CS6510
Applied Machine Learning

Classifiers

19 Aug 2017

Vineeth N Balasubramanian

आई आई टी हैदराबाद
IIT Hyderabad

Administrivia

- Google Classroom portal – please ensure you have access to it. If not, please let us know.
  - Separate portal for EMDS and non-EMDS students?
- HW1 posted and is due on 27th Aug
- Evaluation of HW and Assignments
  - Viva voce will be held after assignment submission
  - Final score on HW/assignment = graded score * viva score
  - Not attending the viva scheduled by TA -> zero viva score
- Any questions, please let us know

Classification Methods

- k-Nearest Neighbors
- Decision Trees
- Naïve Bayes
- Support Vector Machines
- Logistic Regression
- Neural Networks
- Ensemble Methods (Boosting, Random Forests)

Probability: Review

Random variable	• Result of tossing a coin is from {Heads, Tails} • Random var X from {1,0} • Bernoulli: $P\{X=1\} = p_o^X (1 - p_o)^{(1-X)}$
Joint and conditional probability	P(A|B) = P(A, B)/P(B)
Bayes Theorem	P(A|B) = P(B|A) P(A)/P(B)

Illustration

A	0	0	1	1	1	0
B	0	1	1	0	1	1

- $P(A=1) = 3/6 = 1/2$, $P(A=0) = 3/6 = 1/2$.
- $P(B=1) = 4/6 = 2/3$, $P(B=0) = 2/6 = 1/3$.
- $P(A=1, B = 1) = 2/6 = 1/3$.
- $P(A=1 \mid B = 1) = P(A=1, B = 1) / P(B=1) = 1/2$.
- $P(B=1 \mid A = 1) = P(B=1, A = 1) / P(A=1) = 2/3$.
- $P(A=1 \mid B = 1) P(B=1)/P(A=1) = 2/3 = P(B=1 \mid A =1)$.
  – Bayes' Theorem

Naïve Bayes Classifier

- Goal: Learning function f: x -> y
  - *Y*: One of *k* classes (e.g. spam/ham, digit 0-9)
  - $X=X_1,\ldots,X_n$: Values of attributes (numeric or categorical)
- Probabilistic classification
  - Most probable class given observation: $\hat{y} = \arg\max_{y} P(y|x)$
- Bayesian probability of a class

$$P(y|x) = \frac{\overbrace{P(x|y)}^{\text{class model}}\,\overbrace{P(y)}^{\text{prior}}}{\underbrace{\sum_{y'} P(x|y')P(y')}_{\text{normalizer } P(x)}}$$

Bayes Theorem

Bayes Theorem: Example

- Given:
  - A doctor knows that meningitis causes stiff neck 50% of the time
  - Prior probability of any patient having meningitis is 1/50,000
  - Prior probability of any patient having stiff neck is 1/20

- If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

What is Naïve about it?

- Consider each attribute and class label as random variables
- Given a record with attributes $(A_1, A_2, \ldots, A_n)$
  - Goal is to predict class C
  - Specifically, we want to find the value of C that maximizes $P(C | A_1, A_2, \ldots, A_n)$
- Can we estimate $P(C | A_1, A_2, \ldots, A_n)$ directly from data?

What is Naïve about it?

- Approach:
  - compute the posterior probability $P(C \mid A_1, A_2, \ldots, A_n)$ for all values of C using the Bayes theorem

Posterior

$$\boxed{P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}}$$

Prior

Likelihood

  - Choose value of C that maximizes
    $P(C \mid A_1, A_2, \ldots, A_n)$
  - Equivalent to choosing value of C that maximizes
    $P(A_1, A_2, \ldots, A_n \mid C)\, P(C)$

Maximum A Posteriori (MAP) Rule

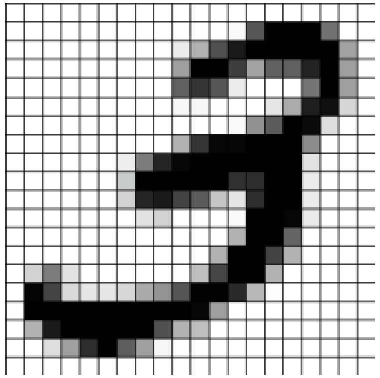
- How to estimate $P(A_1, A_2, \ldots, A_n \mid C)$?

What is Naïve about it?

$$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

How to compute if x is made of multiple attributes?

- 20 x 20 image of digit = 2^{400} possible combinations!



Naïve Bayes Classifier

- Assume independence among attributes A_i when class is given:
  - $P(A_1, A_2, \ldots, A_n \mid C_j) = P(A_1 \mid C_j)\, P(A_2 \mid C_j) \ldots P(A_n \mid C_j)$
  - Can estimate $P(A_i \mid C_j)$ for all A_i and C_j.
  - New point is classified to C_j if
    $P(C_j)\ \Pi_j\, P(A_i \mid C_j) = P(C_j)\, P(A_1 \mid C_j)\, P(A_2 \mid C_j) \ldots P(A_n \mid C_j)$
    is maximal

Maximum Likelihood Hypothesis

- Assume that all hypotheses (classes) are equally probable a priori, i.e., $P(C_i) = P(C_j)$ for all i,j.
- This is called assuming a uniform prior. It simplifies computing the posterior:
  - $C_{ML} = \mathrm{argmax}_c\, P(A_1, A_2, \ldots, A_n | C)$
- This hypothesis is called the maximum likelihood hypothesis.

Example

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Example: Learning Phase

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=*Yes*	Play=No
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

$P(\text{Play}=Yes) = 9/14$ $P(\text{Play}=No) = 5/14$

Example: Test Phase

- Given a new instance,
- x'=(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)
- Look up tables

P(Outlook=*Sunny*|Play=*Yes*) = 2/9	P(Outlook=*Sunny*|Play=*No*) = 3/5
P(Temperature=*Cool*|Play=*Yes*) = 3/9	P(Temperature=*Cool*|Play==*No*) = 1/5
P(Huminity=*High*|Play=*Yes*) = 3/9	P(Huminity=*High*|Play=*No*) = 4/5
P(Wind=*Strong*|Play=*Yes*) = 3/9	P(Wind=*Strong*|Play=*No*) = 3/5
P(Play=*Yes*) = 9/14	P(Play=*No*) = 5/14

- MAP rule

$P(Yes|\mathbf{x}')$: $[P(Sunny|\text{Yes})P(Cool|\text{Yes})P(High|\text{Yes})P(Strong|\text{Yes})]P(\text{Play}=Yes) = 0.0053$

$P(No|\mathbf{x}')$: $[P(Sunny|\text{No})\ P(Cool|\text{No})P(High|No)P(Strong|No)]P(\text{Play}=No) = 0.0206$

Given the fact $P(Yes|\mathbf{x}') < P(No|\mathbf{x}')$, we label $\mathbf{x}'$ to be "*No*".

Example: Another

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

A: attributes

M: mammals

N: non-mammals

$$P(A|M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A|N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A|M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A|N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

P(A|M)P(M) > P(A|N)P(N)

=> Mammals

Pros and Cons

- **Combines prior knowledge and observed data:** prior probability of a hypothesis multiplied with probability of the hypothesis given the training data
- **Probabilistic hypothesis:** outputs not only a classification, but a probability distribution over all classes
- Robust to isolated noise points
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- **Incrementality:** With each training example, the prior and the likelihood can be updated dynamically: flexible and robust to errors
- Independence assumption may not hold always

Practical Issues

- For continuous attributes:
  - Discretize the range into bins
    - one ordinal attribute per bin
    - violates independence assumption
  - Two-way split: $(A < v)$ or $(A > v)$
    - choose only one of the two splits as new attribute
  - Probability density estimation:
    - Assume attribute follows a parametrized distribution, e.g. normal distribution
    - Use data to estimate parameters of distribution (e.g., mean and standard deviation) using Maximum Likelihood Estimation
    - Once probability distribution is known, can use it to estimate the conditional probability $P(A_i|c)$

Underflow Prevention

- Multiplying lots of probabilities, which are between 0 and 1 by definition, can result in floating-point underflow.
- Since log(xy) = log(x) + log(y), it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities.
- Class with highest final un-normalized **log probability score** is still the most probable.

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \log P(c_j) + \sum_{i \in positions} \log P(x_i \mid c_j)$$

Density Estimation in Naïve Bayes

- Assume independence among attributes A_i when class is given:
  - $P(A_1, A_2, \ldots, A_n \mid C_j) = P(A_1 \mid C_j)\, P(A_2 \mid C_j) \ldots P(A_n \mid C_j)$
  - Can estimate $P(A_i \mid C_j)$ for all A_i and C_j.
  - New point is classified to C_j if

    $P(C_j)\ \Pi_j\ P(A_i \mid C_j) = P(C_j)\, P(A_1 \mid C_j)\, P(A_2 \mid C_j) \ldots P(A_n \mid C_j)$

    is maximal

We use density estimation methods (e.g. Expectation-Maximization) to obtain the parameters of the distribution. More later when we cover unsupervised learning.

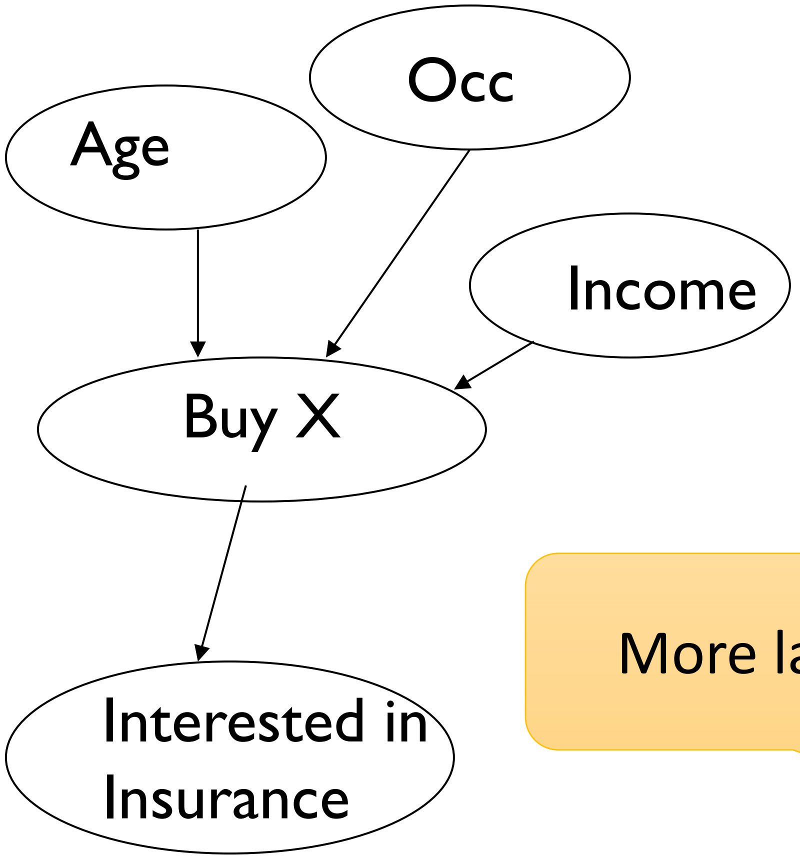
What if this conditional distribution was Gaussian? Or a mixture of Gaussians? Or any other distribution?

