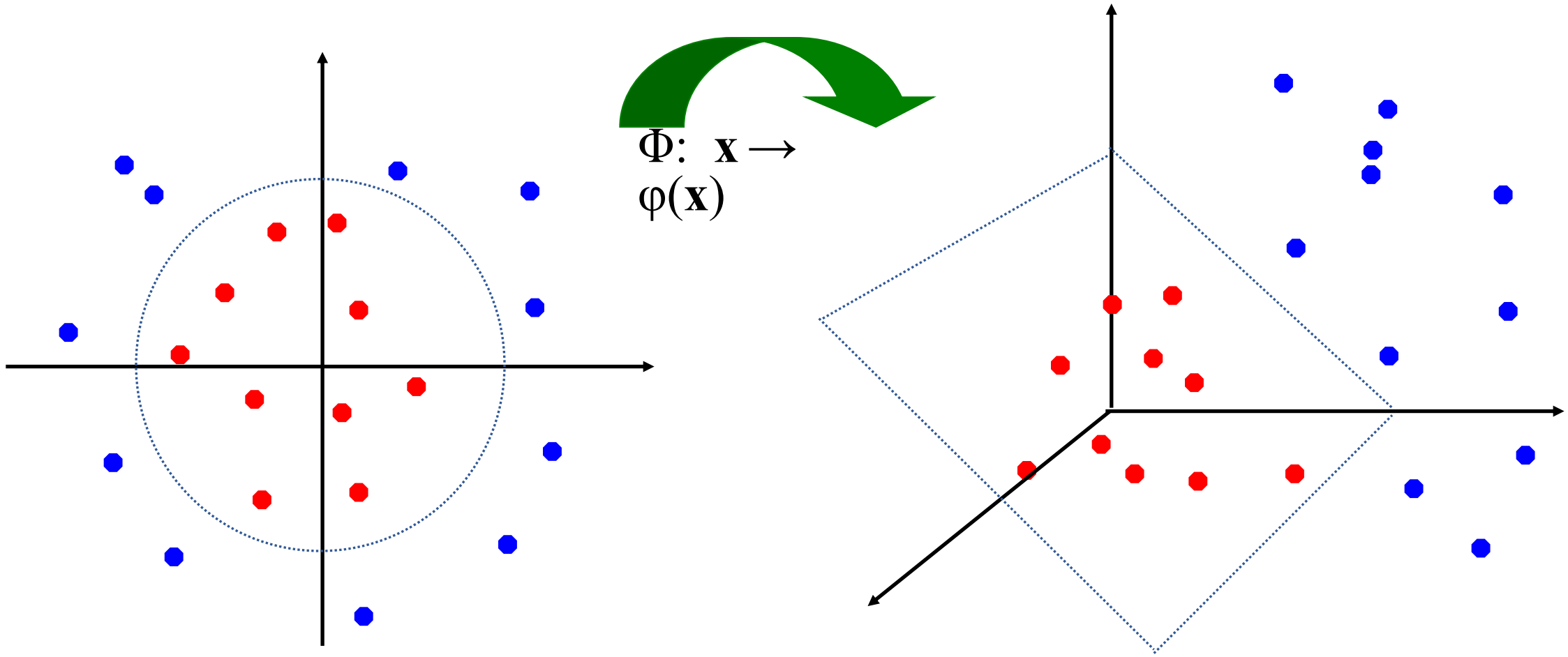
Non-linear separation





Non-linear SVMs: Feature spaces

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



The "Kernel Trick"

- The linear classifier relies on an inner product between vectors $K(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i^T\mathbf{x}_j$
- If every datapoint is mapped into high-dimensional space via some transformation Φ: $\mathbf{x} \rightarrow \phi(\mathbf{x})$, the inner product becomes:

$$K(\mathbf{x}_i,\mathbf{x}_j)= \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$$

- A *kernel function* is some function that corresponds to an inner product in some expanded feature space.
- Example:

  2-dimensional vectors $\mathbf{x}=[x_1 \;\; x_2]$; let $K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2$,

  Need to show that $K(\mathbf{x}_i,\mathbf{x}_j)= \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$:

  $K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2 = 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2} + x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} =$

  $= [1 \;\; x_{i1}^2 \;\; \sqrt{2}\, x_{i1}x_{i2} \;\; x_{i2}^2 \;\; \sqrt{2}x_{i1} \;\; \sqrt{2}x_{i2}]^T [1 \;\; x_{j1}^2 \;\; \sqrt{2}\, x_{j1}x_{j2} \;\; x_{j2}^2 \;\; \sqrt{2}x_{j1} \;\; \sqrt{2}x_{j2}]$

  $= \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ where $\phi(\mathbf{x}) = [1 \;\; x_1^2 \;\; \sqrt{2}\, x_1x_2 \;\; x_2^2 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2]$

Some popular kernels

Polynomial: $K_p(\mathbf{X},\mathbf{Y}) = (1+\mathbf{X}\cdot\mathbf{Y})^{\kappa}$

Gaussian: $K_G(\mathbf{X},\mathbf{Y}) = \exp\left(-\|\mathbf{X}-\mathbf{Y}\|^2/2\sigma^2\right)$

Sigmoidal: $K_s(\mathbf{X},\mathbf{Y}) = \tanh\left(\kappa_1\mathbf{X}\cdot\mathbf{Y}+\kappa_2\right)$

Distance: $K_d(\mathbf{X},\mathbf{Y}) = \|\mathbf{X}-\mathbf{Y}\|^b$

$$\mathbf{X}^{(i)}\cdot\mathbf{X}^{(j)} \rightarrow \Phi\left(\mathbf{X}^{(i)}\right)\cdot\Phi\left(\mathbf{X}^{(j)}\right) = K\left(\mathbf{X}^{(i)},\mathbf{X}^{(j)}\right)$$