Overcoming the Independence Assumption

- Naïve Bayes assumption of conditional independence too restrictive
  - But it is intractable without some such assumptions
- **Bayesian Belief network (Bayesian net)** describe conditional independence among subsets of variables (attributes): combining prior knowledge about dependencies among variables with observed training data.
- Bayesian Net
  - Node = variables
  - Arc = dependency
  - DAG, with direction on arc representing causality
  - Variable A with parents B1, …., Bn has a conditional probability table P (A | B1, …., Bn)

Bayesian Networks: Example

- Age, Occupation and Income determine if customer will buy this product.
- Given that customer buys product, whether there is interest in insurance is now independent of Age, Occupation, Income.
- P(Age, Occ, Inc, Buy, Ins) = P(Age)P(Occ)P(Inc) P(Buy|Age,Occ,Inc)P(Int|Buy)



More later

Classification Methods

- k-Nearest Neighbors
- Decision Trees
- Naïve Bayes
- Support Vector Machines
- Logistic Regression
- Neural Networks
- Ensemble Methods (Boosting, Random Forests)

SVM: Overview and History

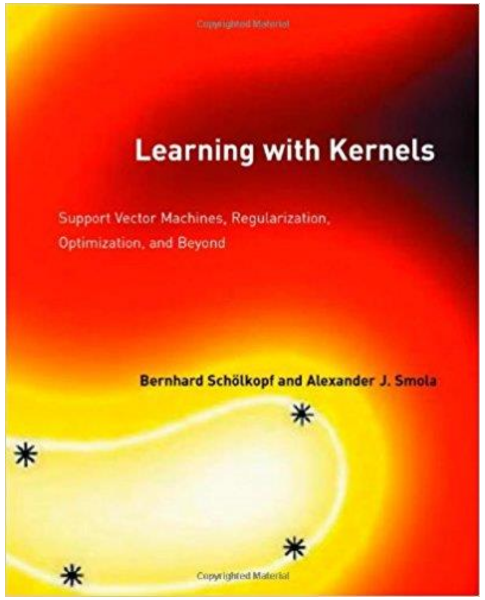
- A discriminative classifier
  - Non-parametric, Inductive
- SVM is inspired from statistical learning theory
- SVM was developed in 1992 by Vapnik, Guyon and Boser
- SVM became popular because of its success in handwritten digit recognition
- Has been one of the go-to methods in machine learning since the mid-1990s (only recently displaced to some extent by deep learning)

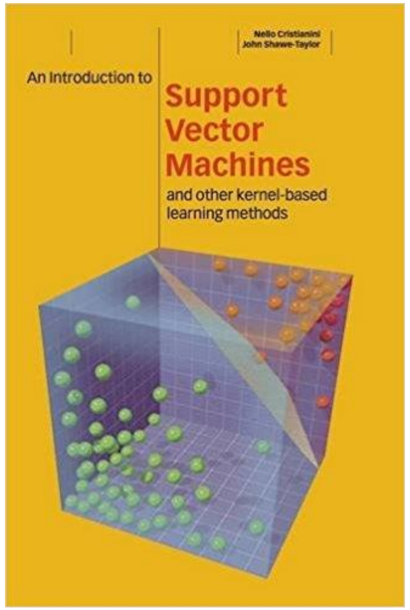
Papers that introduced SVM in its current form

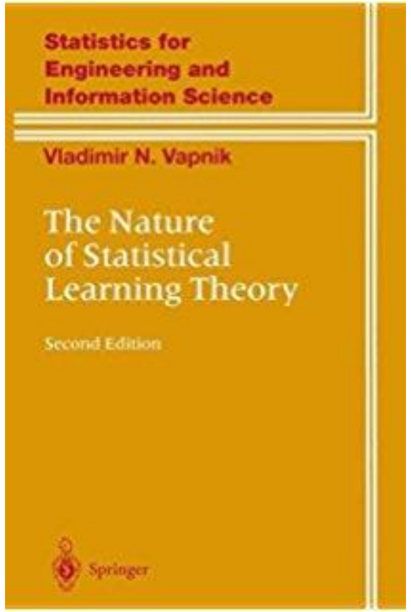
- Boser, B. E.; Guyon, I. M.; Vapnik, V. N. (1992). "A training algorithm for optimal margin classifiers". Proceedings of the fifth annual workshop on Computational learning theory – COLT '92.
- Cortes, C.; Vapnik, V. (1995). "Support-vector networks". Machine Learning. 20 (3): 273–297.

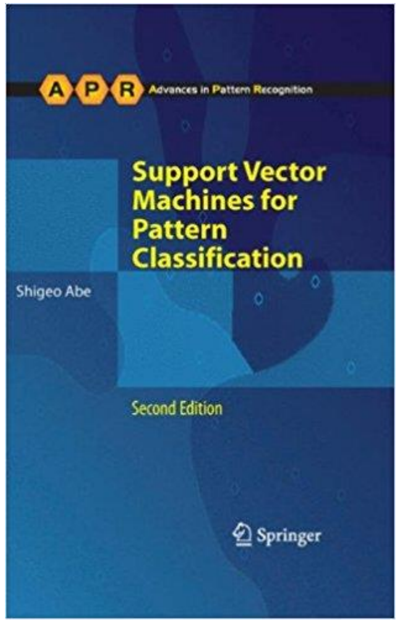
SVM: Overview and History

- Associated key words
  - Large-margin classifier, Max-margin classifier, Kernel methods, Reproducing kernel Hibert space, Statistical learning theory

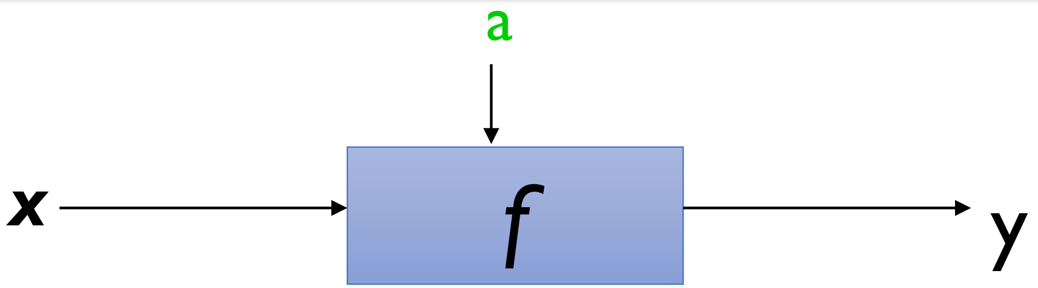




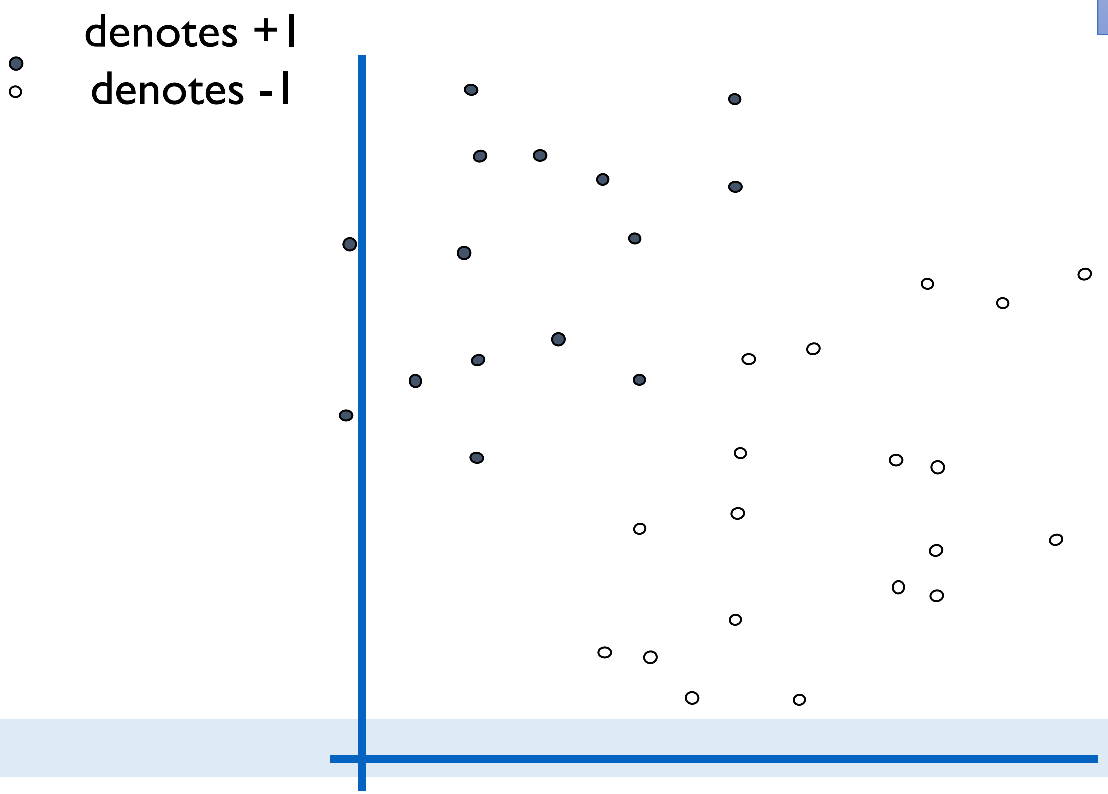




Linear Classifiers

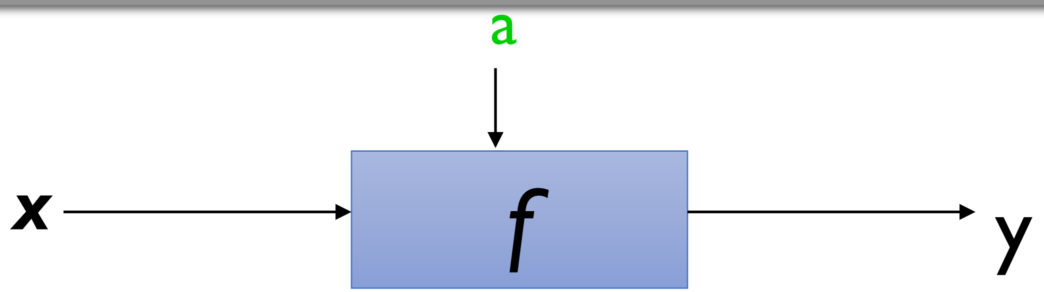


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$

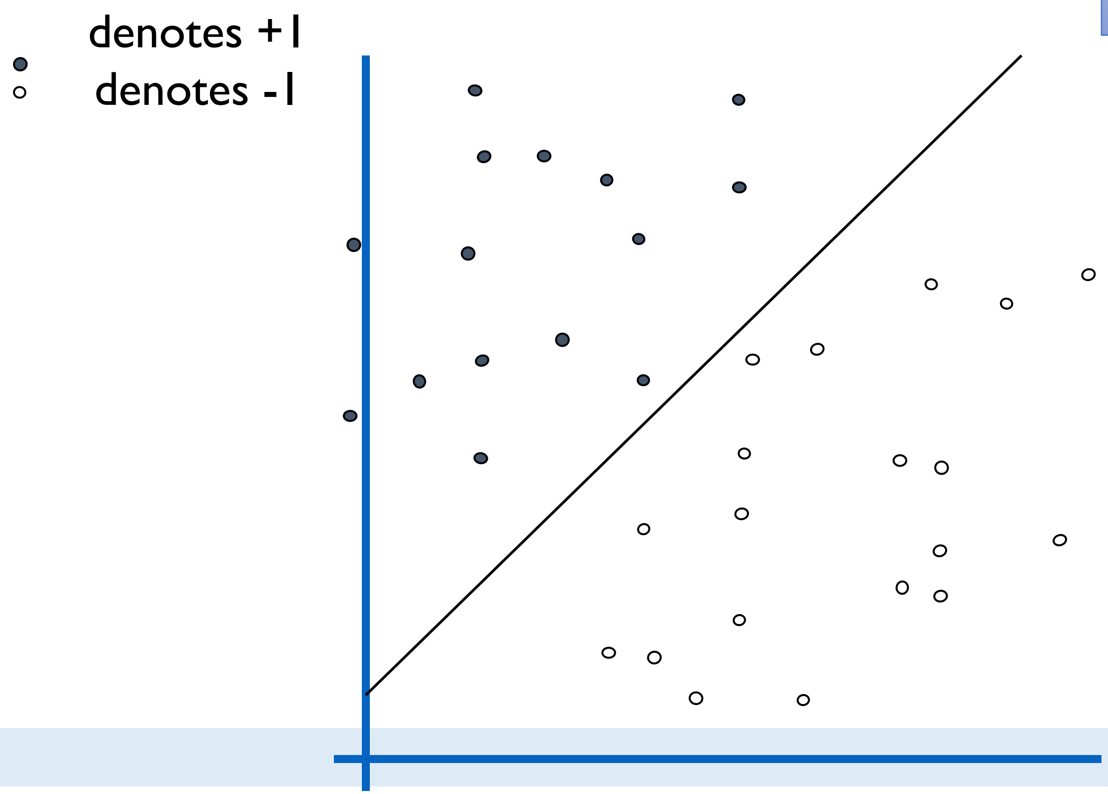


How would you classify this data?

Linear Classifiers

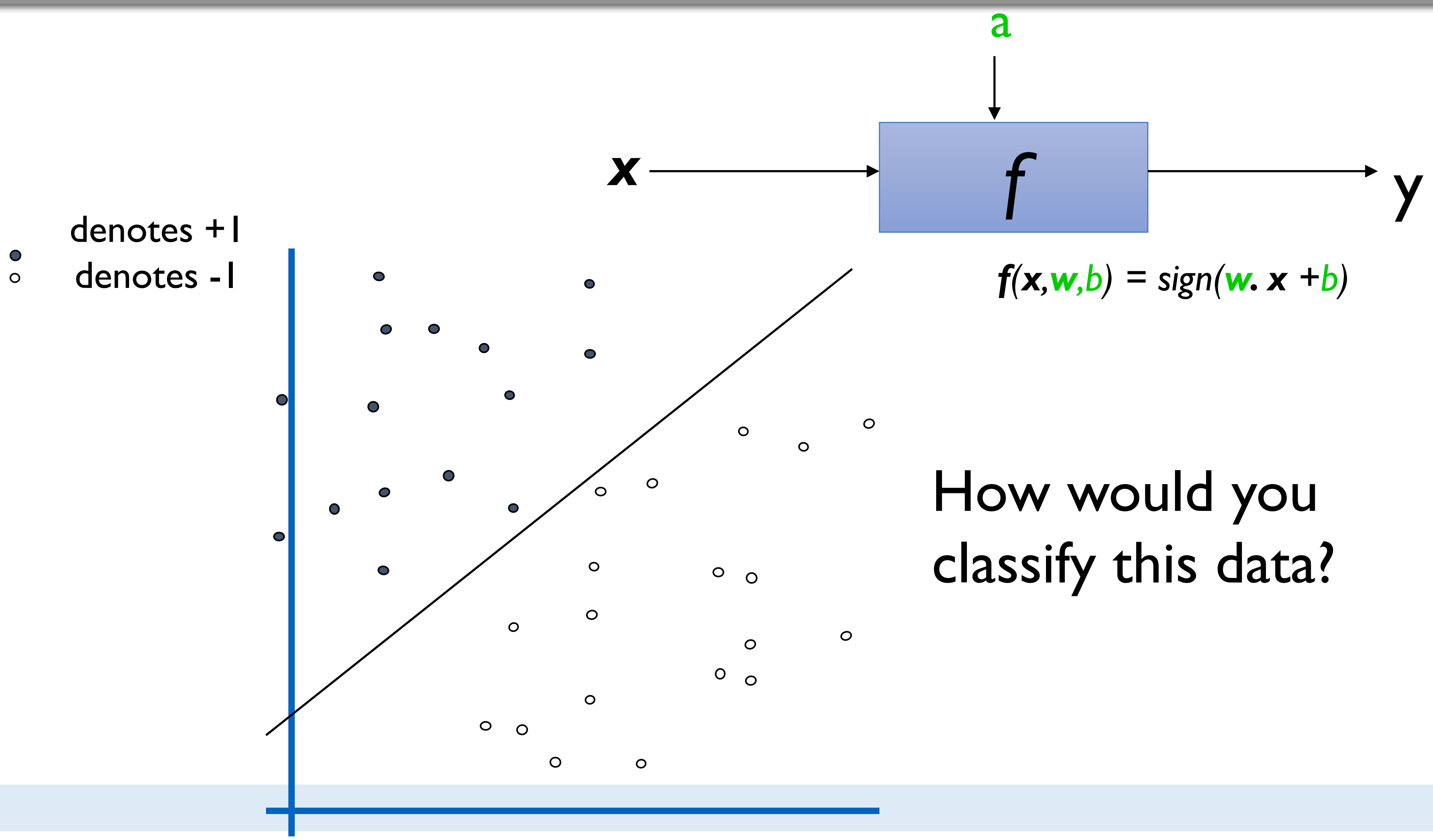


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$



How would you classify this data?

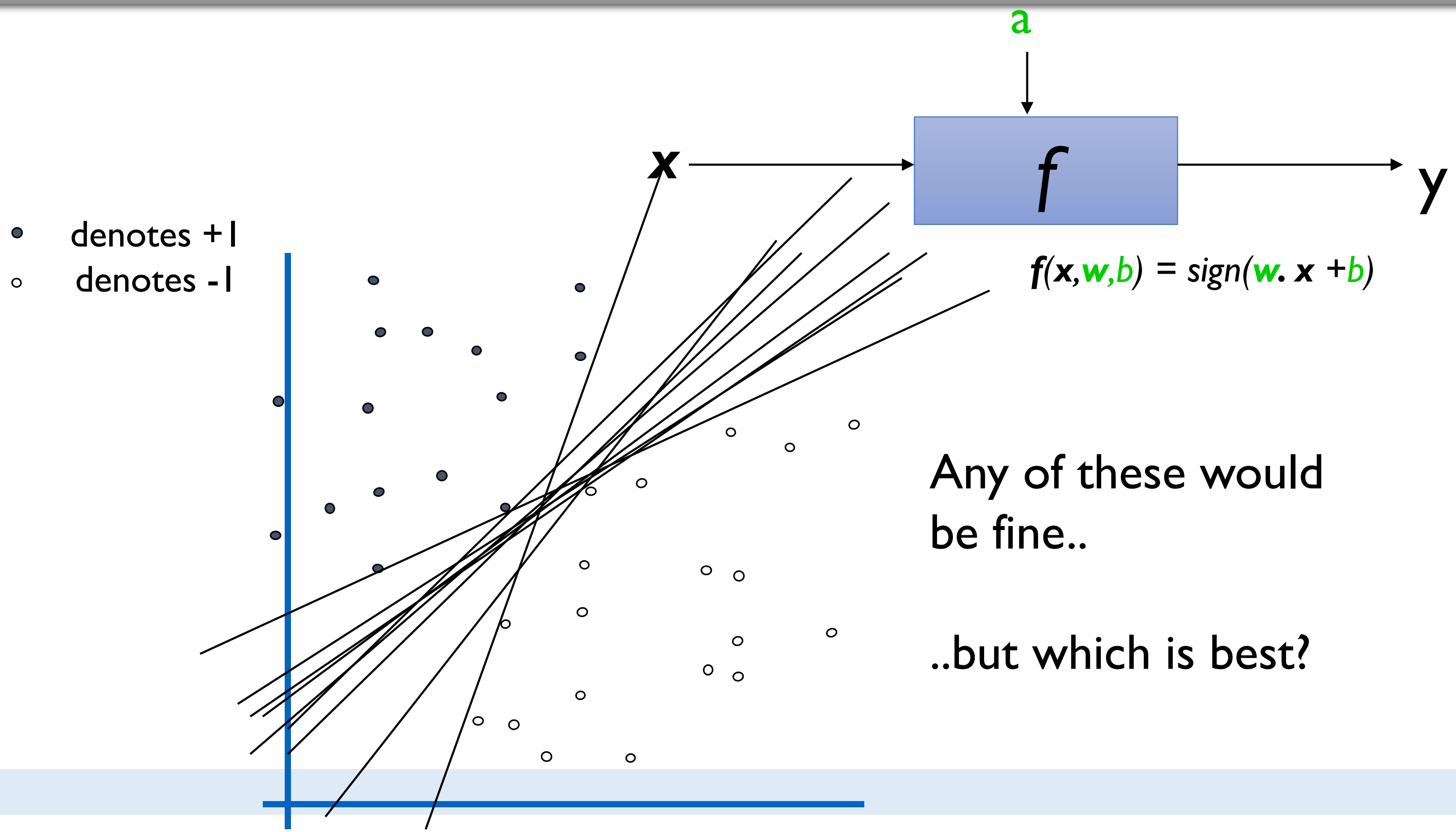
Linear Classifiers



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$

How would you classify this data?

Linear Classifiers

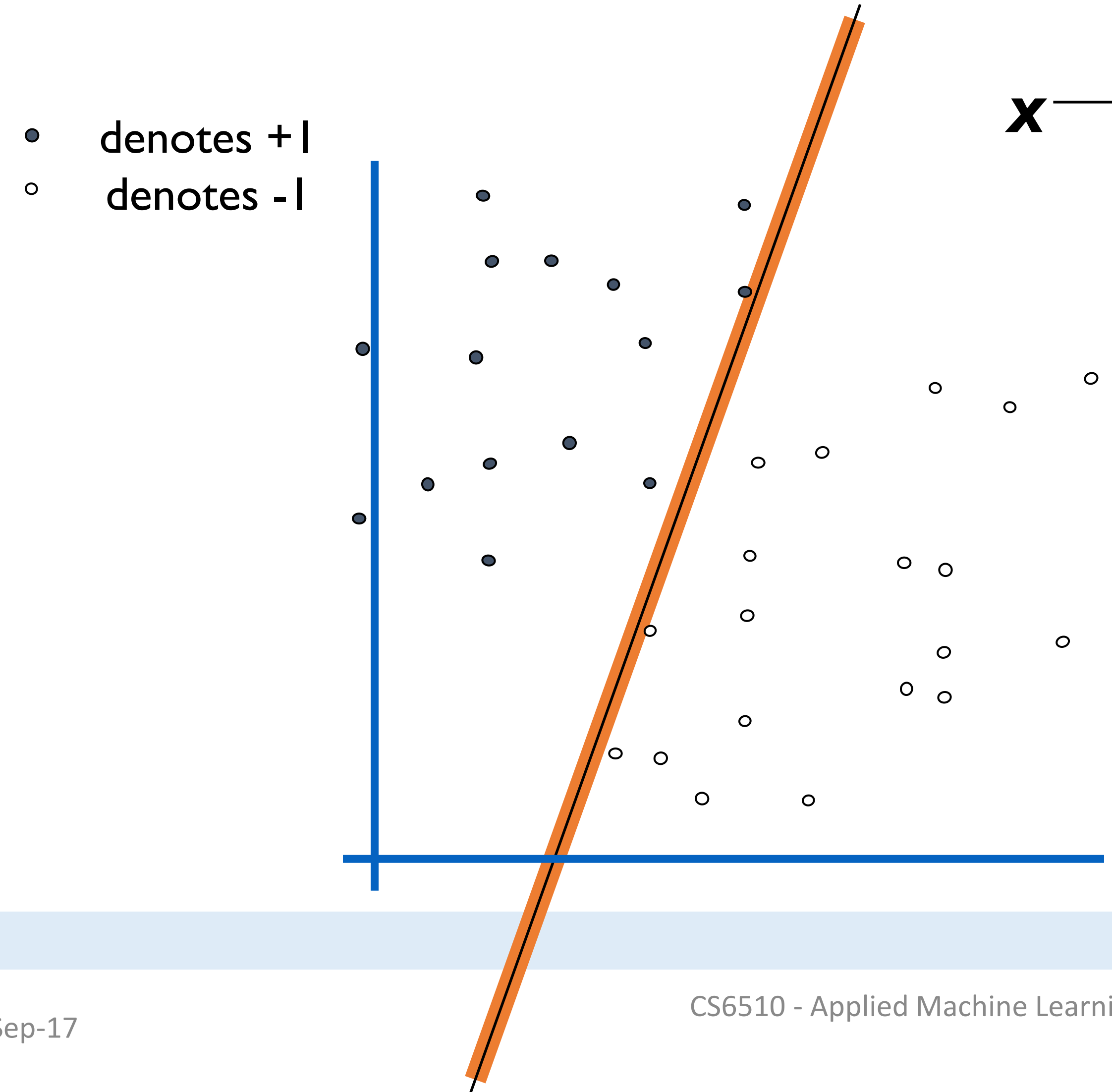


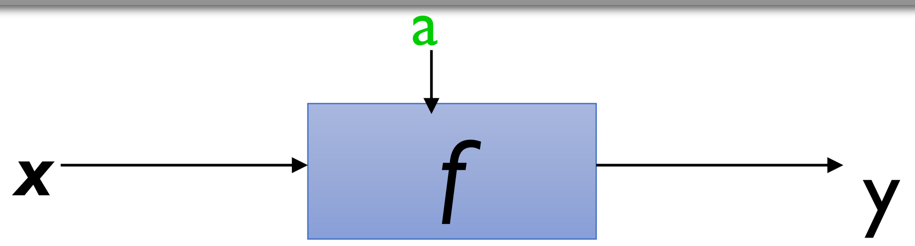
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$

Any of these would be fine..

..but which is best?