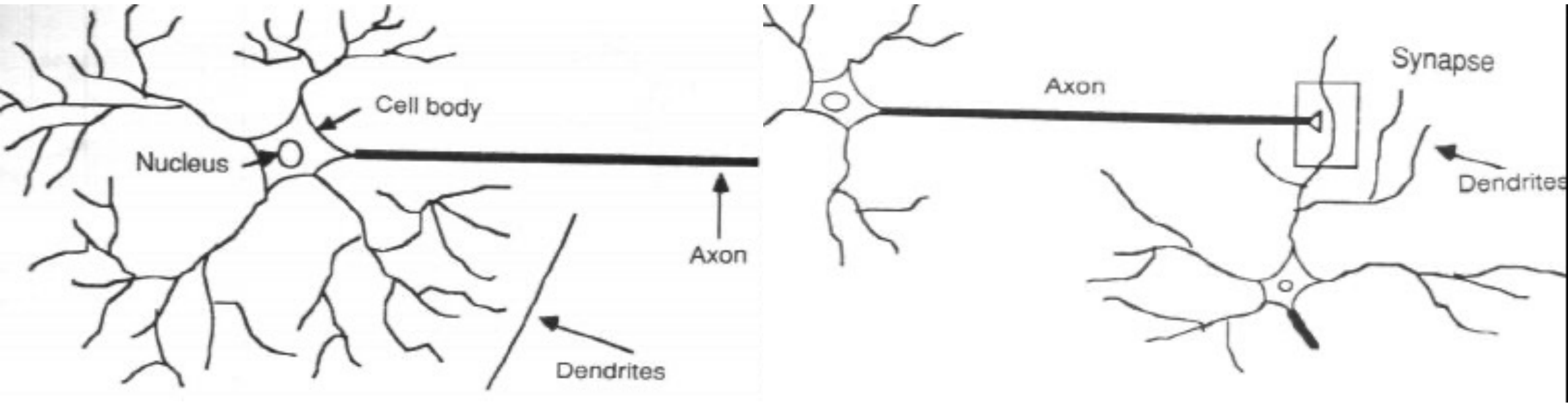
Some applications

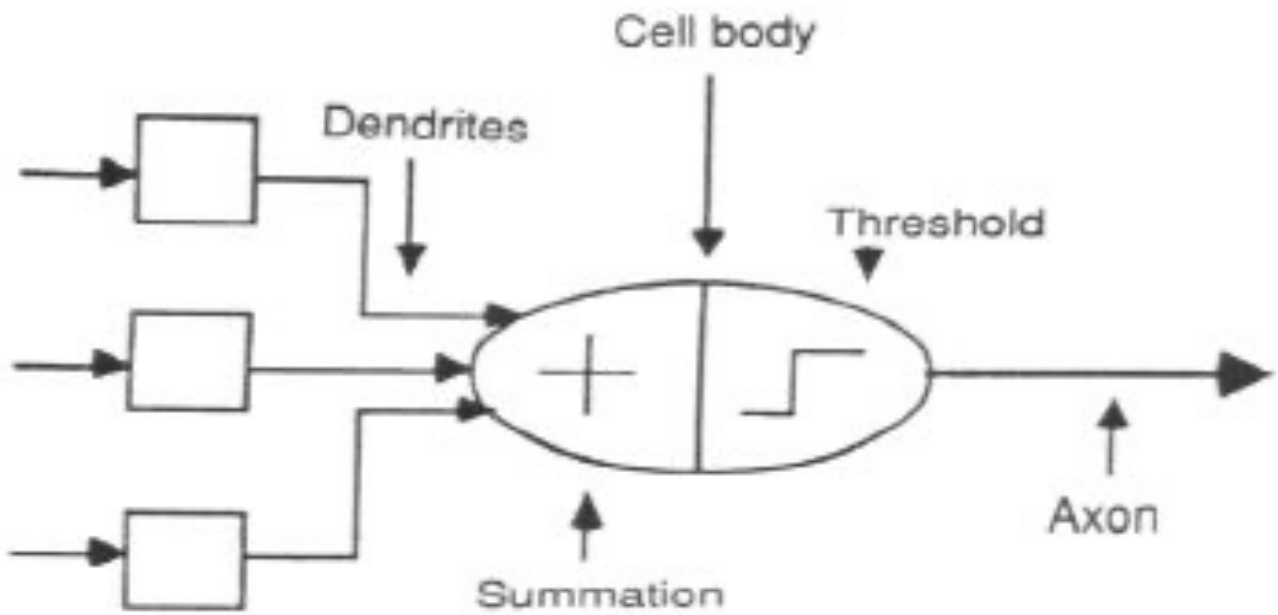
A few interesting applications, with highly competitive results:

- On-line Handwriting Recognition, zip codes
- 3D object recognition
- Stock forecasting
- Intrusion Detection Systems (IDSs)
- Image classification
- Detecting Steganography in digital images
- Medical applications: diagnostics, survival rates ...
- Technical: Combustion Engine Knock Detection
- Elementary Particle Identification in High Energy Physics
- Bioinformatics: protein properties, genomics, microarrays
- Information retrieval, text categorization

Artificial Neural Network

- “Simulating” biological neural systems
- Consisting of many nodes or *neurons* linked by weighted interconnections
- Paralleling many competing optimization hypothesis
- Input patterns (vectors) ➔output patterns
  - Supervised
  - Unsupervised