Linear Classifiers

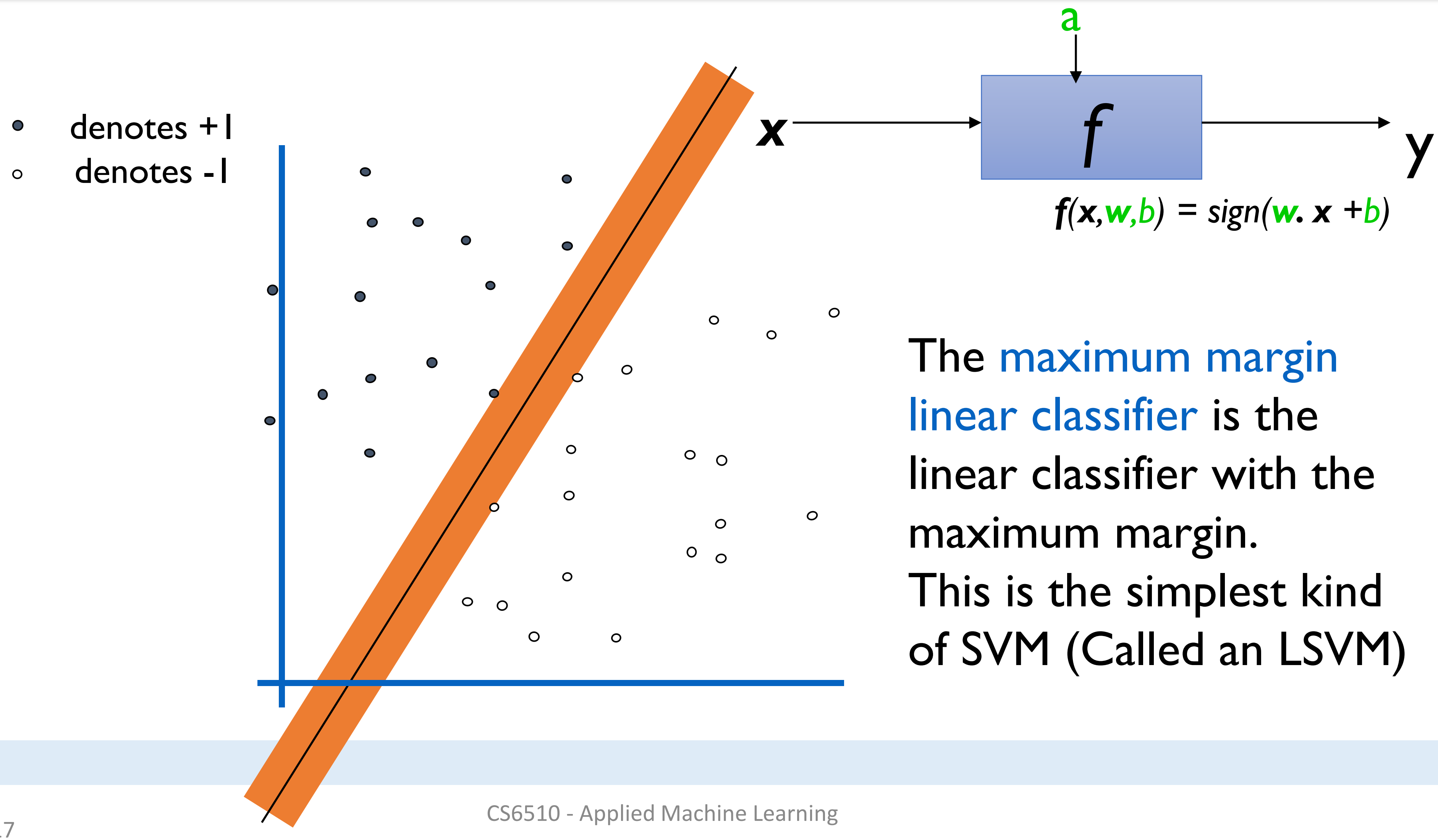




$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$

Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

Linear Classifiers

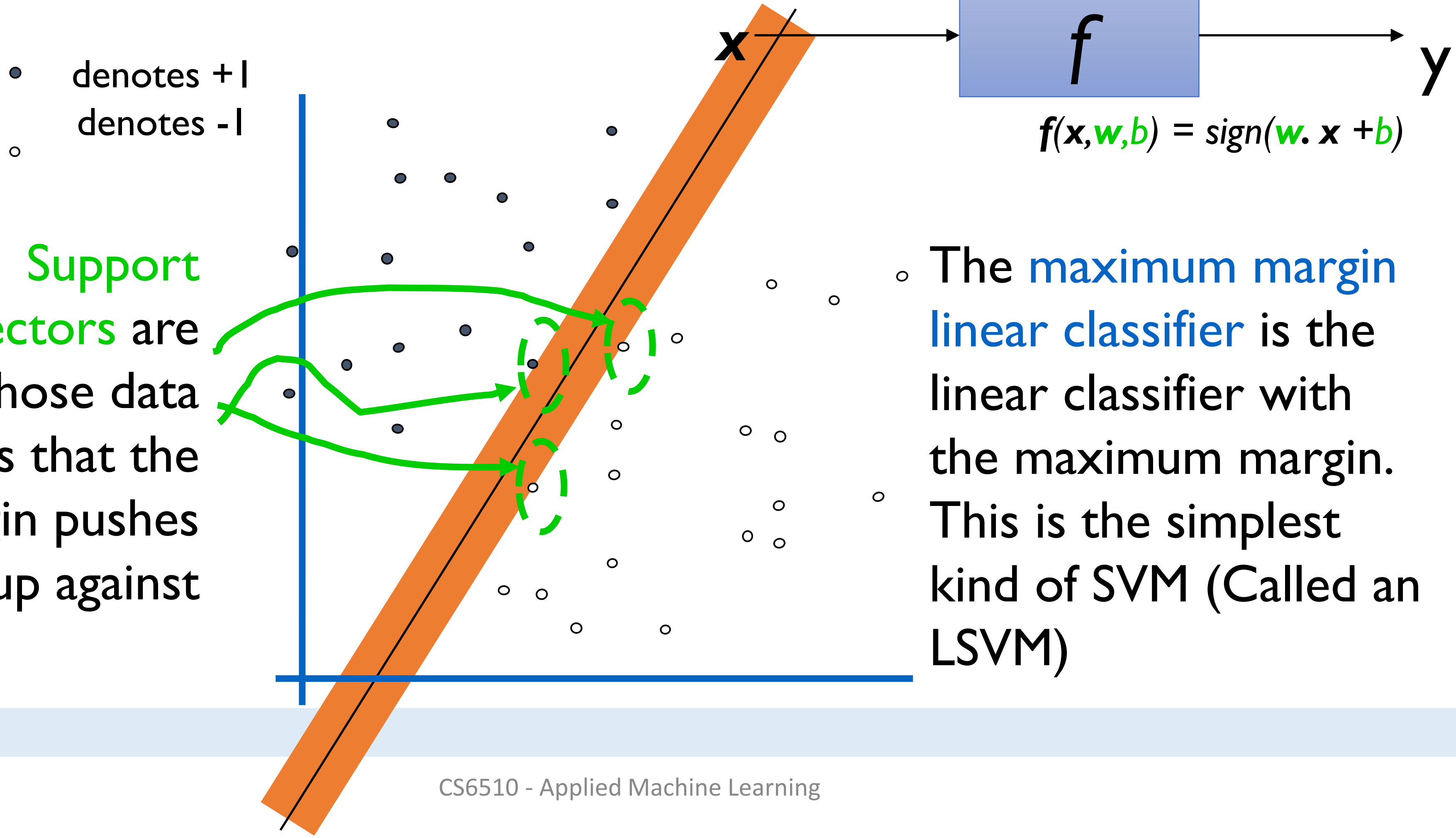


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$

The maximum margin linear classifier is the linear classifier with the maximum margin.

This is the simplest kind of SVM (Called an LSVM)

Maximum Margin Classifier



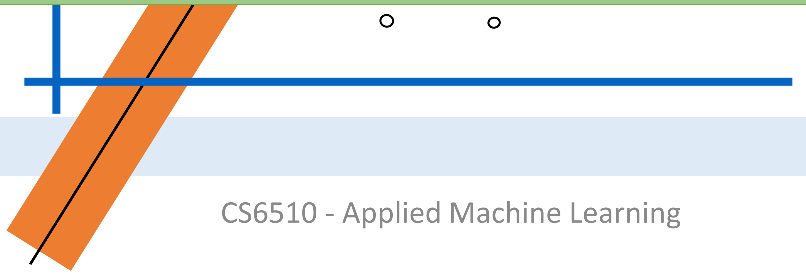
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$

Support Vectors are those data points that the margin pushes up against

The maximum margin linear classifier is the linear classifier with the maximum margin. This is the simplest kind of SVM (Called an LSVM)

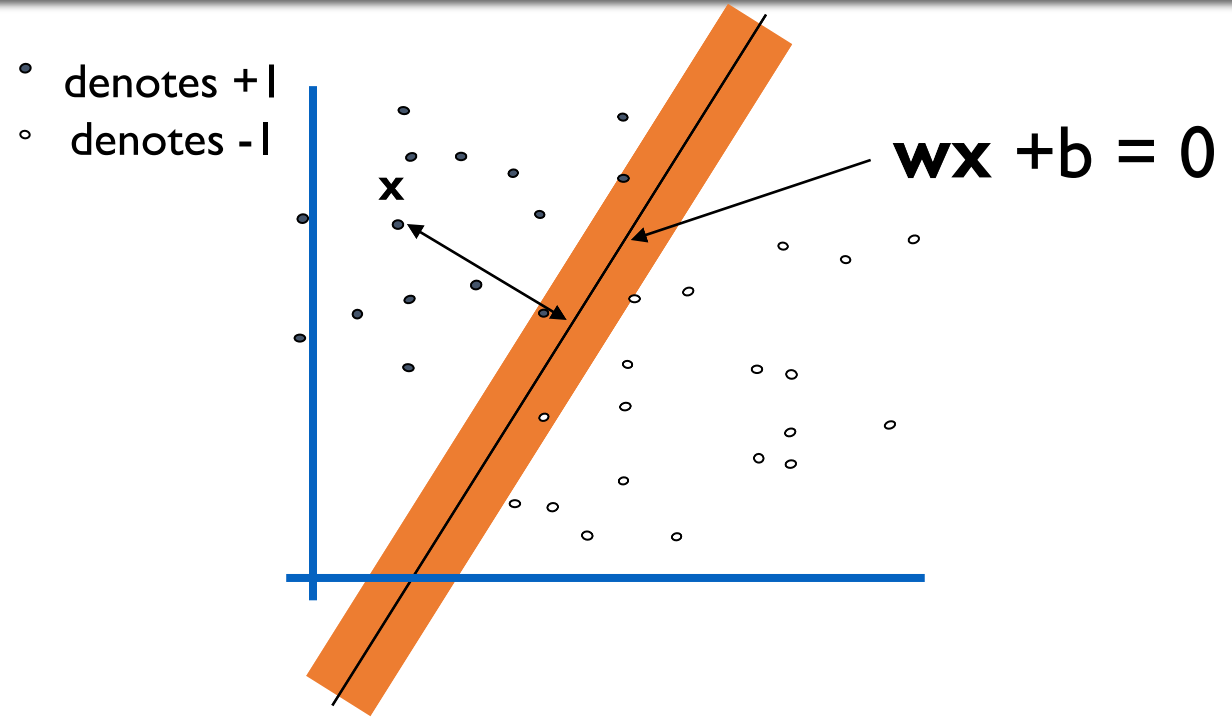
Why Maximum Margin?

- Intuitively this feels safest. If we've made a small error in the location of the boundary this gives us least chance of causing a misclassification.
- The model is immune to removal of any non-support-vector datapoints.
- There's some theory (using VC dimension) that is related to (but not the same as) the proposition that this is a good thing.
- Empirically it works very very well.



LSVM)

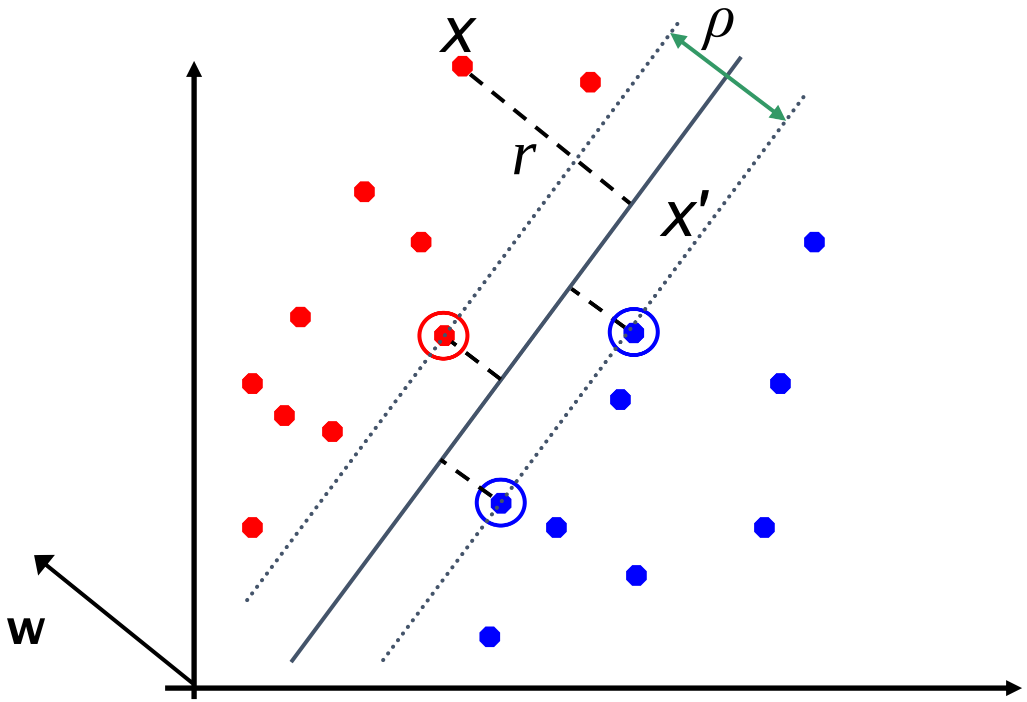
Estimating the Margin



- What is the distance expression for a point **x** to a line **wx**+b= 0?

Estimating the Margin

- Distance from example to the separator is $r = y\dfrac{\mathbf{w}^T\mathbf{x}+b}{\|\mathbf{w}\|}$

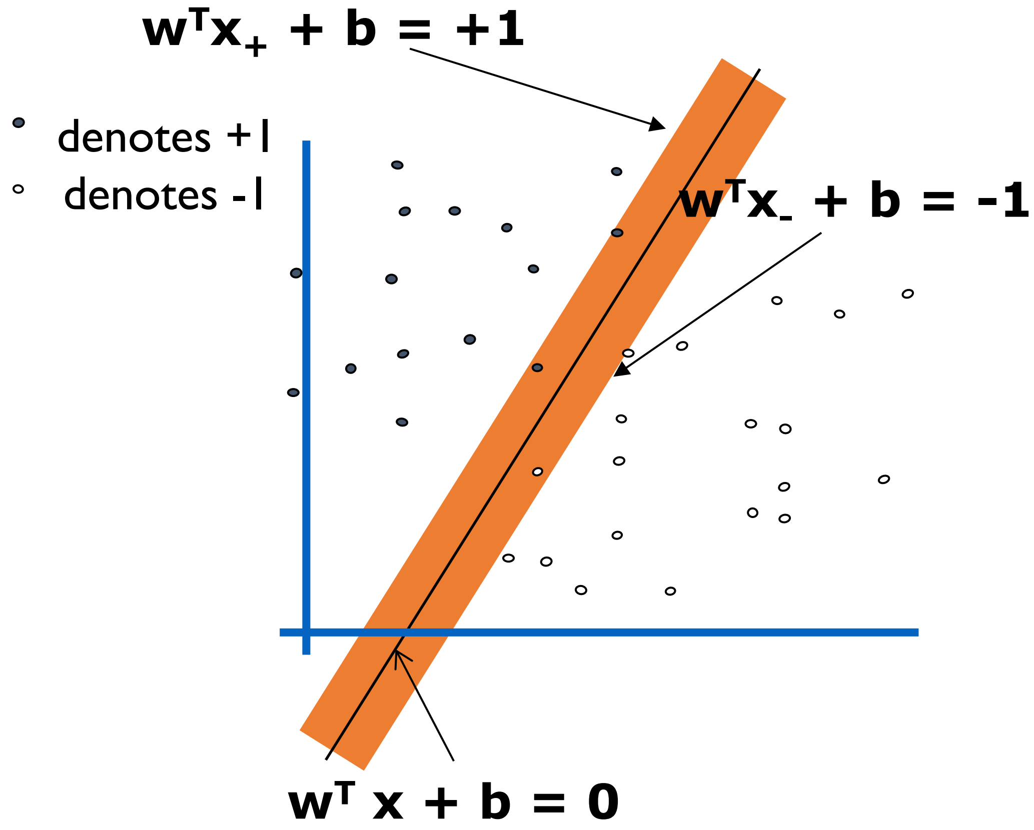


Derivation of finding *r*:

- Dotted line **x'** − **x** is perpendicular to decision boundary, so parallel to **w**.
- Unit vector is **w**/||**w**||, so line is r**w**/||**w**||.
- **x'** = **x** – yr**w**/||**w**||.
- **x'** satisfies $\mathbf{w}^T\mathbf{x}' + b = 0$.
- So $\mathbf{w}^T(\mathbf{x} - yr\mathbf{w}/||\mathbf{w}||) + b = 0$
- Recall that $||\mathbf{w}|| = \mathrm{sqrt}(\mathbf{w}^T\mathbf{w})$.
- So $\mathbf{w}^T\mathbf{x} - yr||\mathbf{w}|| + b = 0$
- So, solving for r gives: $r = y(\mathbf{w}^T\mathbf{x} + b)/||\mathbf{w}||$

Estimating the Margin

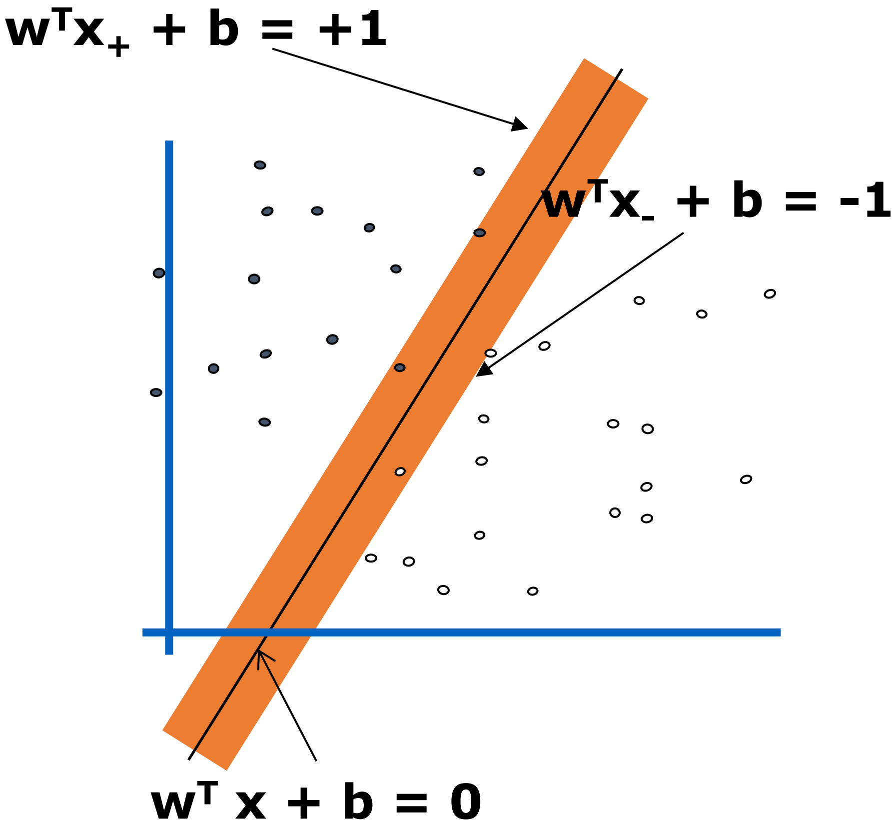
- Since $\mathbf{w}^T\mathbf{x} + b = 0$ and $c(\mathbf{w}^T\mathbf{x} + b) = 0$ define the same plane, we have the freedom to choose the normalization of $\mathbf{w}$ (i.e. c)
- Let us choose normalization such that $\mathbf{w}^T\mathbf{x}_+ + b = +1$ and $\mathbf{w}^T\mathbf{x}_- + b = -1$ for the positive and negative support vectors respectively



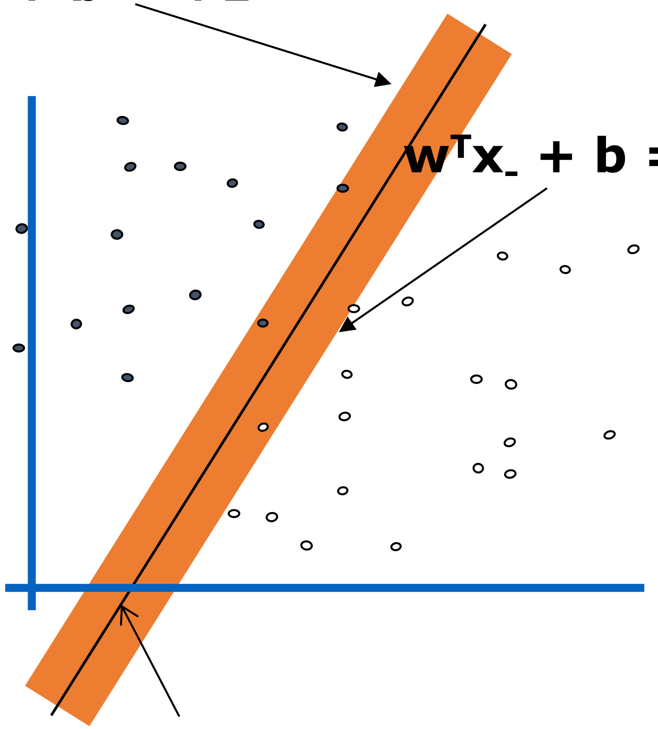
Estimating the Margin

- Since $\mathbf{w}^T\mathbf{x} + \mathbf{b} = 0$ and $c(\mathbf{w}^T\mathbf{x} + \mathbf{b}) = 0$ define the same plane, we have the freedom to choose the normalization of $\mathbf{w}$ (i.e. c)
- Let us choose normalization such that $\mathbf{w}^T\mathbf{x}_+ + \mathbf{b} = +1$ and $\mathbf{w}^T\mathbf{x}_- + \mathbf{b} = -1$ for the positive and negative support vectors respectively
- Hence, margin now is:

$$(+1) * \frac{\mathbf{w}^T\mathbf{x}_+ + b}{\|\mathbf{w}\|} + (-1) . \frac{\mathbf{w}^T\mathbf{x}_- + b}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$$



Maximizing the Margin



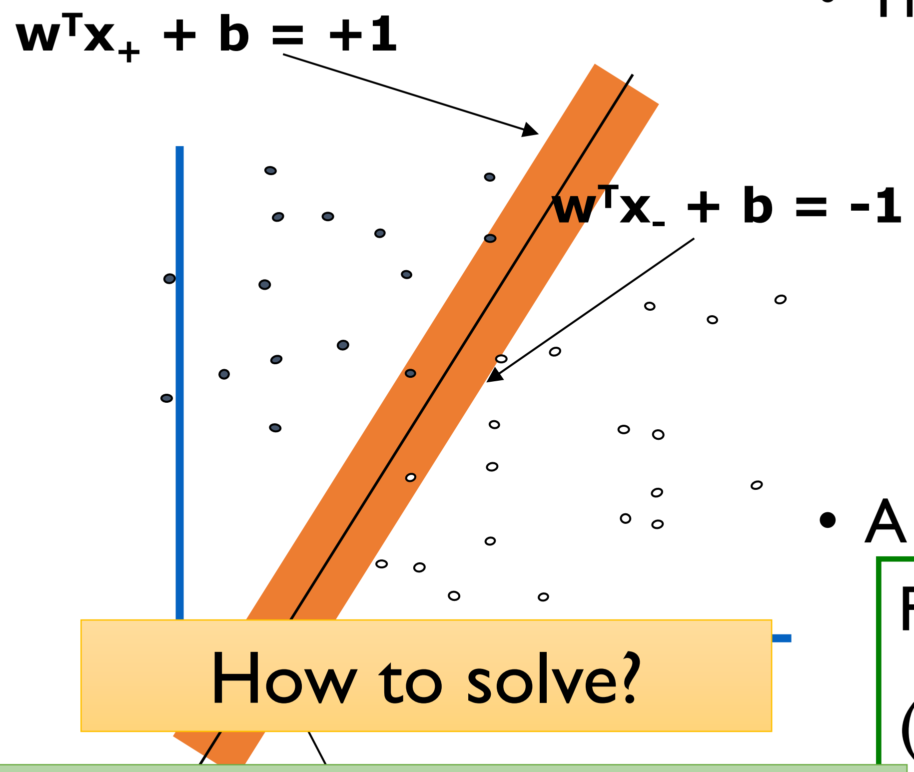
- Then we can formulate the *quadratic optimization problem:*

> Find **w** and *b* such that
>
> $r = \frac{2}{\|\mathbf{w}\|}$ is maximized; and for all $\{(\mathbf{x_i}, y_i)\}$
>
> $\mathbf{w}^T\mathbf{x_i} + b \geq 1$ if $y_i = +1$; $\mathbf{w}^T\mathbf{x_i} + b \leq -1$ if $y_i = -1$

- A better formulation (min $||\mathbf{w}||$ = max $1/ ||\mathbf{w}||$):

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

Maximizing the Margin



- Then we can formulate the *quadratic optimization problem:*

> Find **w** and b such that
>
> $r = \dfrac{2}{\|\mathbf{w}\|}$ is maximized; and for all $\{(\mathbf{x_i}, y_i)\}$
>
> $\mathbf{w}^T\mathbf{x_i} + b \geq 1$ if $y_i = +1$; $\mathbf{w}^T\mathbf{x_i} + b \leq -1$ if $y_i = -1$

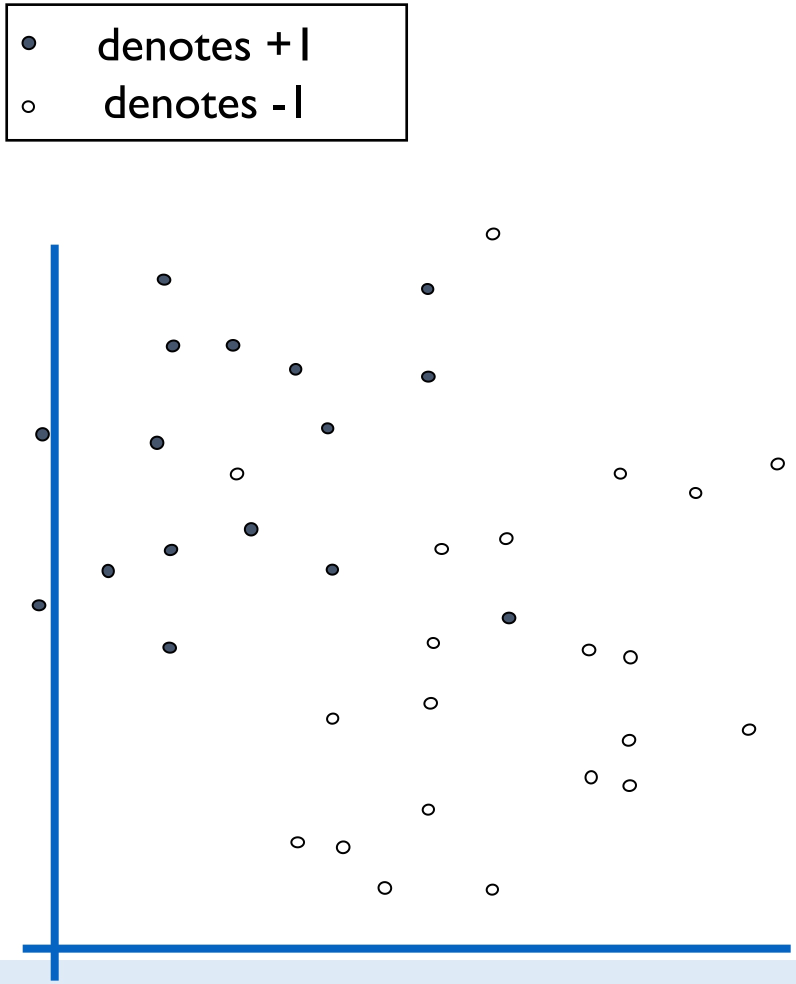
- A better formulation (min $||\mathbf{w}||$ = max $1/ ||\mathbf{w}||$):

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

How to solve?