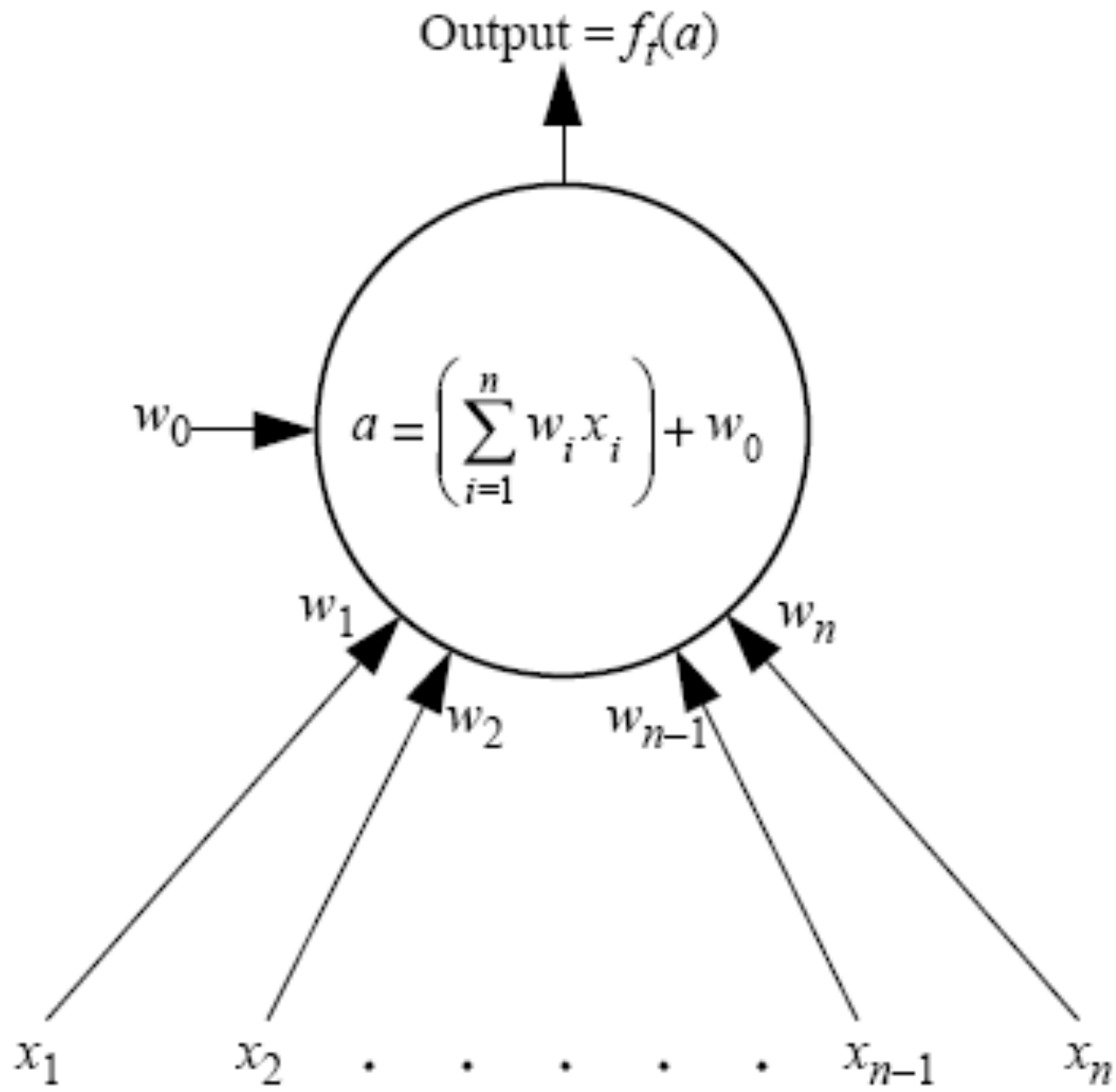

Cell body
Nucleus
Axon
Dendrites
Axon
Synapse
Dendrites


Cell body
Dendrites
Threshold
Summation
Axon

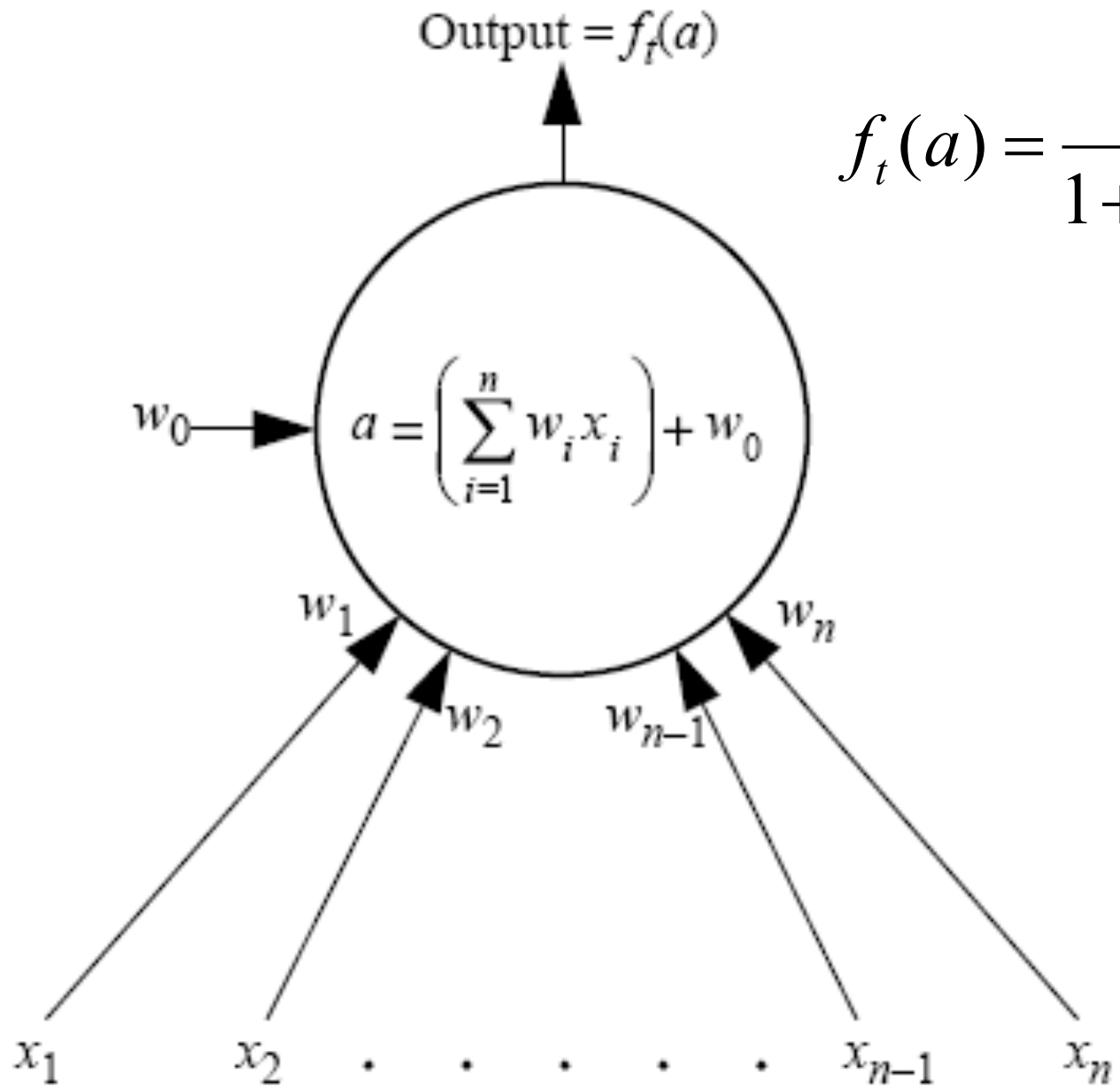
ANN Applications

- Nonlinear estimation
- Classification
- Clustering
- Pattern recognition

Neural Activation

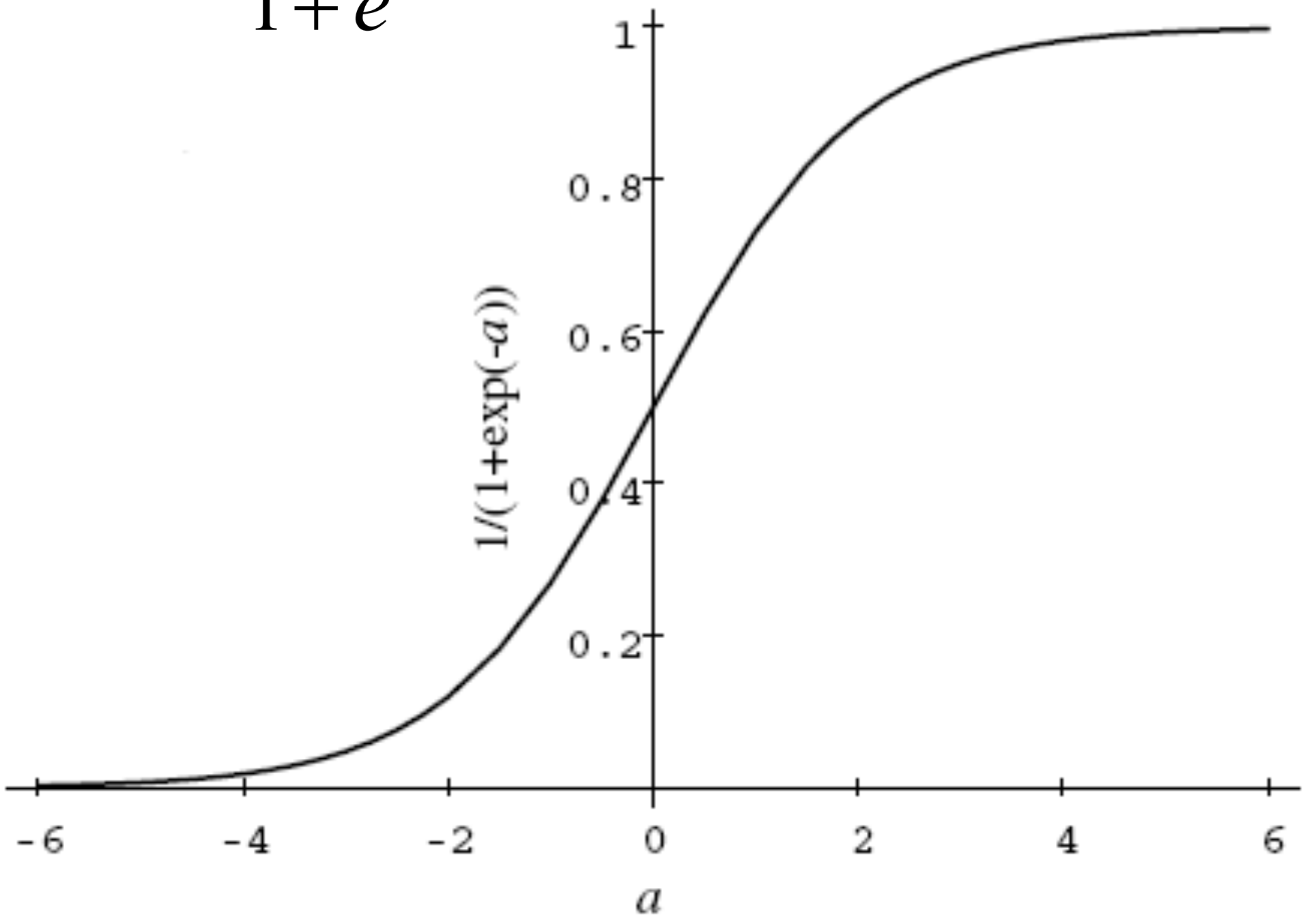


Neural Activation



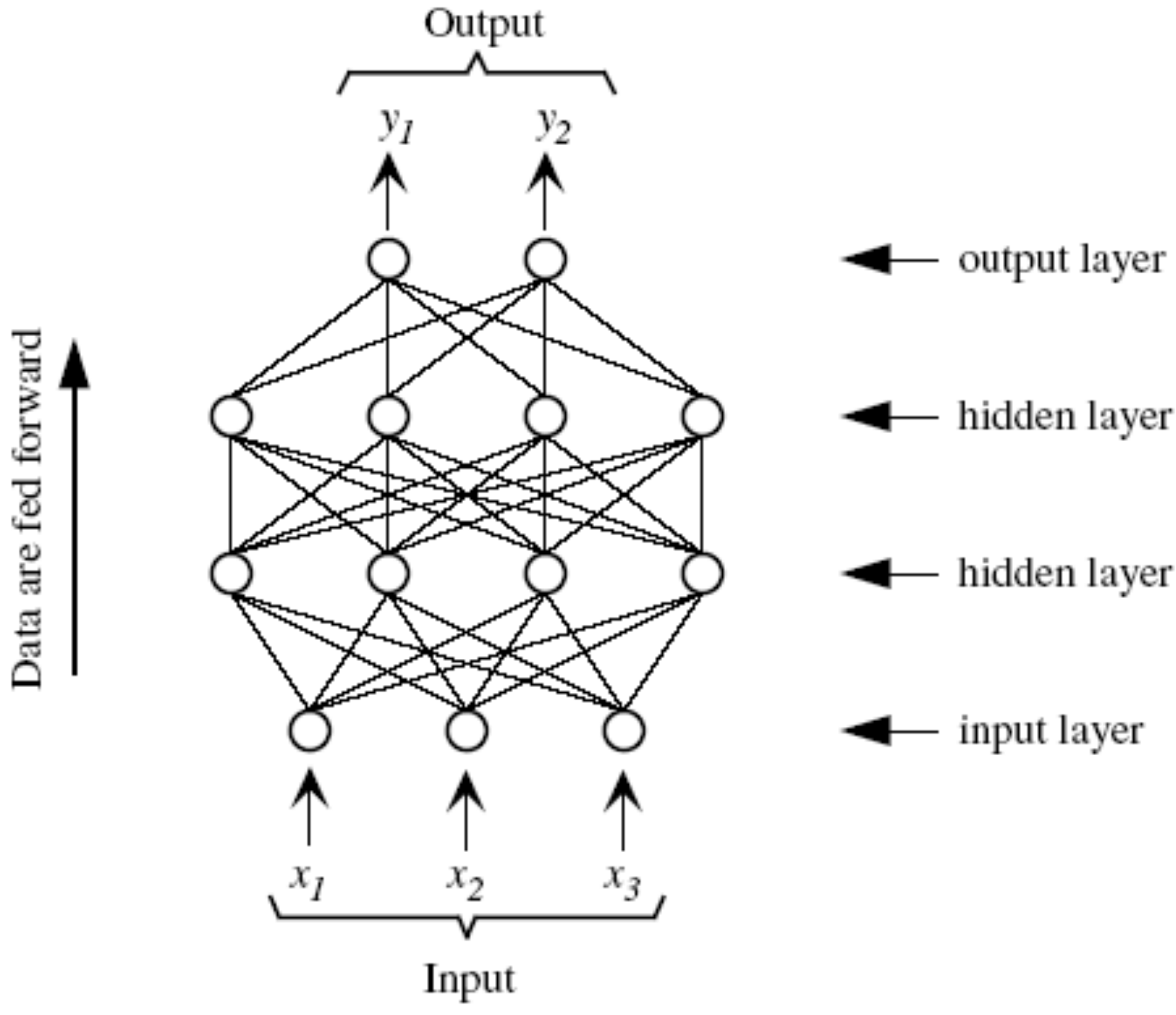
$$f_t(a) = \frac{1}{1+e^{-a}}$$

$$f_t(a) = \frac{1}{1+e^{-a}}$$



Single-multi layer perceptrons

- Perceptrons are organized as layers
- Each neuron is totally connected to ones in above and below, not in the same layers
- Multilayer Perceptron (MLP) or feedforward networks
- Input – Hidden – Output layers
- No hidden ➔ single layer perceptron (SLP)


Output
y_1
y_2
output layer
hidden layer
hidden layer
input layer
Data are fed forward
x_1
x_2
x_3
Input