Quadratic Programming

Non-separable Data



This is going to be a problem!
What should we do?

SVM for Noisy Data

$$\{\vec{w}^*, b^*\} = \min_{\vec{w},b} \sum_{i=1}^{d} w_i^2 + c\sum_{j=1}^{N} \varepsilon_j$$

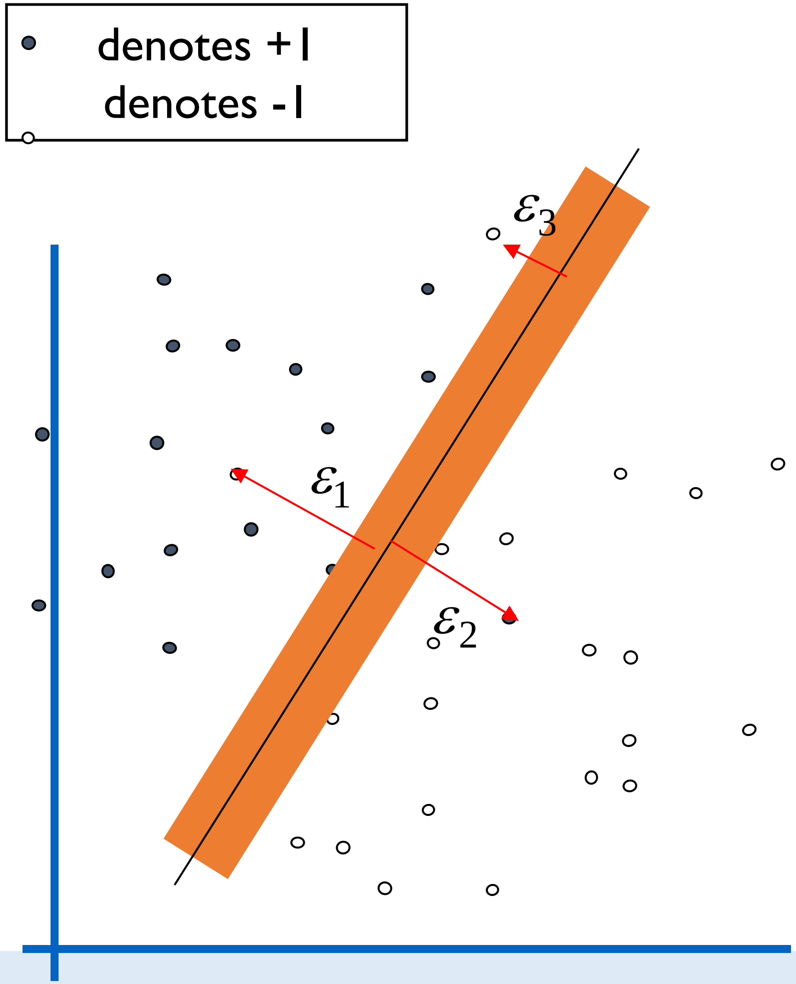
$$y_1\left(\vec{w}\cdot\vec{x}_1 + b\right) \geq 1 - \varepsilon_1, \varepsilon_1 \geq 0$$

$$y_2\left(\vec{w}\cdot\vec{x}_2 + b\right) \geq 1 - \varepsilon_2, \varepsilon_2 \geq 0$$

...

$$y_N\left(\vec{w}\cdot\vec{x}_N + b\right) \geq 1 - \varepsilon_N, \varepsilon_N \geq 0$$

Balance the trade off between margin and classification errors

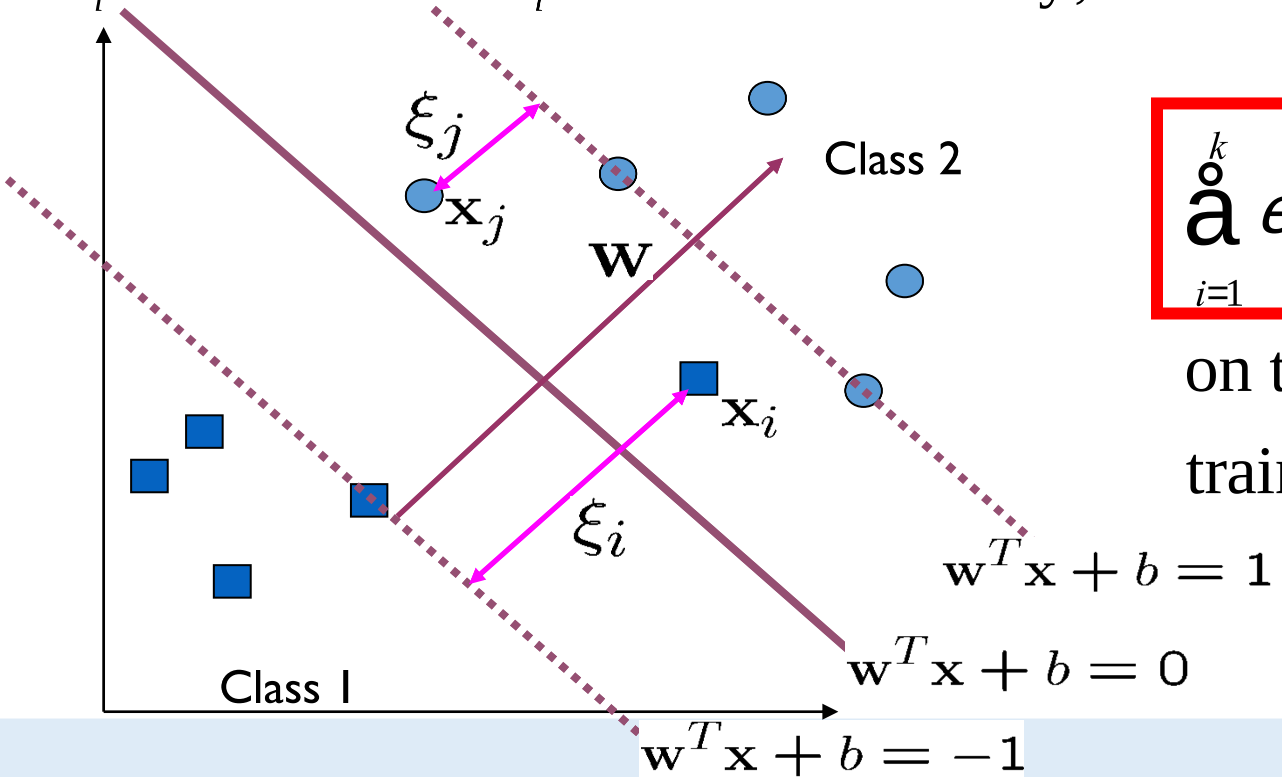


SVM for Noisy Data

$\varepsilon_i \geq 1 \quad \Leftrightarrow \quad y_i(wx_i + b) < 0,$ i.e., misclassification

$0 \prec \varepsilon_i \prec 1 \quad \Leftrightarrow \quad x_i$ is correctly classified, but lies inside the margin

$\varepsilon_i = 0 \quad \Leftrightarrow \quad x_i$ is classified correctly, and lies outside the margin



$\boxed{\sum_{i=1}^{k} \varepsilon_i}$ is an upper bound on the number of training errors.

SVM for Noisy Data

- Use the Lagrangian formulation for the optimization problem.
- Introduce a positive Lagrangian multiplier for each inequality constraint.

$$y_i(x_i \bullet w + b) - 1 + \varepsilon_i \geq 0, \text{ for all i.}$$

α_i

Lagrangian multipliers

$$\varepsilon_i \geq 0, \text{ for all i.}$$

β_i

Get the following Lagrangian: $L_p = \|w\|^2 + c\sum_i \varepsilon_i - \sum_i \alpha_i \{y_i(x_i \bullet w + b) - 1 + \varepsilon_i\} - \sum_i \beta_i \varepsilon_i$

SVM for Noisy Data

$$L_p = \|w\|^2 + c\sum_i \varepsilon_i - \sum_i \alpha_i \{y_i(x_i \bullet w + b) - 1 + \varepsilon_i\} - \sum_i \beta_i \varepsilon_i$$

$$\frac{\partial L_p}{\partial w} = 2w - \sum_i \alpha_i y_i x_i = 0 \quad \Rightarrow \quad w = \frac{1}{2}\sum_i \alpha_i y_i x_i$$

$$\frac{\partial L_p}{\partial b} = -\frac{1}{2}\sum_i \alpha_i y_i = 0 \quad \Rightarrow \quad \sum_i \alpha_i y_i = 0$$

$$\frac{\partial L_p}{\partial \varepsilon_i} = c - \beta_i - \alpha_i = 0 \quad \Rightarrow \quad c = \beta_i + \alpha_i$$

Take the derivatives of L_p with respect to w, b, and ε_i.

Karush-Kuhn-Tucker Conditions

$$0 \le \alpha_i \le c \quad \forall i$$

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i \bullet x_j)$$

Both ε_i and its multiplier β_i are not involved in the function.

SVM Lagrangian Dual

Maximize $$\sum_{k=1}^{R}\alpha_k - \frac{1}{2}\sum_{k=1}^{R}\sum_{l=1}^{R}\alpha_k\alpha_l Q_{kl}$$ where $$Q_{kl} = y_k y_l (\mathbf{x}_k \cdot \mathbf{x}_l)$$

subject to constraints:

$$0 \le \alpha_k \le c \quad \forall k \qquad \sum_{k=1}^{R}\alpha_k y_k = 0$$

Once solved, we obtain w and b using:

$$\mathbf{w} = \frac{1}{2}\sum_{k=1}^{R}\alpha_k y_k \mathbf{x}_k$$

$$y_i(x_i \bullet w + b) - 1 = 0$$

$$b = -y_i(y_i(x_i \bullet w) - 1)$$

Then classify with:

$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$

SVM Lagrangian Dual

Maximize $\sum_{k=1}^{R}\alpha_k - \frac{1}{2}\sum_{k=1}^{R}\sum_{l=1}^{R}\alpha_k\alpha_l Q_{kl}$ where $Q_{kl} = y_k y_l (\mathbf{x}_k \cdot \mathbf{x}_l)$

subject to constraints: $0 \le \alpha_k \le c \quad \forall k \qquad \sum_{k=1}^{R}\alpha_k y_k = 0$

Datapoints with $\alpha_k > 0$ will be the support vectors

Once solved, we obtain w and b using:

$$\frac{1}{2}\sum_{k=1}^{R}\alpha_k y_k \mathbf{x}_k$$

..so this sum only needs to be over the support vectors.

$$y_i(x_i \bullet w + b) - 1 = 0$$

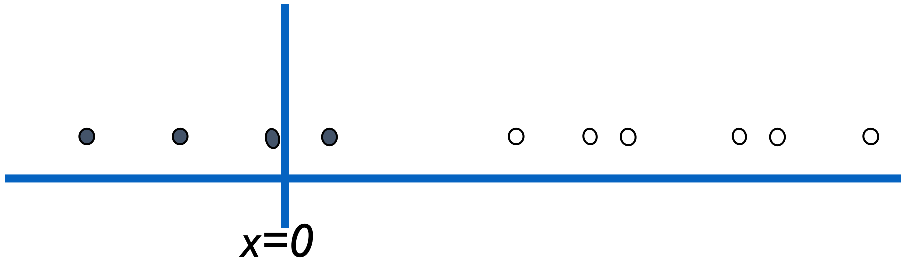
$$b = -y_i(y_i(x_i \bullet w) - 1)$$

Then classify with:

***f*(x,w,b) = *sign*(w. x + b)**

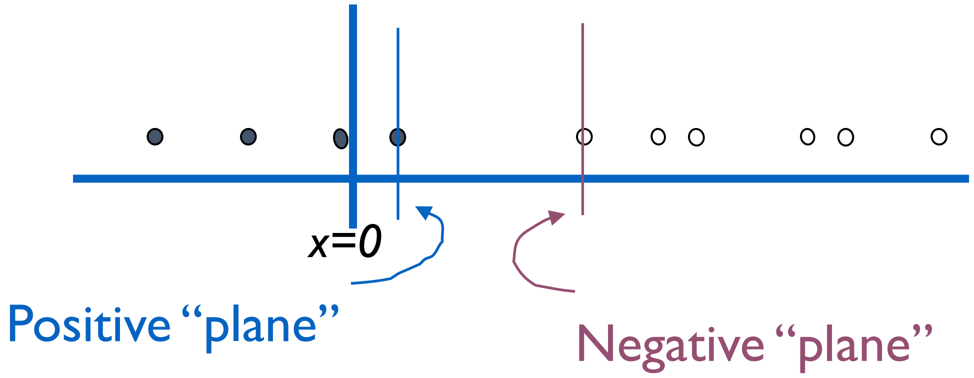
Assume we are in 1-dimension

What would SVMs do with this data?



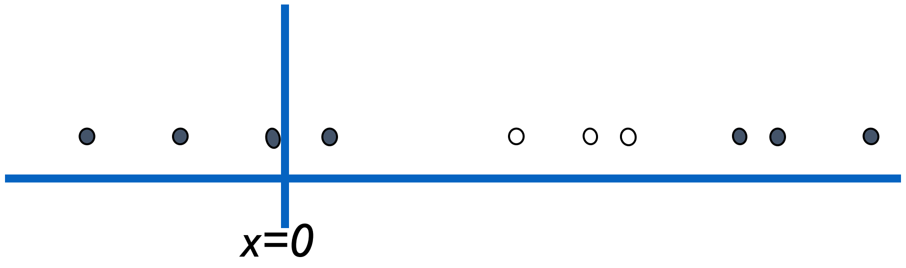
Assume we are in 1-dimension

Not a big surprise

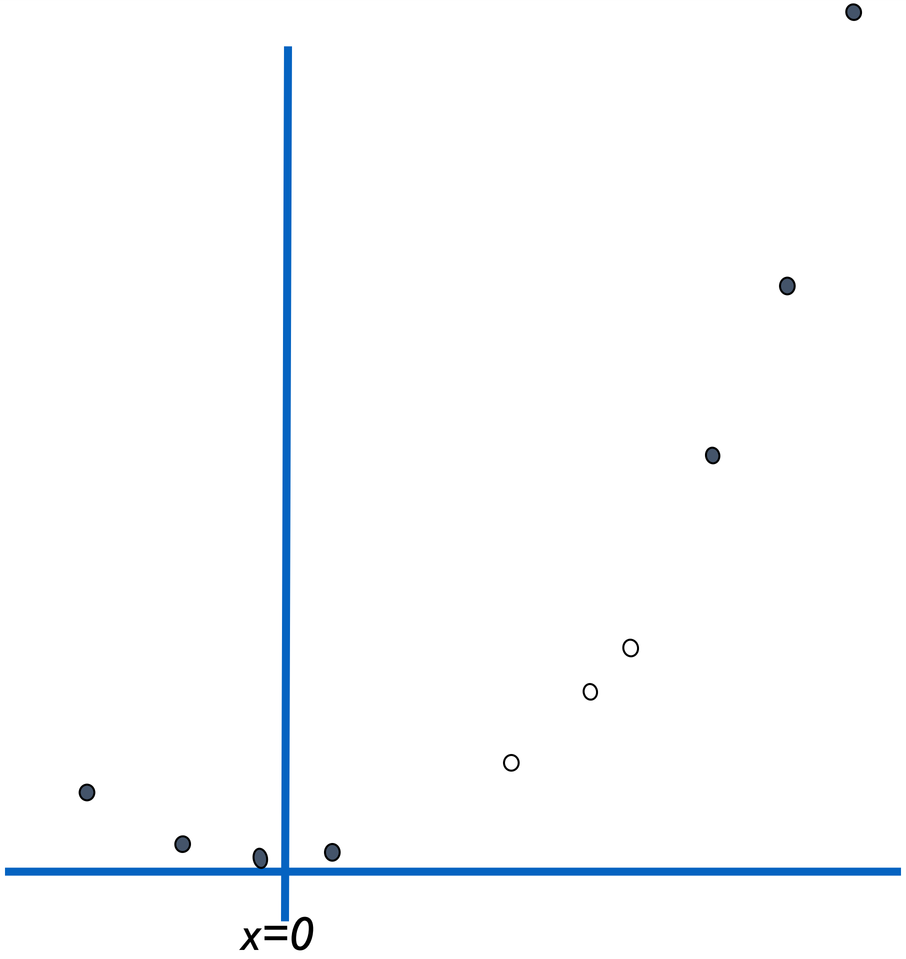


Harder 1-dimensional Dataset

What can be done about this?



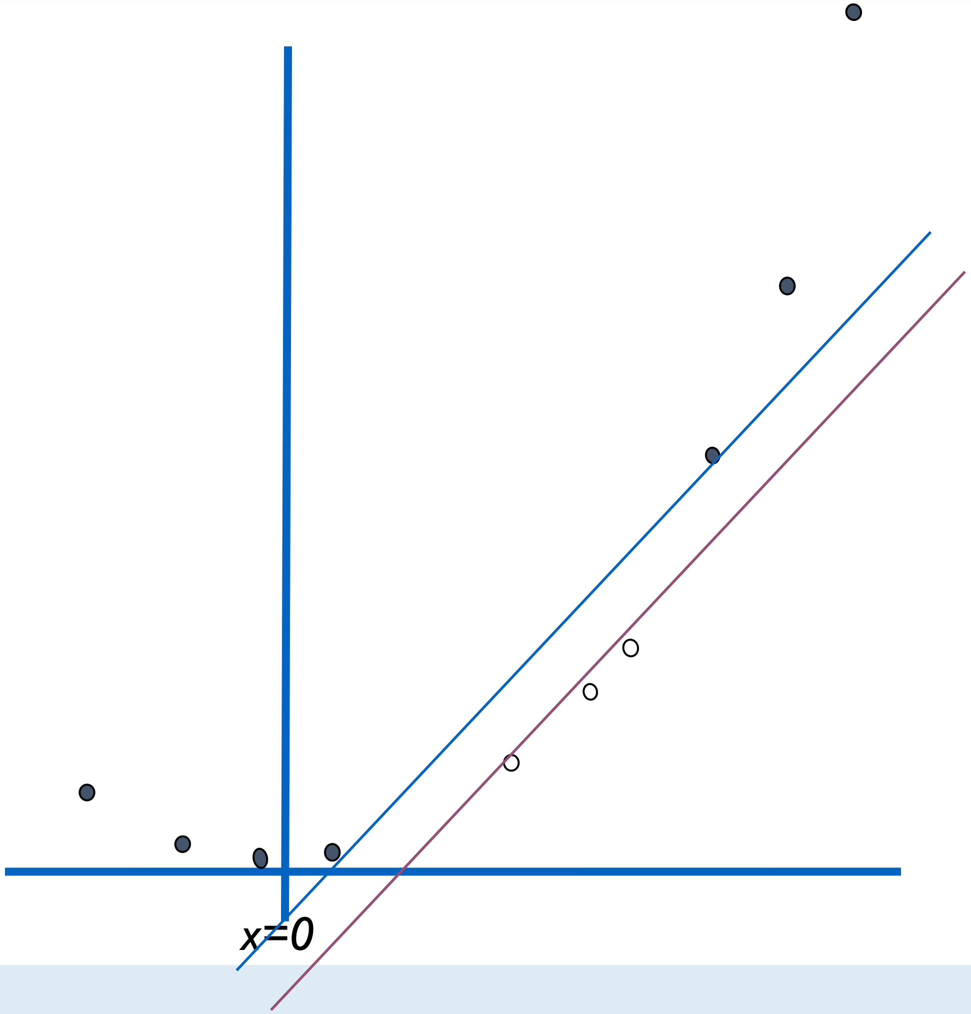
Harder 1-dimensional Dataset



Apply the following map

$$\mathbf{z}_k = (x_k, x_k^2)$$

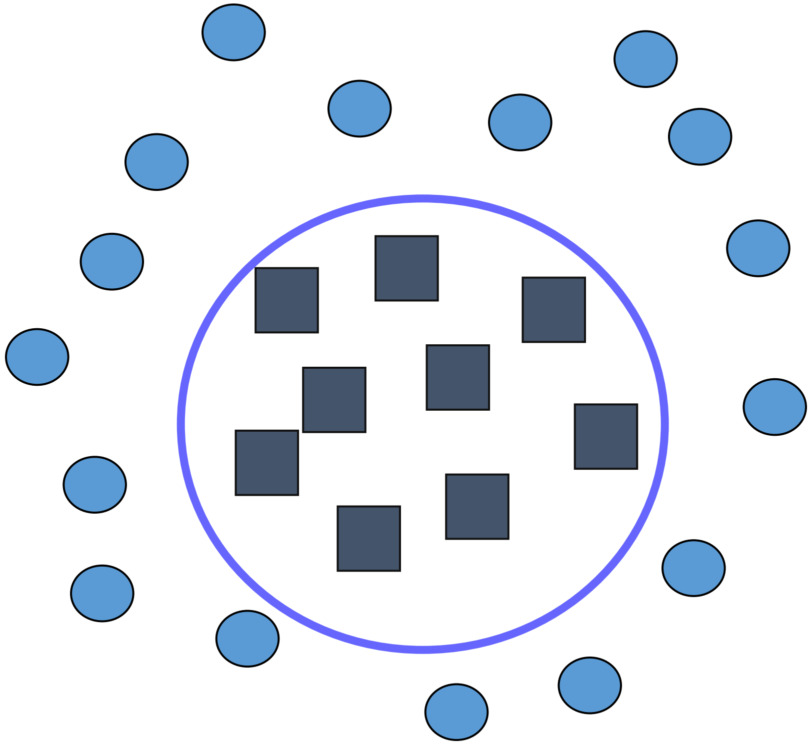
Harder 1-dimensional Dataset



Apply the following map

$$\mathbf{z}_k = (x_k, x_k^2)$$

Harder 2-dimensional Dataset



Apply the following map

$$\mathbf{z}_k = (x_k, y_k, x_k^2, y_k^2, x_k y_k)$$

Common Basis Functions

$\mathbf{z}_k$ = (polynomial terms of $\boldsymbol{x}_k$ of degree 1 to q)

$\mathbf{z}_k$ = (radial basis functions of $\boldsymbol{x}_k$)

$$\mathbf{z}_k[j] = \varphi_j(\mathbf{x}_k) = \exp\left(-\frac{|\mathbf{x}_k - \mathbf{c}_j|^2}{\sigma^2} \right)$$

$\mathbf{z}_k$ = (sigmoid functions of $\boldsymbol{x}_k$)

Recall: SVM Lagrangian Dual

Maximize $\sum_{k=1}^{R} \alpha_k - \frac{1}{2} \sum_{k=1}^{R} \sum_{l=1}^{R} \alpha_k \alpha_l Q_{kl}$ where $Q_{kl} = y_k y_l (\mathbf{x}_k \cdot \mathbf{x}_l)$

subject to constraints: $0 \le \alpha_k \le c \quad \forall k \qquad \sum_{k=1}^{R} \alpha_k y_k = 0$

Once solved, we obtain w and b using:

$$\mathbf{w} = \frac{1}{2} \sum_{k=1}^{R} \alpha_k y_k \mathbf{x}_k$$

$$y_i\left(x_i \bullet w + b\right) - 1 = 0$$

$$b = -y_i\left(y_i\left(x_i \bullet w\right) - 1\right)$$

Then classify with:

***f*(x,w,*b*) = *sign*(w. x + *b*)**

SVM QP with Basis Functions

Maximize $\sum_{k=1}^{R} \alpha_k - \frac{1}{2} \sum_{k=1}^{R} \sum_{l=1}^{R} \alpha_k \alpha_l Q_{kl}$ where $Q_{kl} = y_k y_l (\mathbf{\Phi}(\mathbf{x}_k).\mathbf{\Phi}(\mathbf{x}_l))$

subject to constraints: $0 \le \alpha_k \le C \quad \forall k$ $\quad \sum_{k=1}^{R} \alpha_k y_k = 0$

Then compute:

$$\mathbf{w} = \sum_{k \text{ s.t. } \alpha_k > 0} \alpha_k y_k \mathbf{\Phi}(\mathbf{x}_k)$$

Then classify with:

$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{\Phi}(\mathbf{x}) + b)$

Most important change:

$\mathbf{x} \rightarrow \mathbf{\Phi}(\mathbf{x})$

SVM QP with Basis Functions

Maximize $\sum_{k=1}^{R} \alpha_k - \frac{1}{2} \sum_{k=1}^{R} \sum_{l=1}^{R} \alpha_k \alpha_l Q_{kl}$ where $Q_{kl} = y_k y_l (\mathbf{\Phi}(\mathbf{x}_k).\mathbf{\Phi}(\mathbf{x}_l))$

subject to constraints: $0 \leq \alpha_k \leq C$

Then compute:

$$\mathbf{w} = \sum_{k \text{ s.t. } \alpha_k > 0} \alpha_k y_k \mathbf{\Phi}(\mathbf{x}_k)$$

We must do $R^2/2$ dot products to get this matrix ready

Assuming a quadratic polynomial kernel, each dot product requires $m^2/2$ additions and multiplications (where m is the dimension of x)

The whole thing costs $R^2 m^2 /4$.