Classification Problem

- Winner takes all rules
- Separation of state space

$$\left(\sum_{i=1}^{n} w_i x_i\right) + w_0 = 0$$

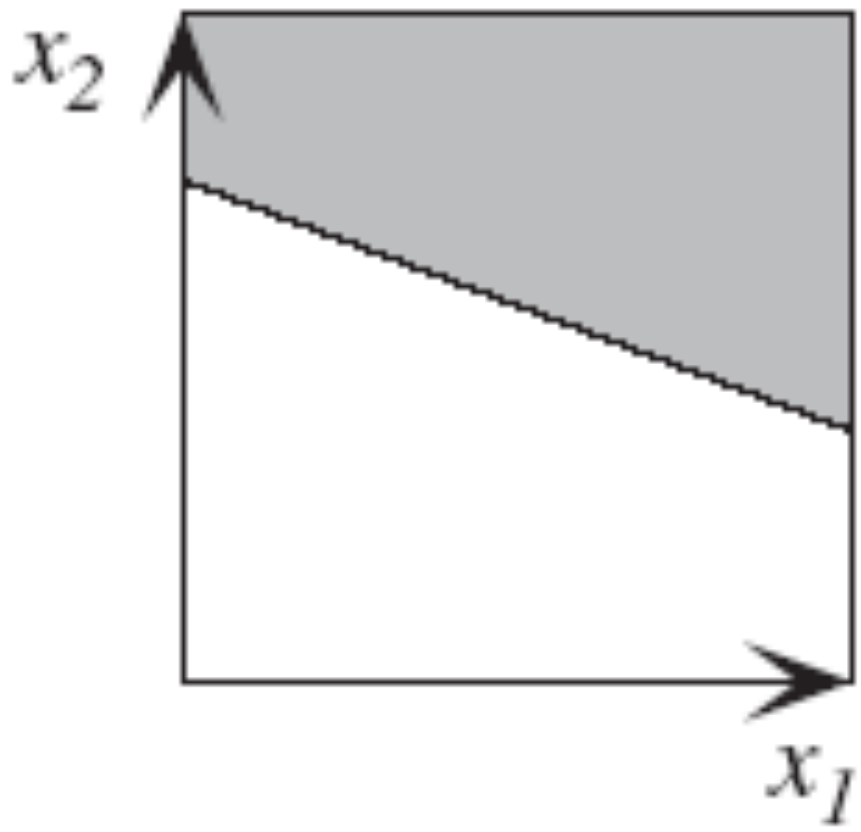
Classification Problem

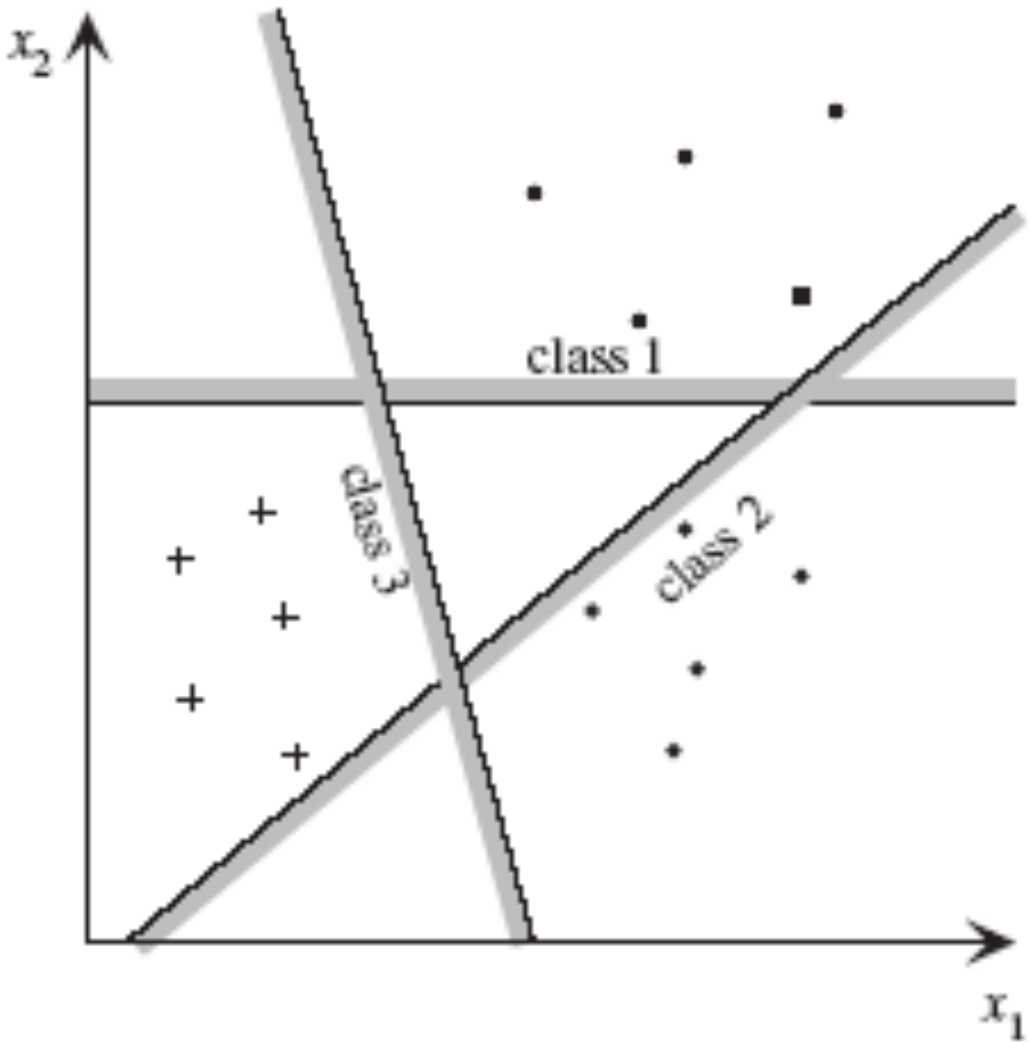
- Winner takes all rules
- Separation of state space

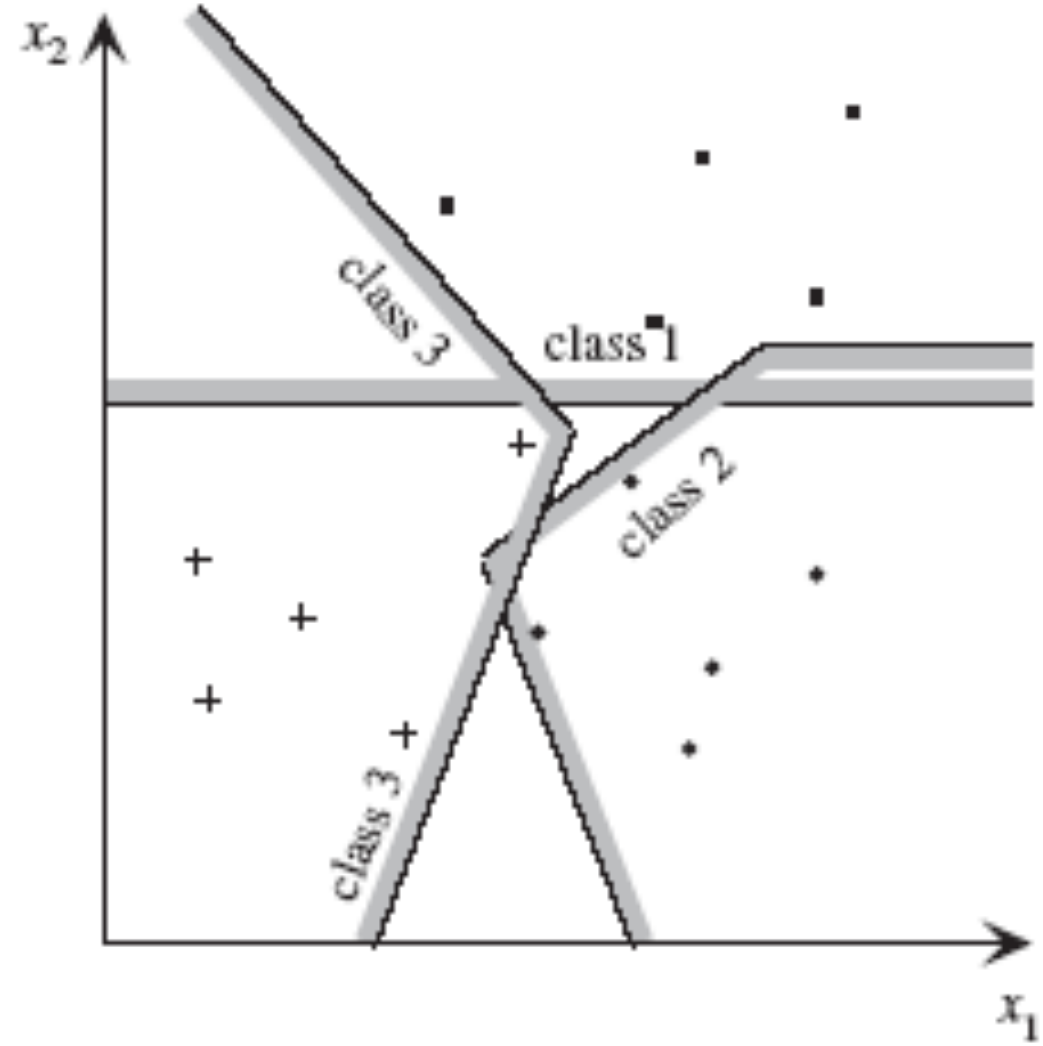
$$\left(\sum_{i=1}^{n} w_i x_i\right) + w_0 = 0$$

- No hidden, 2 input, 1 output

$$x_2 = \frac{-w_1}{w_2} x_1 - \frac{w_0}{w_2}$$

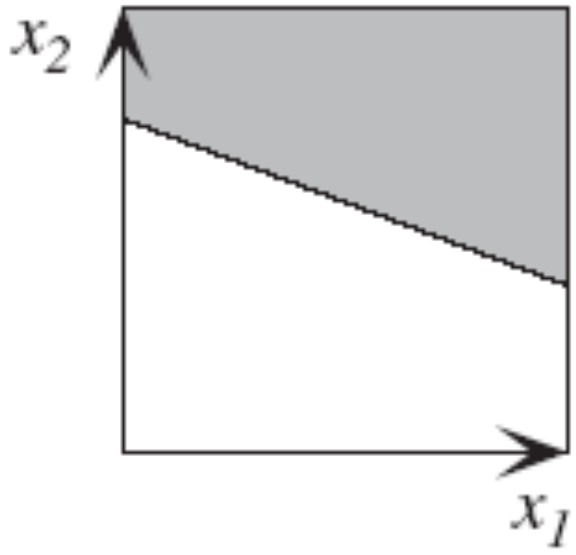

x_2
x_1


x_2
class 1
class 3
class 2
x_1


x_2
class 3
class 1
class 2
class 3
x_1

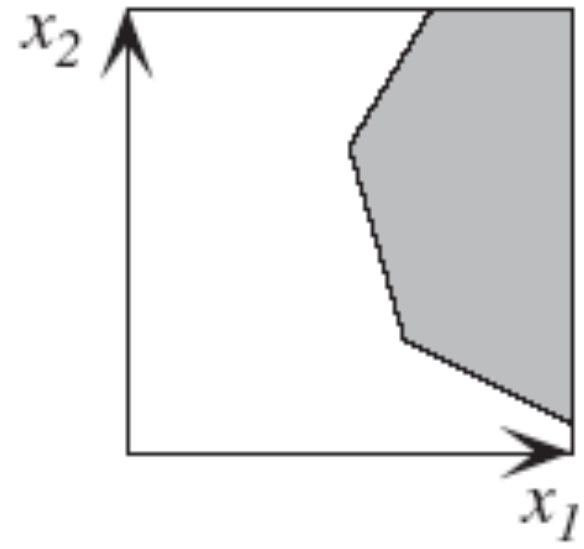
(a) No hidden layers

Region is a half plane bounded by a hyperplane



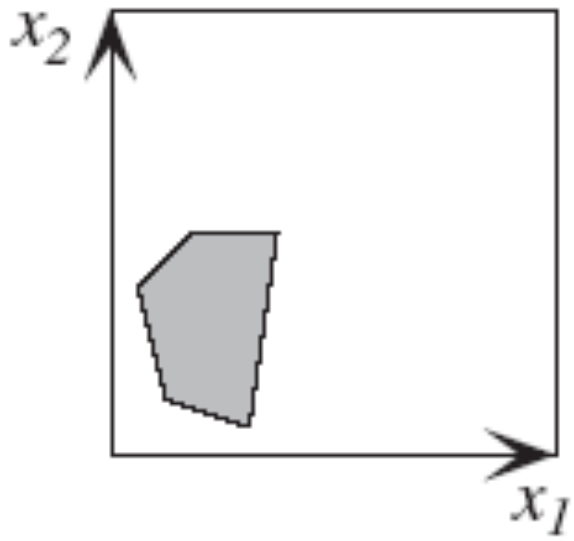
(b) 1 hidden layer, step transfer function

Convex open region



(c) 1 hidden layer, step transfer function

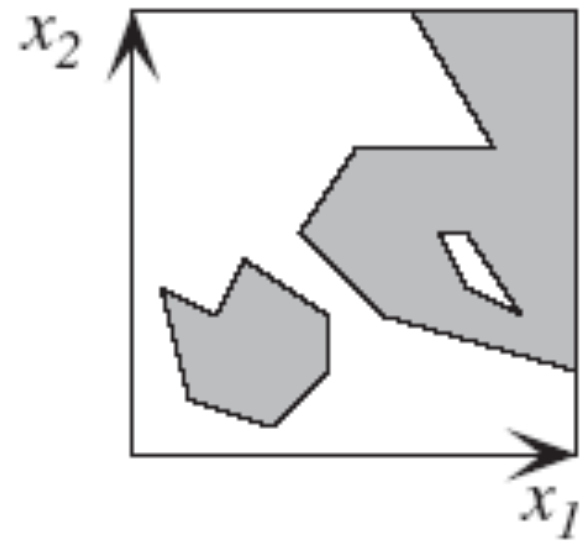
Convex closed region



(d) 2 hidden layers, step transfer function; o

1 hidden layer, smooth transfer function

Regions of arbitrary complexity can be defined



Training perceptrons

- At node B_i, connected to A_j, output from A_j is y_{Aj}

$$w_{Bii} = y_{Aj}$$

Training perceptrons

- At node B_i, connected to A_j, output from A_j is y_{Aj}

$$w_{Bii} = \delta_{Bi} y_{Aj}$$

  - $\delta_{Bi:}$ associated error term

Training perceptrons

- At node B_i, connected to A_j, output from A_j is y_{Aj}

$$w_{Bii} = \eta \delta_{Bi} y_{Aj}$$

  - η. learning rate

Training perceptrons

- At node B_i, connected to A_j, output from A_j is y_{Aj}

$$w_{Bii} = \eta \delta_{Bi} y_{Aj} + \alpha(w_{Bij})$$

  - α: momentum

Training perceptrons

- At node B_i, connected to A_j, output from A_j is y_{Aj}

$$w_{Bii} = \eta \delta_{Bi} y_{Aj} + \alpha(w_{Bij})$$

Training perceptrons

- At node B_i, connected to A_j, output from A_j is y_{Aj}

$$w_{Bii} = \eta\delta_{Bi}y_{Aj} + \alpha(w_{Bij})$$

- Output layer:
  - $\delta_{Bi} = f'_t(y_{Bi})(d_i - y_{Bi}) = y_{Bi}(1-y_{Bi})(d_i - y_{Bi})$

Training perceptrons

- At node B_i, connected to A_j, output from A_j is y_{Aj}

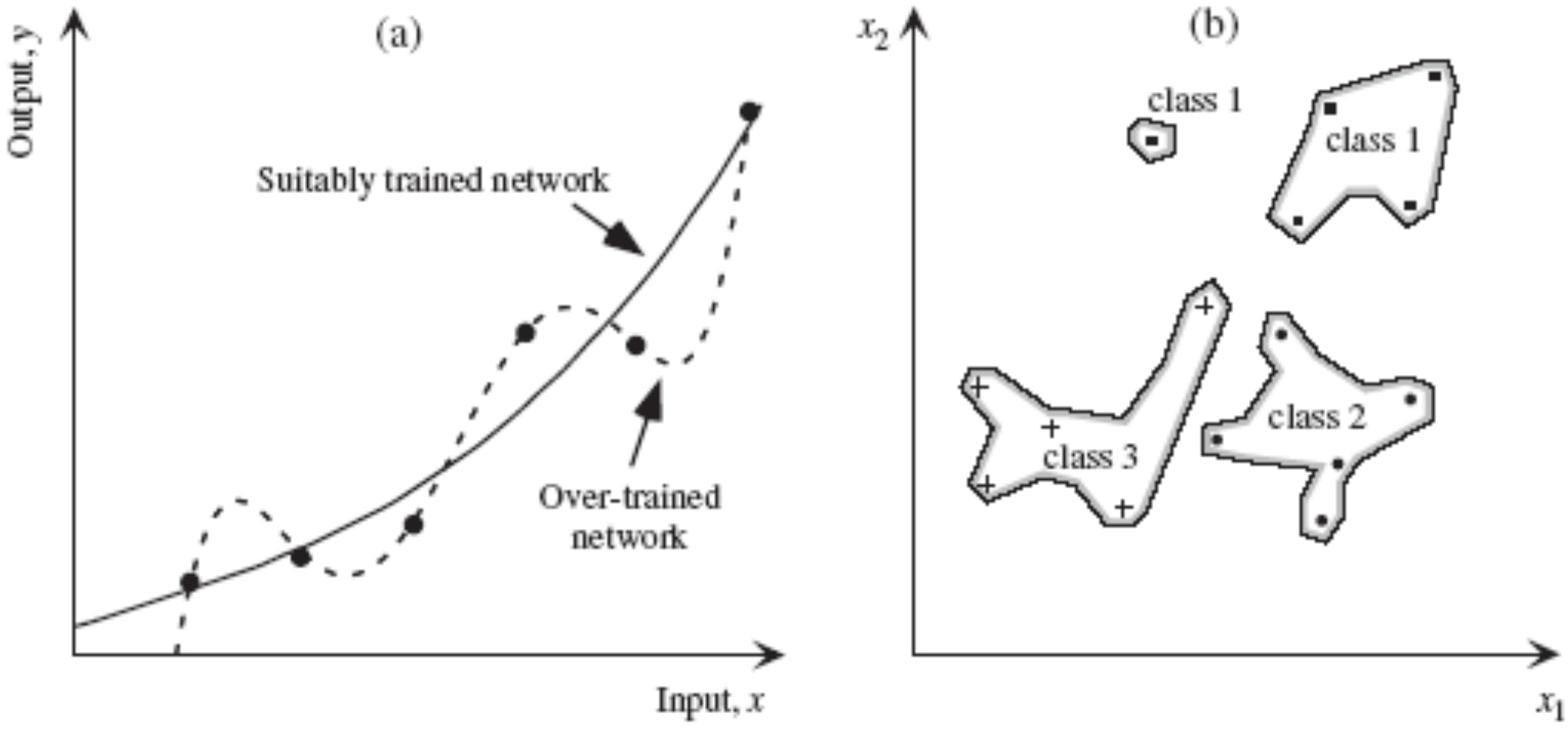
$$w_{Bii} = \eta \delta_{Bi} y_{Aj} + \alpha(w_{Bij})$$

- Hidden layer:
  - $\delta_{Bi} = f'_t(y_{Bi}) \sum \delta_{Bi} w_{Bij} = y_{Bi}(1-y_{Bi}) \sum \delta_{Bi} w_{Bij}$

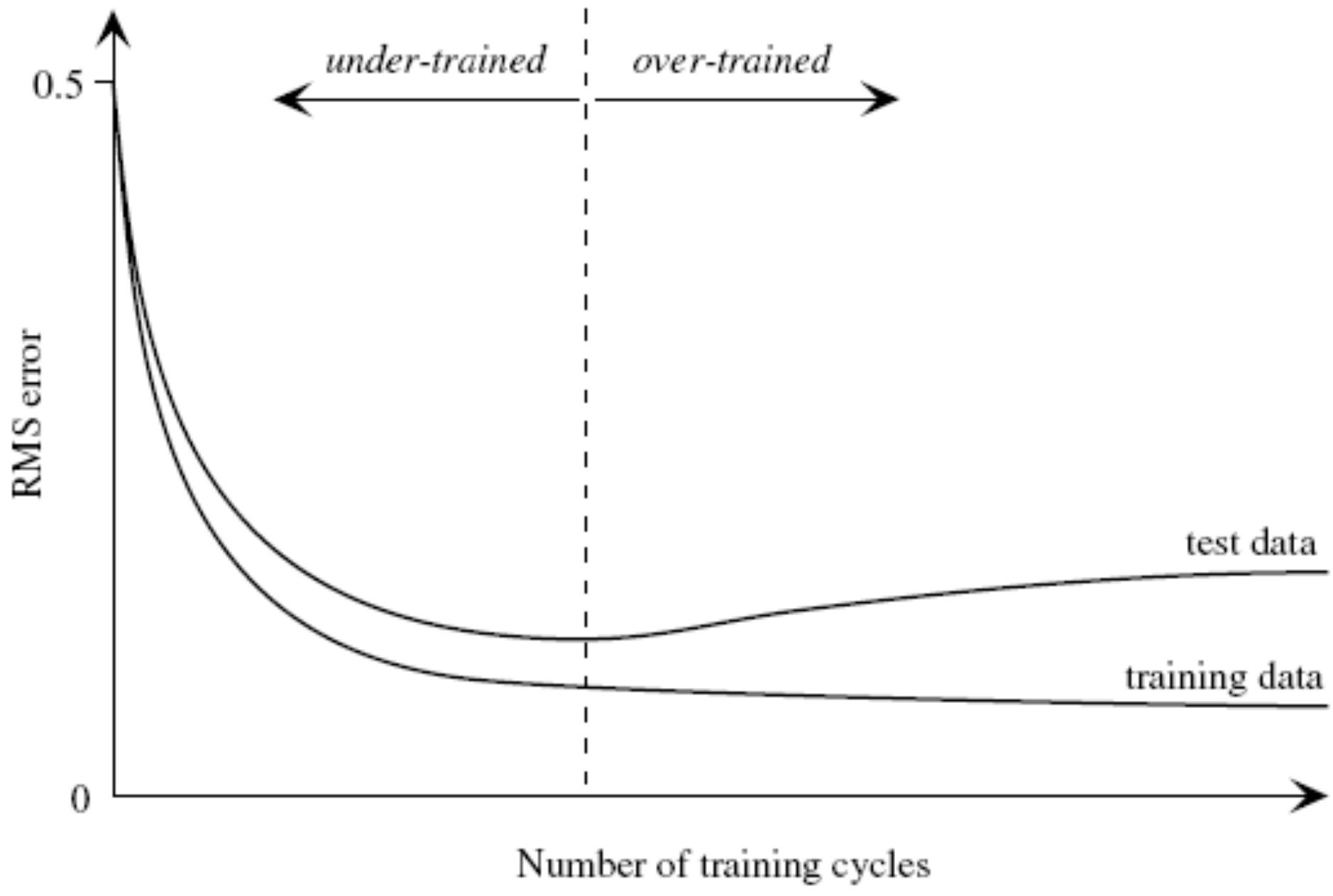
Training perceptrons

- Algorithm: Figure 8.7

Over-training



Under-training vs. over-training



Feature Selection

- Why we need FS:
  1. to improve performance (in terms of speed, predictive power, simplicity of the model).
  2. to visualize the data for model selection.
  3. To reduce dimensionality and remove noise.

- *Feature Selection* is a process that chooses an optimal subset of features according to a certain criterion.

Perspectives:
Selection Criteria

- Information Measures.
  - Information serves to measure the uncertainty of the receiver when she/he receives a message.
  - Shannon's Entropy:

$$-\sum_i P(c_i)\log_2 P(c_i).$$

- Information gain:

$$IG(A) = I(D) - \sum_{j=1}^{p} \frac{|D_j|}{|D|} I(D_j^A)$$

Perspectives:
Selection Criteria

- Distance Measures.
  - Measures of separability, discrimination or divergence measures . The most typical is derived from distance between the class conditional density functions.

	Mathematical form
Euclidean distance	$D_e = \left\{\sum_{i=1}^{m} (x_i - y_i)^2\right\}^{\frac{1}{2}}$
City-block distance	$D_{cb} = \sum_{i=1}^{m} \lvert x_i - y_i \rvert$
Cebyshev distance	$D_{ch} = \max_i \lvert x_i - y_i \rvert$
Minkowski distance of order m	$D_M = \left\{\sum_{i=1}^{m} (x_i - y_i)^m\right\}^{\frac{1}{m}}$
Quadratic distance Q, positive definite	$D_q = \sum_{i=1}^{m} \sum_{j=1}^{m} (x_i - y_i) Q_{ij} (x_j - y_j)$
Canberra distance	$D_{ca} = \sum_{i=1}^{m} \frac{\lvert x_i - y_i \rvert}{x_i + y_i}$
Angular separation	$D_{as} = \frac{\sum_{i=1}^{m} x_i \cdot y_i}{\left[\sum_{i=1}^{m} x_i^2 \sum_{i=1}^{m} y_i^2\right]^{\frac{1}{2}}}$

Perspectives:
Selection Criteria

- Dependence Measures.
  - known as measures of association or correlation.
  - Its main goal is to quantify how strongly two variables are correlated or present some association with each other, in such way that knowing the value of one of them, we can derive the value for the other.
- *Pearson correlation* coefficient:

$$\rho(X, Y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\left[\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2\right]^{\frac{1}{2}}}$$

Perspectives: Selection Criteria

- Consistency Measures.
  - They attempt to find a minimum number of features that separate classes as the full set of features can.
  - They aim to achieve **P(C|FullSet) = P(C|SubSet)**.
  - An inconsistency is defined as the case of two examples with the same inputs (same feature values) but with different output feature values (classes in classification).

Case Study: Digitalized Bank

Good vs Not Good

EmailAddress	PhoneNumber	Age
07125015@st.hcmuaf.edu.vn	1267816879	28
10ltranthaibaongoc@gmail.com	918599269	19
1991.hoang@gmail.com	1668638636	25
8395huyen@gmail.com	1284838000	21
aimondnhim@gmail.com	1678681581	21
ainguyen.capricorn@gmail.com	973067210	20
alammx6@gmail.com	978881100	41
alex.doanng@gmail.com	916705553	32

EmailAddress	PhoneNumber	Age
11091997nlmh@gmail.com	1215936841	19
12111026@st.hcmuaf.edu.vn	968590410	22
12112234@st.hcmuaf.edu.vn	1285544046	23
13131555@st.hcmuaf.edu.vn	1282525481	21
13dks01@gmail.com	1629087078	21
15145002@st.hcmuaf.edu.vn	962455810	19
1530010061@sv.hotec.edu.vn	1865095493	20
1551113@HCMUT.EDU.VN	935002331	19
15520080@gm.uit.edu.vn	938451517	19
1554010359van@ou.edu.vn	1699979745	19
1808vananh@gmail.com	1629394246	20

Enriched Database

name	birthday	hometown	location	email	work	subscribers	friends	education	last_update	gender	likes
An An	17-Aug	{"id":"107417125954798","name":"Chaozhou"}				702	1020		1476772484		[]
Anh Xuan	13-Dec					0	166		1476772484	0	[]
Chang Chang		{"id":"108458769184495","name":"Ho Chi Minh City, Viet				0	7		1476772484	0	[]
Linh Vo		{"id":"106521	{"id":"108458769184495","name":"Ho Chi			243			1476772484	0	[{"name":"Fut
Quang Tạ		{"id":"108458	{"id":"108458769184495","name":"Ho Chi			0	310	[{"school":{"ic	1476772484	1	[{"name":"LaL
Jin Lee						102	916		1476772484		[{"name":"Wil
Nguyen Tan Quoc Anh		{"id":"110512	{"id":"108458769184495","r		[{"employer":	0	441	[{"school":{"ic	1476772484	1	[{"name":"Shi
Nguoi Gian Doi			{"id":"106388046062960","name":"Hanoi,			0	0		1476772484	0	[]
Nguyen Le Phuong Uyen					[{"employer":	0		[{"school":{"ic	1476772484	0	[{"name":"Bin
Giang Quan		{"id":"108458	{"id":"108458769184495","name":"Ho Chi			320	326	[{"school":{"ic	1476772484	0	[{"name":"Rev
Trần Khánh						0			1476772484	1	[{"name":"End
Panda Huỳnh		{"id":"114222	{"id":"108458769184495","r		[{"employer":	17	1317	[{"classes":[{"i	1476772484	0	[{"name":"Sib
Nguyen Anh Thoa		{"id":"108458	{"id":"108724149156796","name":"Thành			111			1476772484	0	[{"name":"Aot
Vinh Phạm					[{"description	79	713	[{"school":{"ic	1476772484	1	[{"name":"Nh
Thành Nguyễn						66			1476772484		[{"name":"Toz
Huyen Nguyen		{"id":"108458	{"id":"108724149156796","r		[{"employer":	276		[{"school":{"ic	1476772484	0	[{"name":"H&
Annariss Than	24-Feb	{"id":"112089	{"id":"351759091676222","r		[{"employer":	57		[{"school":{"ic	1476772484		[{"name":"qua
Bondney Tuấn		{"id":"110512632303092","name":"Vinh Long"}				191		[{"concentrati	1476772484	1	[{"name":"TIK
Thanh Thieu		{"id":"108458	{"id":"108458769184495","name":"Ho Chi			0	455		1476772484	1	[{"name":"9G.

Raw information

id	phone	facebook id	name	birthday	hometown	location	email	work	subscribers

friends	education	last_update	gender	likes	interests	verified	interested_in	isgood	Age

Feature Selection

hometown
subscriber_count
friends_count
gender
subscribedto_count
Education occupation
relationship status
languages

age
likes scores
book
Interests games
Groups
Movies music
Sports television
locations

Normalization

No	Feature	How to preprocess features
1.	Hometown	Classified city into levels. Two cities of the same levels should have smaller distances, ranking base on per capital income in a month. Details: [Standardization] Home town.
2	Subscriber_count	Do nothing
3	Friends_count	Do nothing
4	Gender	female = 0, None = 0.5, male = 1
5	Subscribedto_count	do nothing
6	Education	classified into levels. Details: [Standardization] Education
7	Occupation	classified job into levels. Details: [Standardization] Occupation
8	Relationship status	Classified status into levels. Details: [Standardization] Relationship status