Quadratic Dot Products

$$\Phi(\mathbf{a})\bullet\Phi(\mathbf{b}) = \begin{pmatrix} 1 \\ \sqrt{2}a_1 \\ \sqrt{2}a_2 \\ \vdots \\ \sqrt{2}a_m \\ a_1^2 \\ a_2^2 \\ \vdots \\ a_m^2 \\ \sqrt{2}a_1a_2 \\ \sqrt{2}a_1a_3 \\ \vdots \\ \sqrt{2}a_1a_m \\ \sqrt{2}a_2a_3 \\ \vdots \\ \sqrt{2}a_1a_m \\ \vdots \\ \sqrt{2}a_{m-1}a_m \end{pmatrix} \bullet \begin{pmatrix} 1 \\ \sqrt{2}b_1 \\ \sqrt{2}b_2 \\ \vdots \\ \sqrt{2}b_m \\ b_1^2 \\ b_2^2 \\ \vdots \\ b_m^2 \\ \sqrt{2}b_1b_2 \\ \sqrt{2}b_1b_3 \\ \vdots \\ \sqrt{2}b_1b_m \\ \sqrt{2}b_2b_3 \\ \vdots \\ \sqrt{2}b_1b_m \\ \vdots \\ \sqrt{2}b_{m-1}b_m \end{pmatrix}$$

$$1 + \sum_{i=1}^{m} 2a_ib_i + \sum_{i=1}^{m} a_i^2b_i^2 + \sum_{i=1}^{m}\sum_{j=i+1}^{m} 2a_ia_jb_ib_j$$

Quadratic Dot Products

$$\Phi(\mathbf{a}) \bullet \Phi(\mathbf{b}) =$$

$$1 + 2\sum_{i=1}^{m} a_i b_i + \sum_{i=1}^{m} a_i^2 b_i^2 + \sum_{i=1}^{m} \sum_{j=i+1}^{m} 2 a_i a_j b_i b_j$$

Just out of interest, let's look at another function of ***a*** and ***b***:

$$(\mathbf{a}.\mathbf{b} + 1)^2$$

$$= (\mathbf{a}.\mathbf{b})^2 + 2\mathbf{a}.\mathbf{b} + 1$$

$$= \left(\sum_{i=1}^{m} a_i b_i \right)^2 + 2\sum_{i=1}^{m} a_i b_i + 1$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{m} a_i b_i a_j b_j + 2\sum_{i=1}^{m} a_i b_i + 1$$

$$= \sum_{i=1}^{m} (a_i b_i)^2 + 2\sum_{i=1}^{m} \sum_{j=i+1}^{m} a_i b_i a_j b_j + 2\sum_{i=1}^{m} a_i b_i + 1$$

Quadratic Dot Products

They're the same!
And this is only O(m) to compute!

$$\Phi(\mathbf{a}) \bullet \Phi(\mathbf{b}) =$$

$$1 + 2\sum_{i=1}^{m} a_i b_i + \sum_{i=1}^{m} a_i^2 b_i^2 + \sum_{i=1}^{m} \sum_{j=i+1}^{m} 2 a_i a_j b_i b_j$$

Just out of interest, let's look at another function of ***a*** and ***b***:

$$(\mathbf{a}.\mathbf{b} + 1)^2$$

$$= (\mathbf{a}.\mathbf{b})^2 + 2\mathbf{a}.\mathbf{b} + 1$$

$$= \left(\sum_{i=1}^{m} a_i b_i \right)^2 + 2\sum_{i=1}^{m} a_i b_i + 1$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{m} a_i b_i a_j b_j + 2\sum_{i=1}^{m} a_i b_i + 1$$

$$= \sum_{i=1}^{m} (a_i b_i)^2 + 2\sum_{i=1}^{m} \sum_{j=i+1}^{m} a_i b_i a_j b_j + 2\sum_{i=1}^{m} a_i b_i + 1$$

SVM QP with Basis Functions

Maximize $\sum_{k=1}^{R} \alpha_k - \frac{1}{2}\sum_{k=1}^{R}\sum_{l=1}^{R} \alpha_k \alpha_l Q_{kl}$ where $Q_{kl} = y_k y_l (\mathbf{\Phi}(\mathbf{x}_k).\mathbf{\Phi}(\mathbf{x}_l))$

subject to constraints: $0 \le \alpha_k \le C$

We must do $R^2/2$ dot products to get this matrix ready

Now, each dot product now only requires m additions and multiplications

Then compute:

$$\mathbf{w} = \sum_{k \text{ s.t. } \alpha_k > 0} \alpha_k y_k \mathbf{\Phi}(\mathbf{x}_k)$$

Most important change:

$\boldsymbol{x} \rightarrow \mathbf{\Phi}(\boldsymbol{x})$

Higher-Order Polynomials

Poly-nomial	f(x)	Cost to build Q_{kl} matrix traditionally	Cost if 100 dimensions	f(a).f(b)	Cost to build Q_{kl} matrix sneakily	Cost if 100 dimensions
Quadratic	All $m^2/2$ terms up to degree 2	$m^2 R^2 /4$	2,500 R^2	$(a.b+1)^2$	$m R^2 / 2$	50 R^2
Cubic	All $m^3/6$ terms up to degree 3	$m^3 R^2 /12$	83,000 R^2	$(a.b+1)^3$	$m R^2 / 2$	50 R^2
Quartic	All $m^4/24$ terms up to degree 4	$m^4 R^2 /48$	1,960,000 R^2	$(a.b+1)^4$	$m R^2 / 2$	50 R^2

SVM QP with Basis Functions

Maximize $\sum_{k=1}^{R}\alpha_k - \frac{1}{2}\sum_{k=1}^{R}\sum_{l=1}^{R}\alpha_k\alpha_l Q_{kl}$ where $Q_{kl} = y_k y_l K(\mathbf{x}_k, \mathbf{x}_l)$

Kernel gram matrix

Subject to these constraints: $0 \le \alpha_k \le C \quad \forall k \qquad \sum_{k=1}^{R}\alpha_k y_k = 0$

Then define:

$$\mathbf{w} = \sum_{k \text{ s.t. } \alpha_k > 0} \alpha_k y_k \mathbf{\Phi}(\mathbf{x}_k)$$

Then classify with:

$f(\mathbf{x}, \mathbf{w}, b) = \mathrm{sign}(K(\mathbf{w}, \mathbf{x}) - b)$

Most important change:

$$\mathbf{\Phi}(\mathbf{x}_k).\mathbf{\Phi}(\mathbf{x}_l) \rightarrow K(\mathbf{x}_k, \mathbf{x}_l)$$

SVM Kernel Functions

- $K(\mathbf{a},\mathbf{b})=(\mathbf{a} \cdot \mathbf{b} + 1)^d$ is an example of a **kernel function** in SVM
- Beyond polynomials, there are other high-dimensional kernel functions such as:
  – Radial-Basis-style Kernel Function:

$$K(\mathbf{a},\mathbf{b}) = \exp\left(-\frac{(\mathbf{a}-\mathbf{b})^2}{2\sigma^2}\right)$$

  – Sigmoidal function

Kernel Tricks

- Replacing dot product with a kernel function
- Not all functions are kernel functions
  - Need to be decomposable: $K(a,b) = \phi(a) \cdot \phi(b)$
- **Mercer's condition** To expand Kernel function K(x,y) into a dot product, i.e. $K(x,y)=\Phi(x)\cdot\Phi(y)$, K(x, y) has to be positive semi-definite function, i.e., for any function f(x) whose $\int f^2(x)dx$ is finite, the following inequality holds:

$$\int dxdyf(x)K(x,y)f(y) \geq 0$$

How to choose a kernel function?

- Not easy! Remember – this depends on your data geometry
- If linear works, go with it
- RBF kernels are considered good in general, especially for images (and other smooth functions/data)
- For discrete data, chi-square kernel preferred of late (especially for histogram data)
- You can also do Multiple Kernel Learning (we will not cover in lectures, but included in readings)
- Still not sure? Use cross-validation to select a kernel function from some basic options

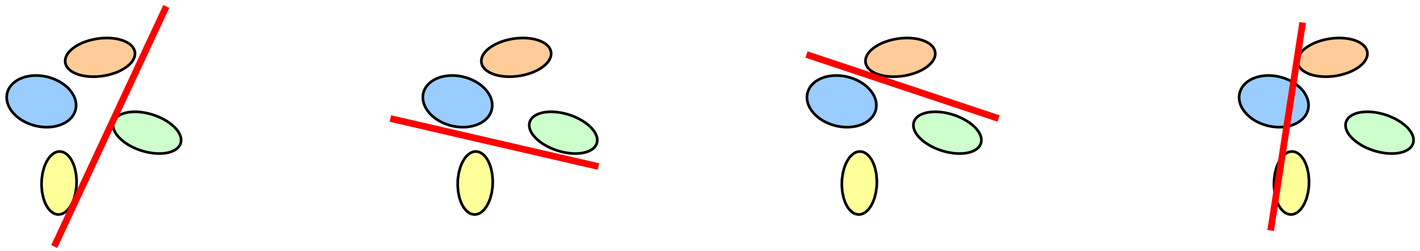
An excellent resource: http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications/

Back to SVMs: Multi-class Classification

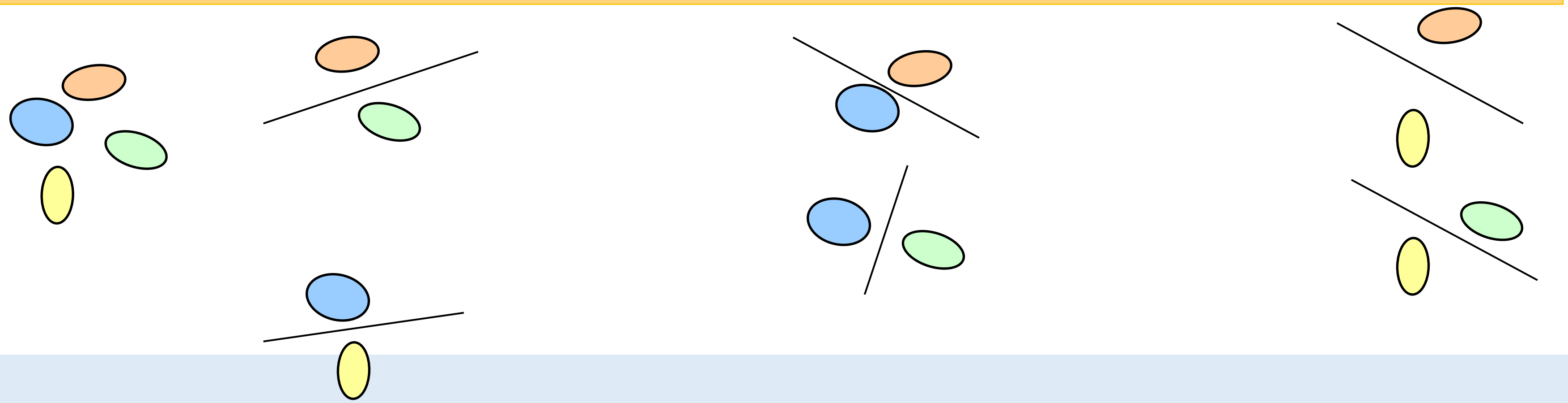
- SVMs can only handle two-class outputs.
- What can be done?
- Answer: with output arity N, learn N SVM's
  - SVM 1 learns "Output==1" vs "Output != 1"
  - SVM 2 learns "Output==2" vs "Output != 2"
  - :
  - SVM N learns "Output==N" vs "Output != N"
- Then to predict the output for a new input, just predict with each SVM and find out which one puts the prediction the furthest into the positive region.
- Other approaches
  - Pair-wise SVM, Tree-structured SVM

Multi-class Classification using SVM

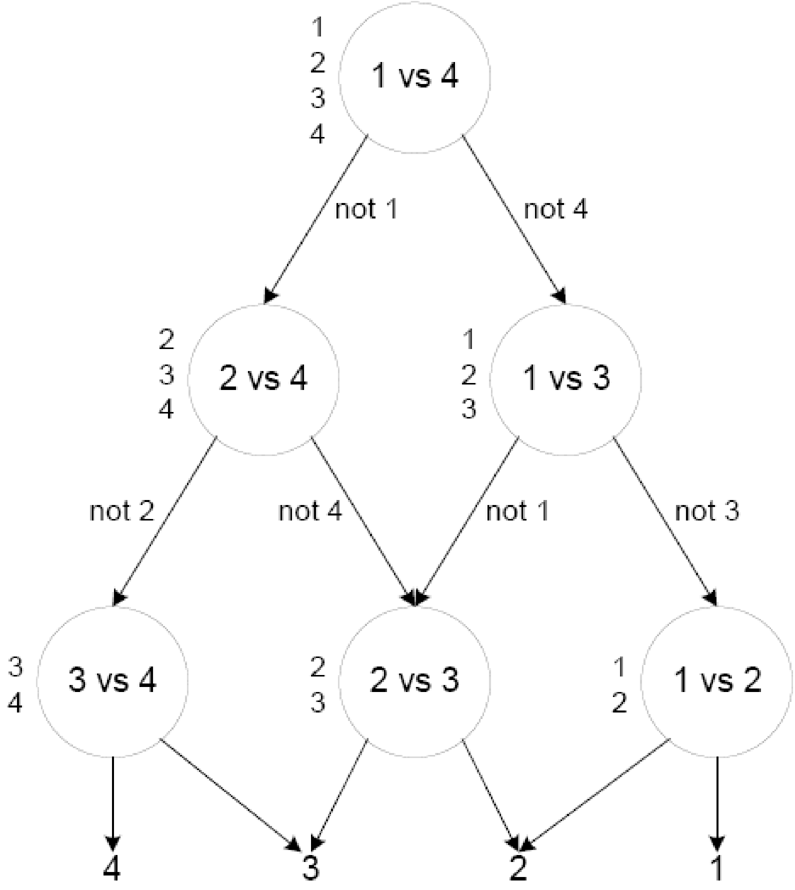
One- versus-all



One- versus-one



Tree-Structured SVM



Also called DAG-SVM (DAG = Directed Acyclic Graph)

Readings

- "Introduction to Machine Learning" by Ethem Alpaydin, 2nd edition, Chapters 3 (3.1-3.4), Chapter 13 (13.1-13.9)
- For kernel functions:
  - http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications/