Support Vector Machines

4 Jun 2019

Vineeth N Balasubramanian

आई आई टी हैदराबाद
IIT Hyderabad

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

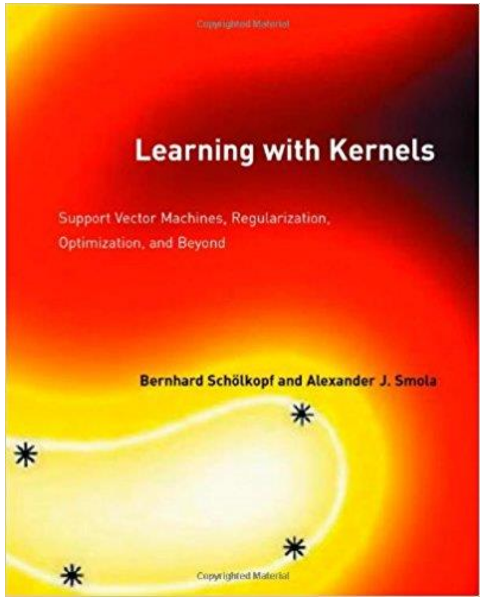
- A discriminative classifier
  - Non-parametric, Inductive
- SVM is inspired from statistical learning theory
- SVM was developed in 1992 by Vapnik, Guyon and Boser
- SVM became popular because of its success in handwritten digit recognition
- Has been one of the go-to methods in machine learning since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

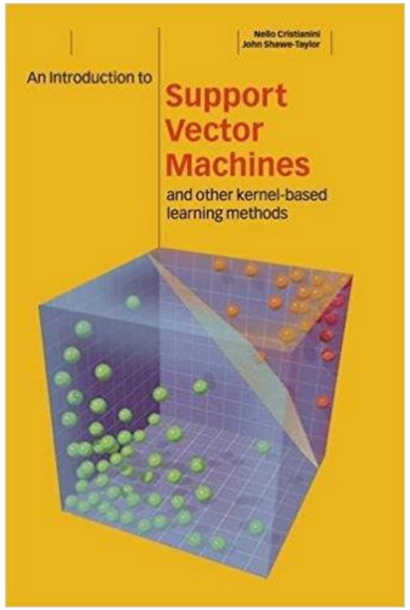
Papers that introduced SVM in its current form

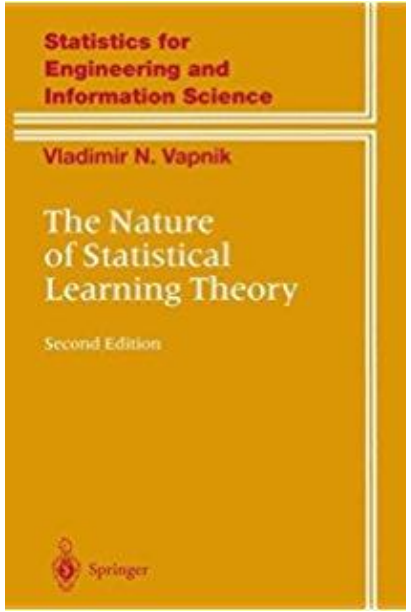
- Boser, B. E.; Guyon, I. M.; Vapnik, V. N. (1992). "A training algorithm for optimal margin classifiers". Proceedings of the fifth annual workshop on Computational learning theory – COLT '92.
- Cortes, C.; Vapnik, V. (1995). "Support-vector networks". Machine Learning. 20 (3): 273–297.

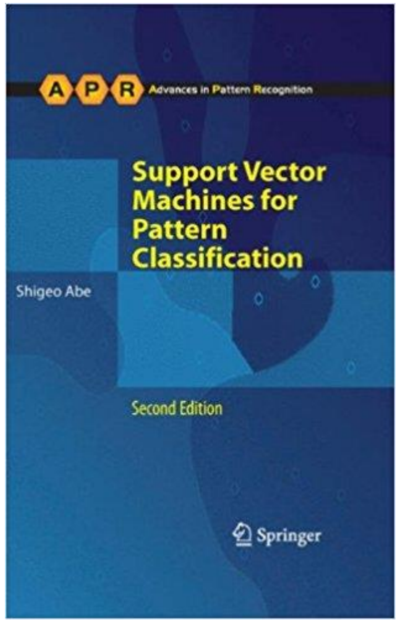
SVM: Overview and History

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

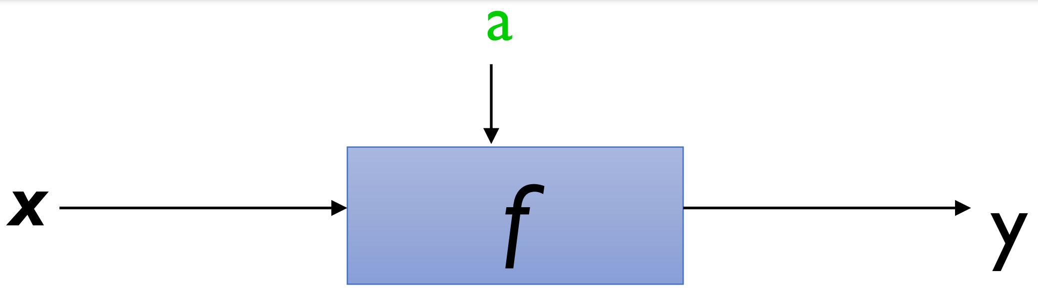




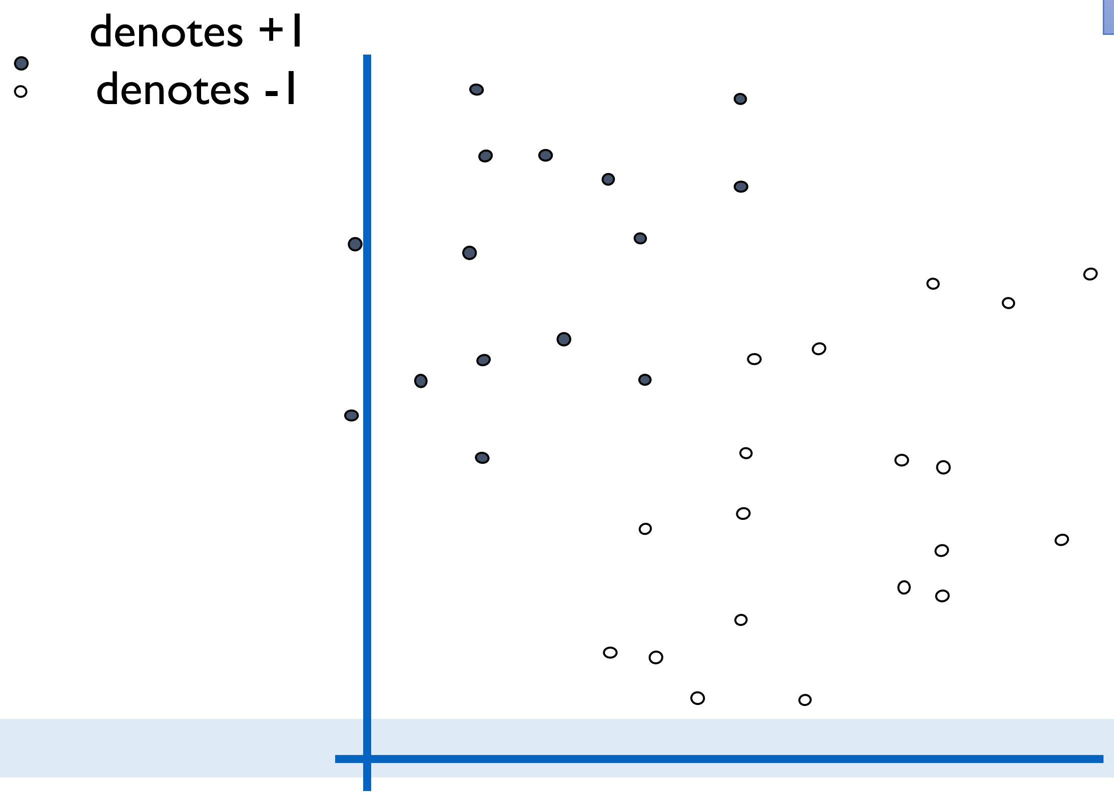




Linear Classifiers

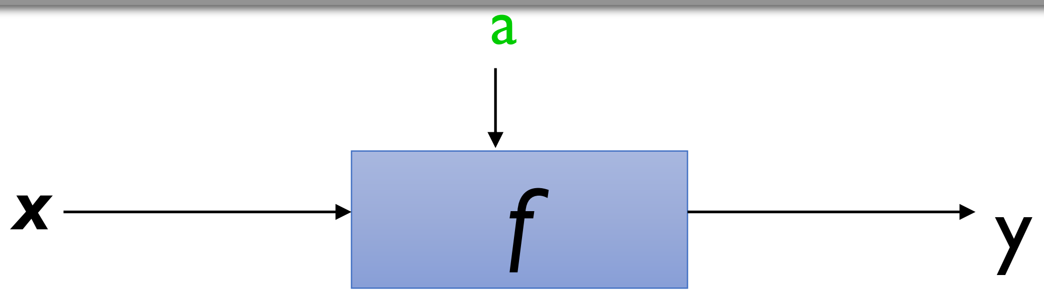


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$

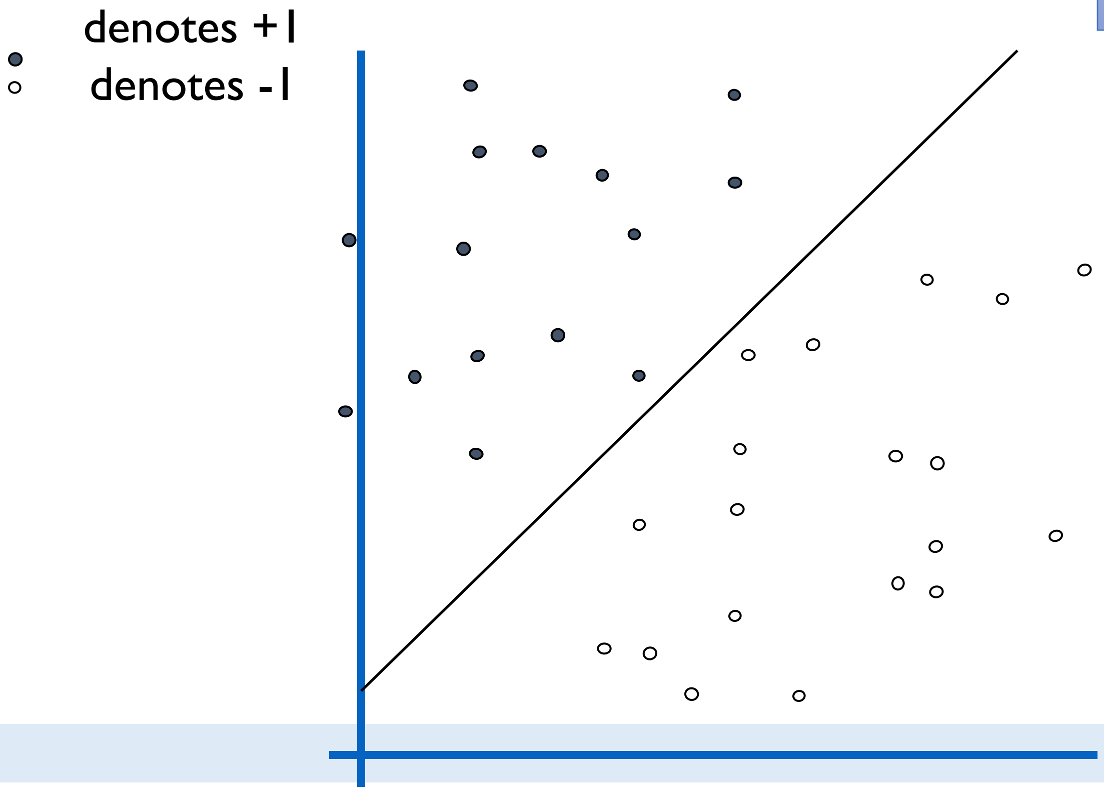


How would you classify this data?

Linear Classifiers

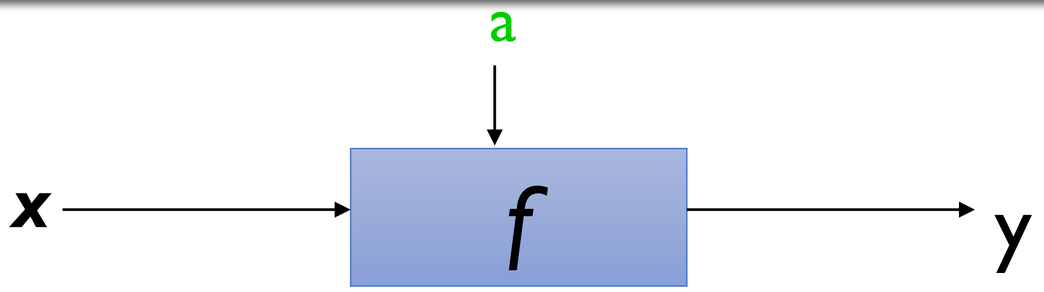


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w} \cdot \mathbf{x} + b)$

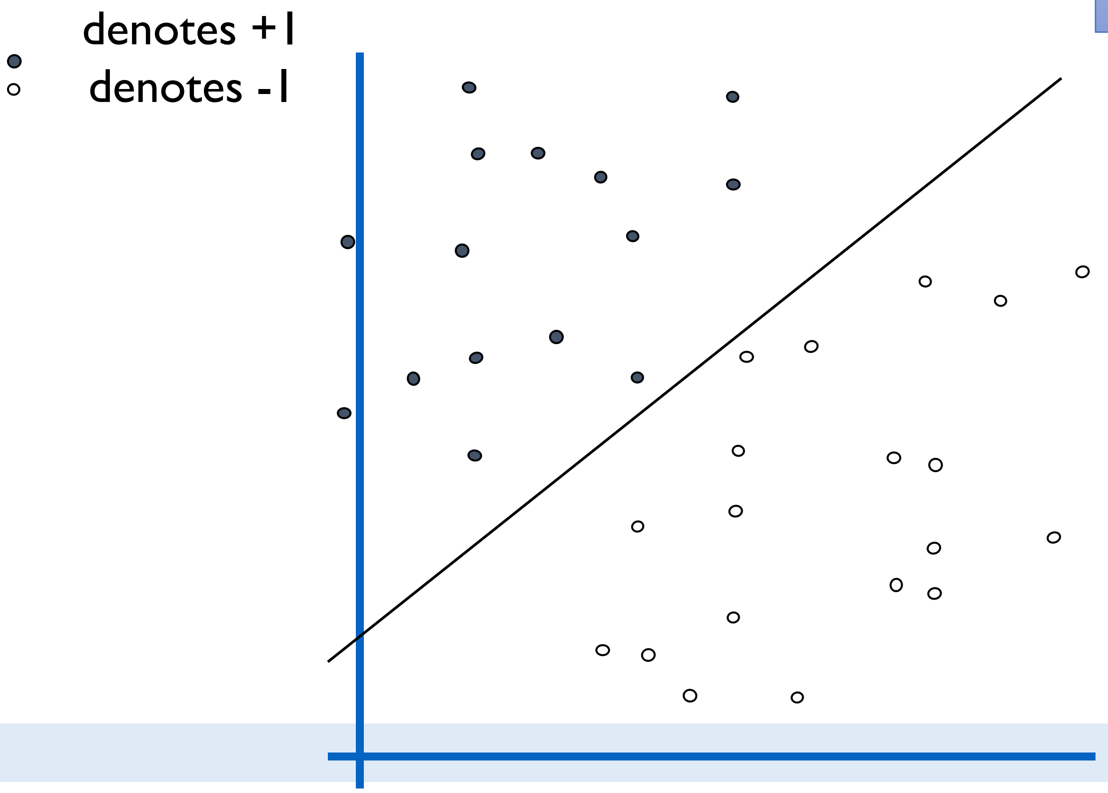


How would you classify this data?

Linear Classifiers

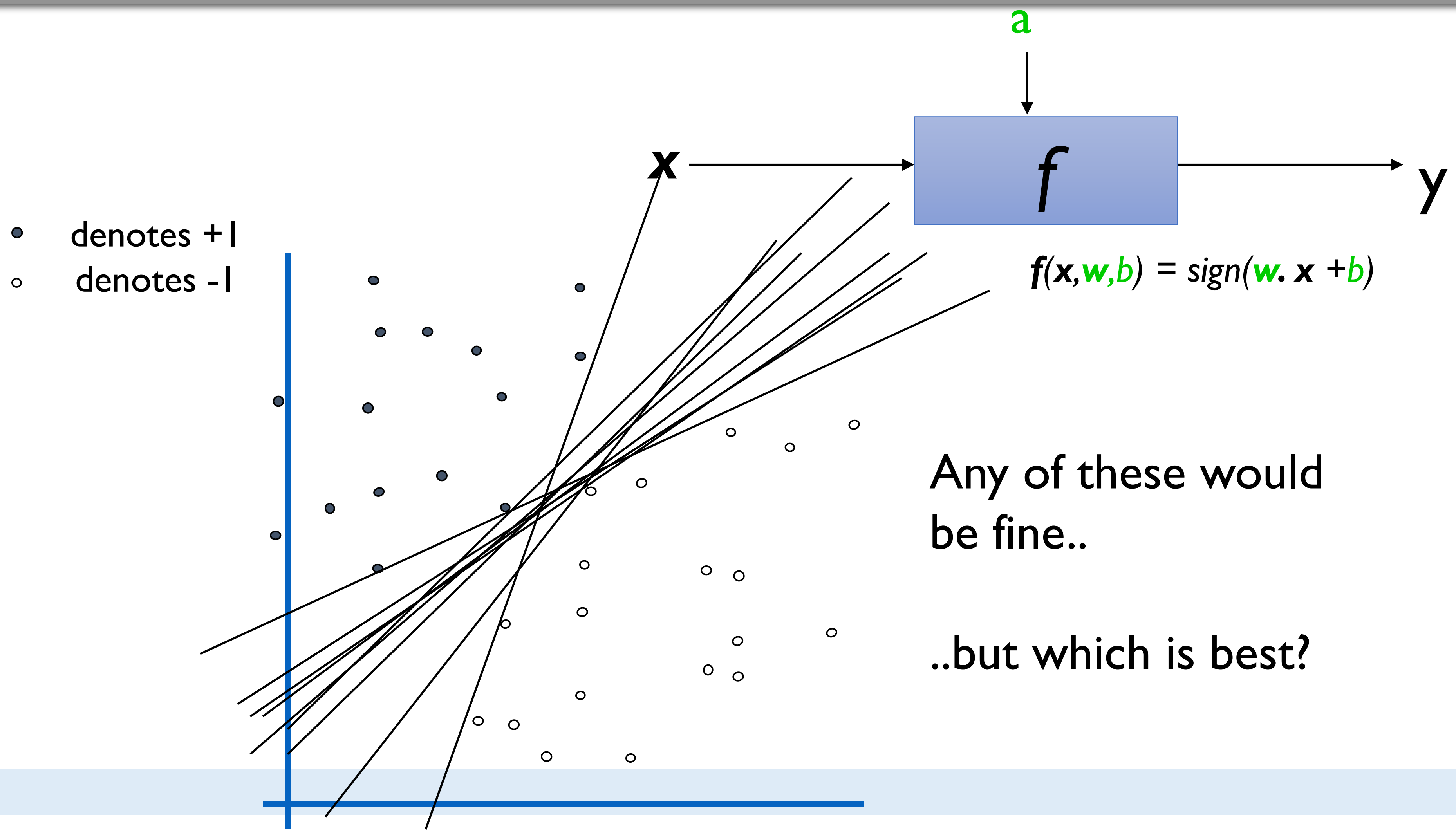


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$



How would you classify this data?

Linear Classifiers



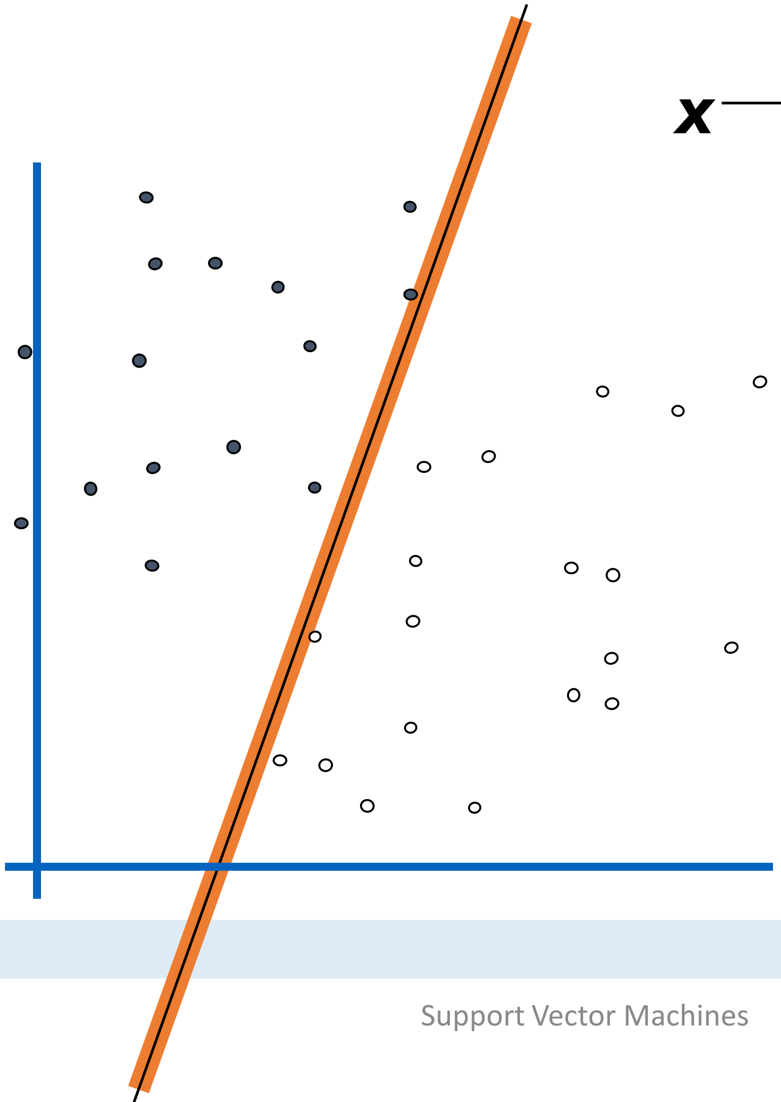
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$

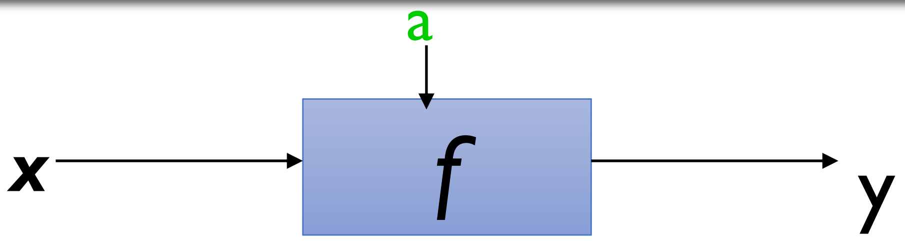
Any of these would be fine..

..but which is best?

Linear Classifiers

● denotes +1
○ denotes -1

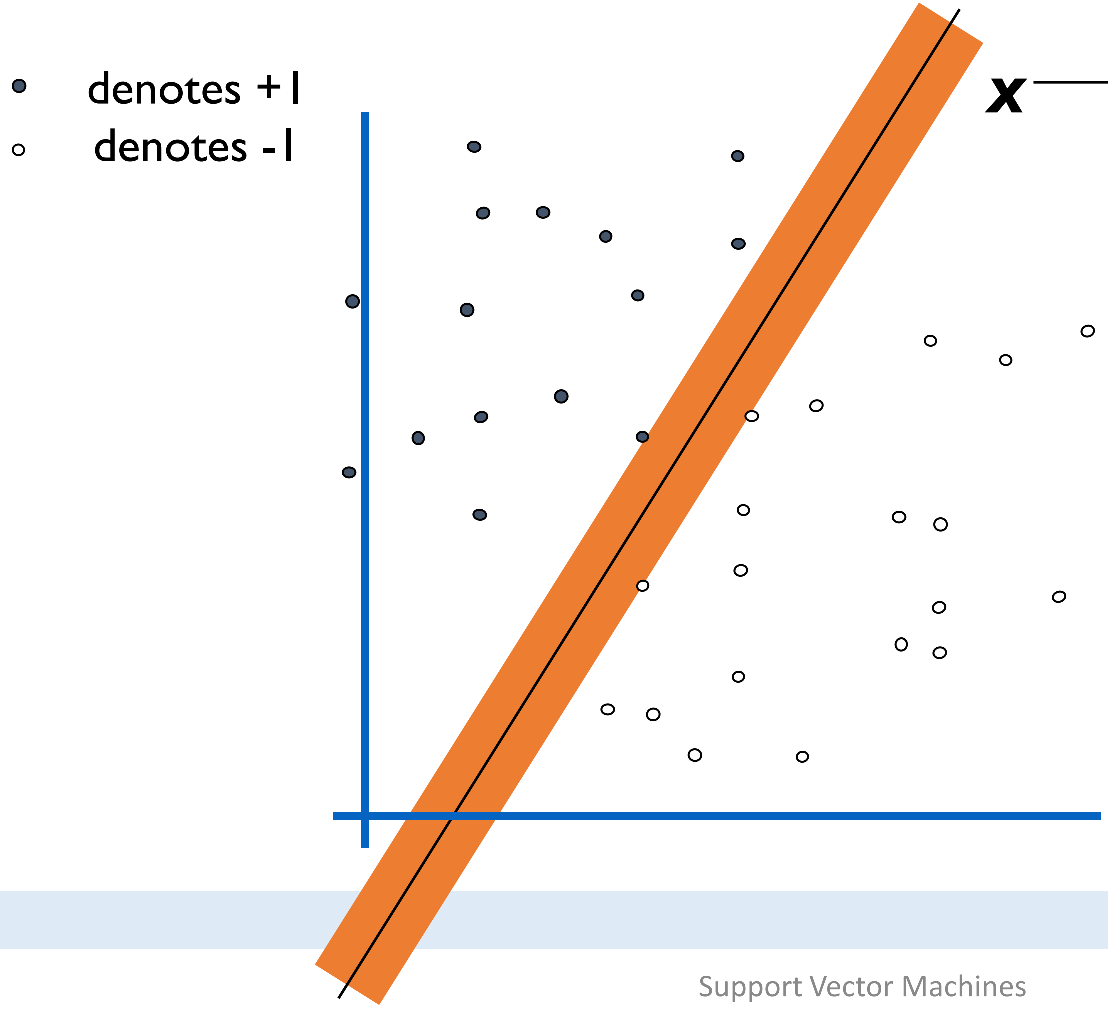


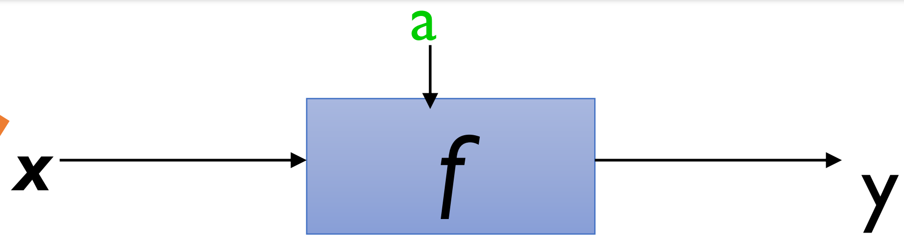


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$

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

Linear Classifiers



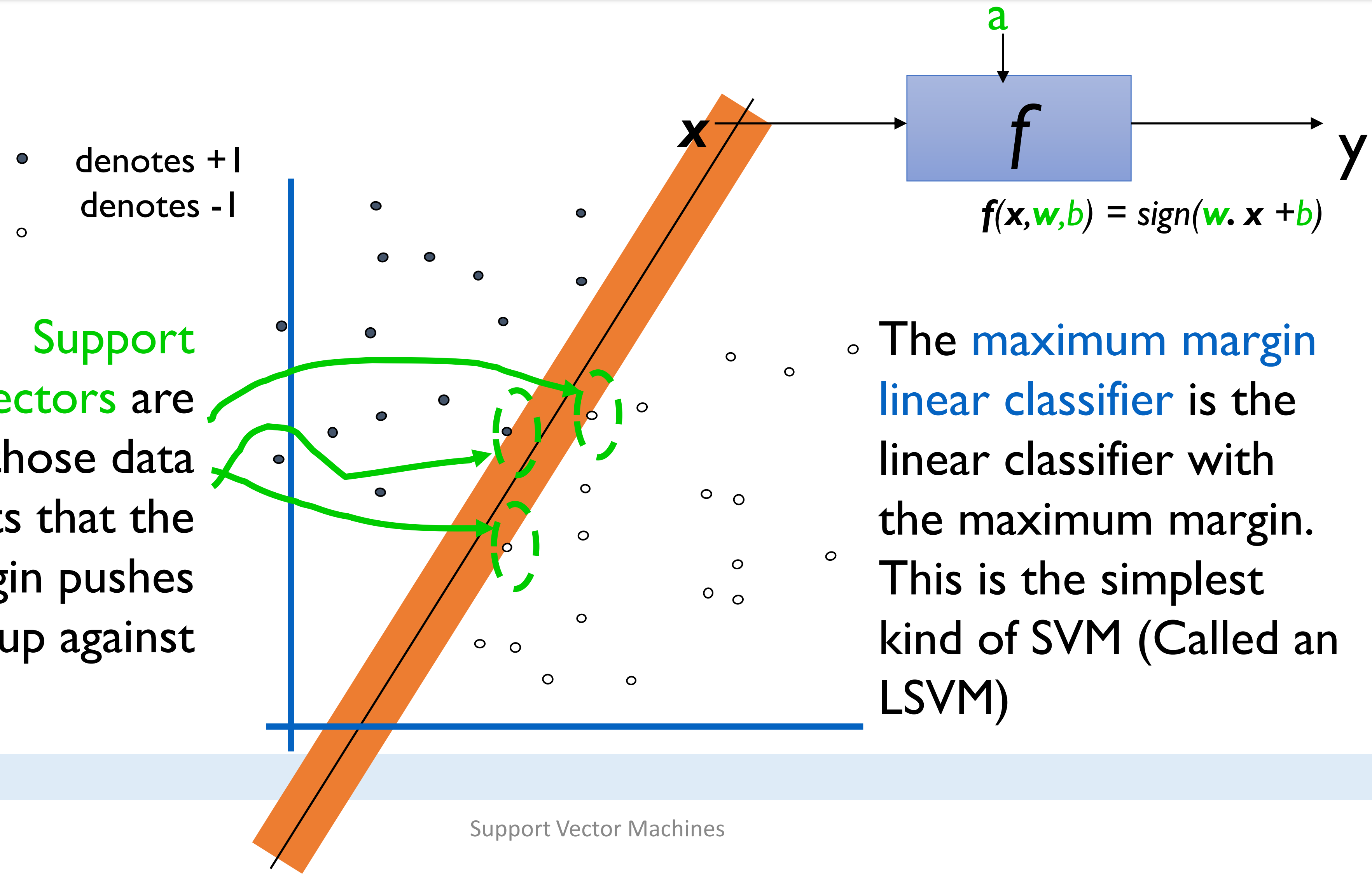


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$

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

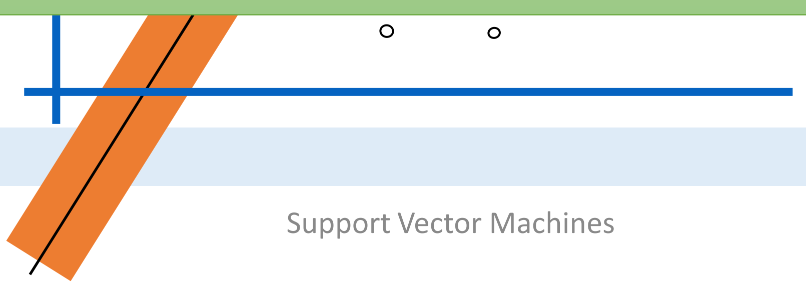


Support Vectors are those data points that the margin pushes up against

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

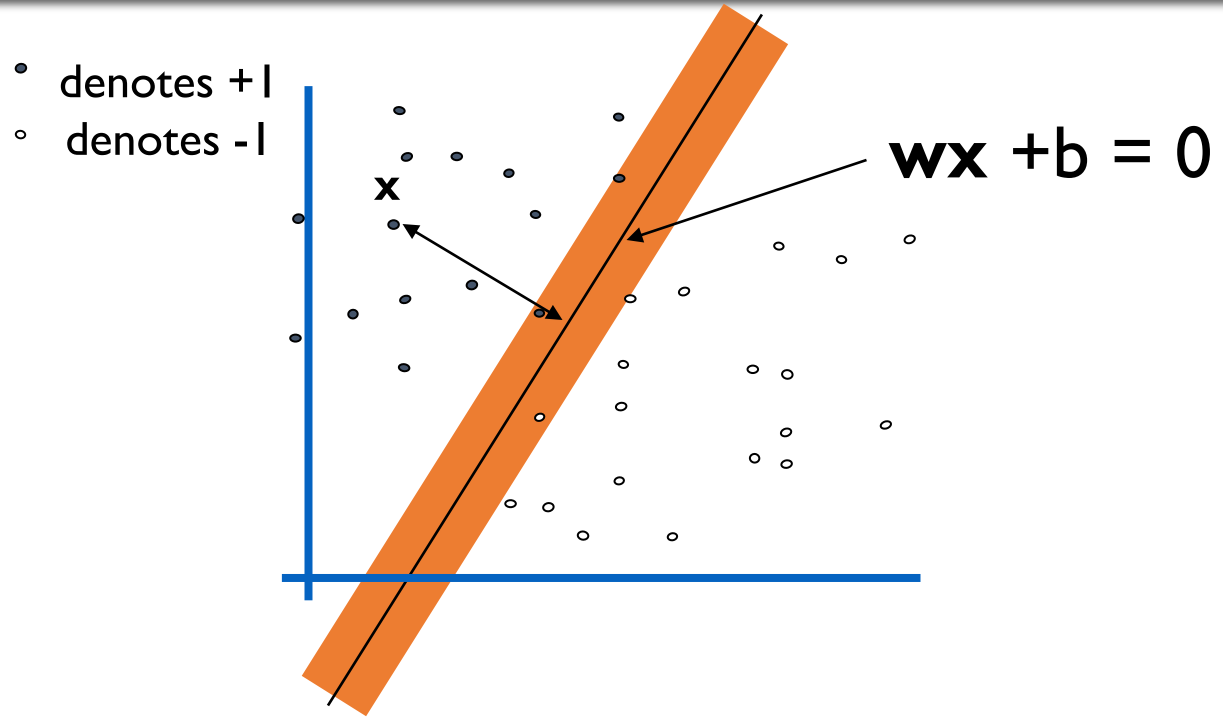
Why Maximum Margin?

- 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 well.



LSVM)

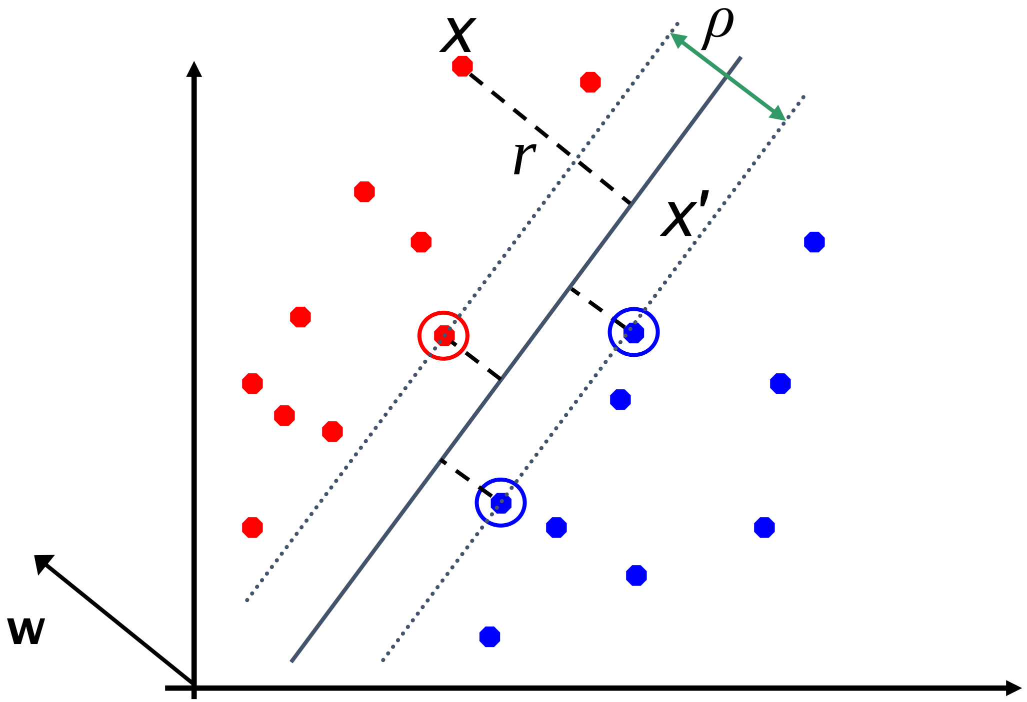
Estimating the Margin



- 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 $r = y\dfrac{\mathbf{w}^T\mathbf{x}+b}{\|\mathbf{w}\|}$

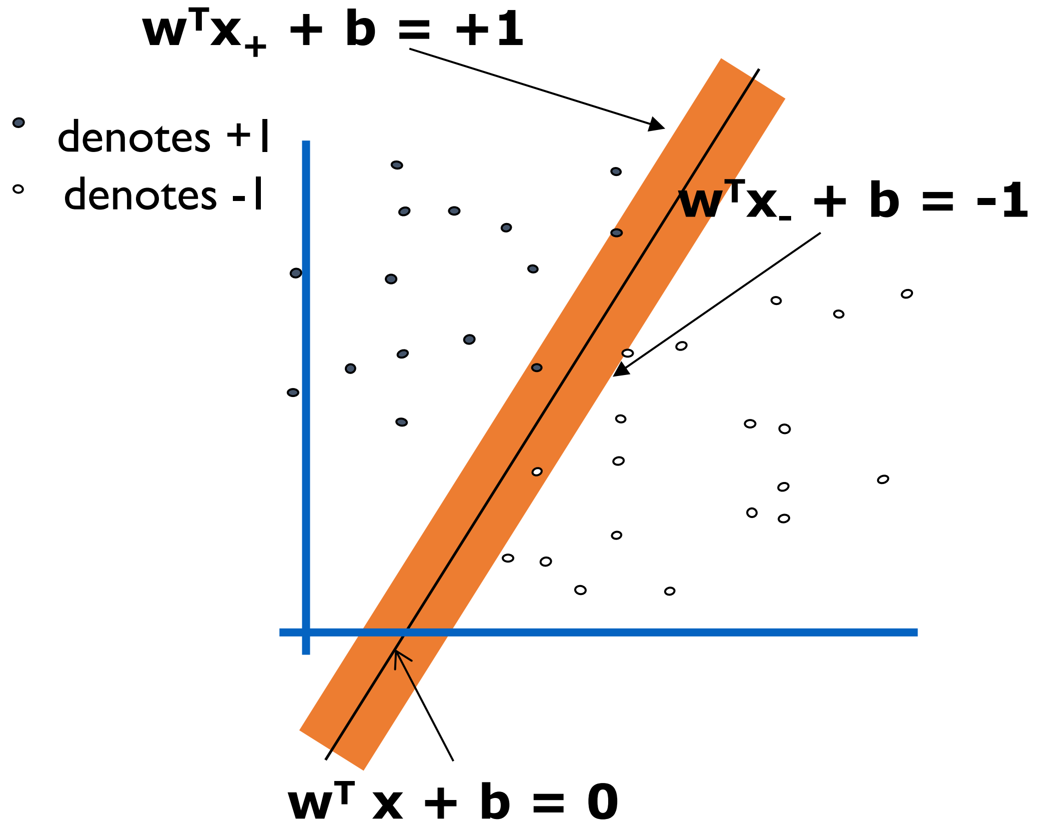


Derivation of finding *r*:

- 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 $||\mathbf{w}|| = \text{sqrt}(\mathbf{w}^T\mathbf{w})$.
- 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

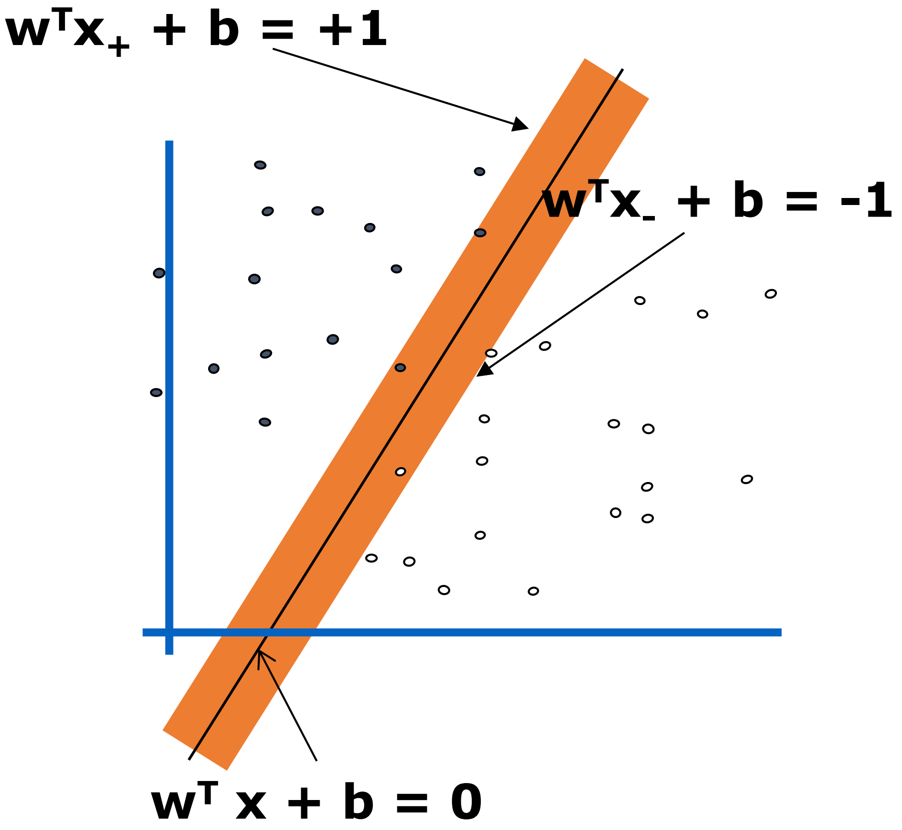
- Since $\mathbf{w}^T\mathbf{x} + b = 0$ and $c(\mathbf{w}^T\mathbf{x} + b) = 0$ define the same plane, we have the freedom to choose the normalization of $\mathbf{w}$ (i.e. c)
- Let us choose normalization such that $\mathbf{w}^T\mathbf{x}_+ + b = +1$ and $\mathbf{w}^T\mathbf{x}_- + b = -1$ for the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



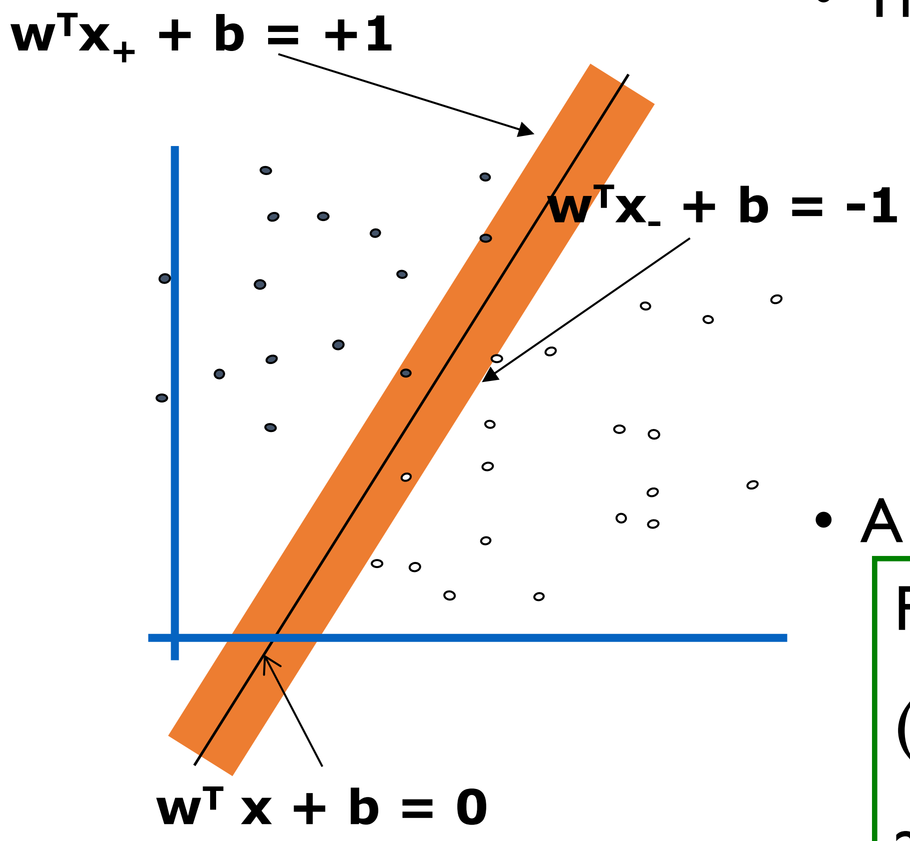
Estimating the Margin

- 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 **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

$r = \dfrac{2}{\|\mathbf{w}\|}$ 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 $y_i = -1$

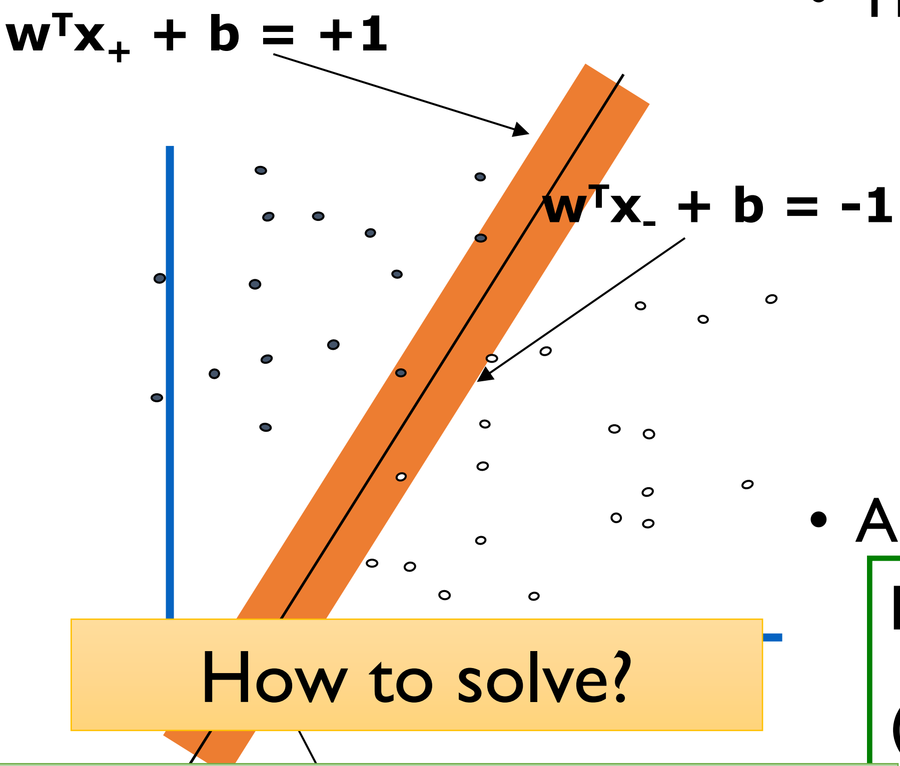
- A better formulation (min $\|\mathbf{w}\|$ = max $1/\|\mathbf{w}\|$):

Find **w** and *b* such that

$(\frac{1}{2}\,\mathbf{w}^T\mathbf{w})$ is minimized

and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i\,(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

Maximizing the Margin



How to solve?

Quadratic Programming

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

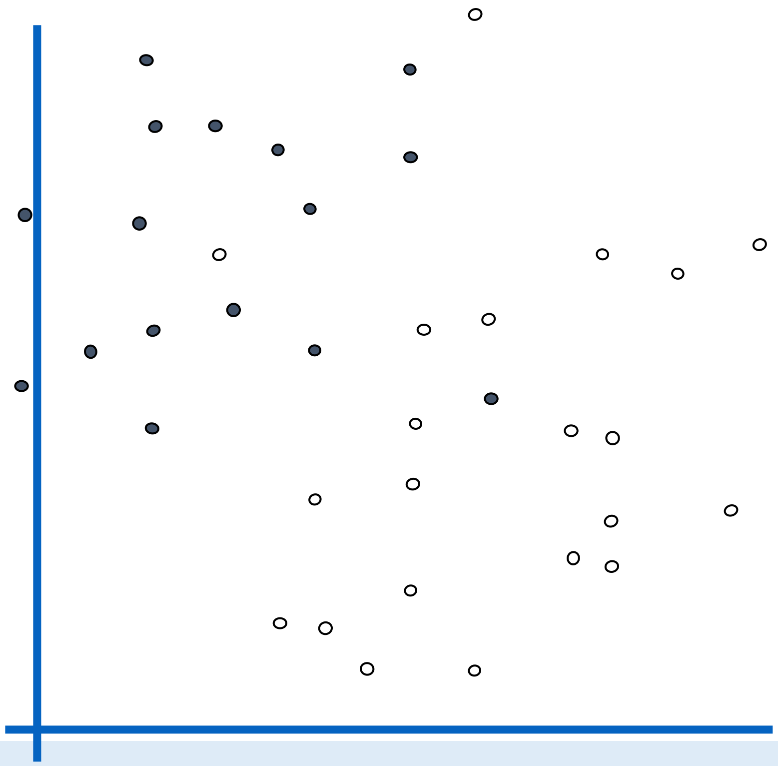
> Find $\mathbf{w}$ and b such that
>
> $$r = \frac{2}{\|\mathbf{w}\|}$$ is maximized; and for all $\{(\mathbf{x_i}, y_i)\}$
>
> $\mathbf{w}^T\mathbf{x_i} + b \geq 1$ if $y_i = +1$; $\mathbf{w}^T\mathbf{x_i} + b \leq -1$ if $y_i = -1$

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

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

Non-separable Data

- denotes +1
- denotes -1



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

SVM for Noisy Data

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

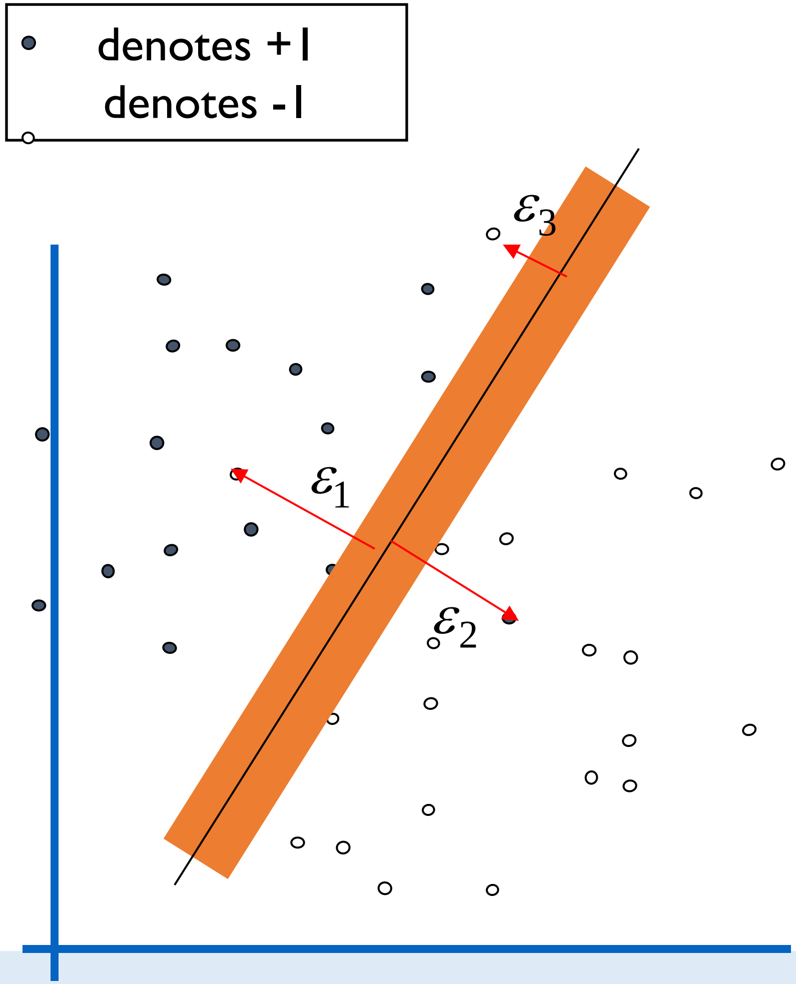
$$y_1\left(\vec{w}\cdot\vec{x}_1 + b\right) \geq 1 - \varepsilon_1, \varepsilon_1 \geq 0$$

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

...

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

Balance the trade off between margin and classification errors

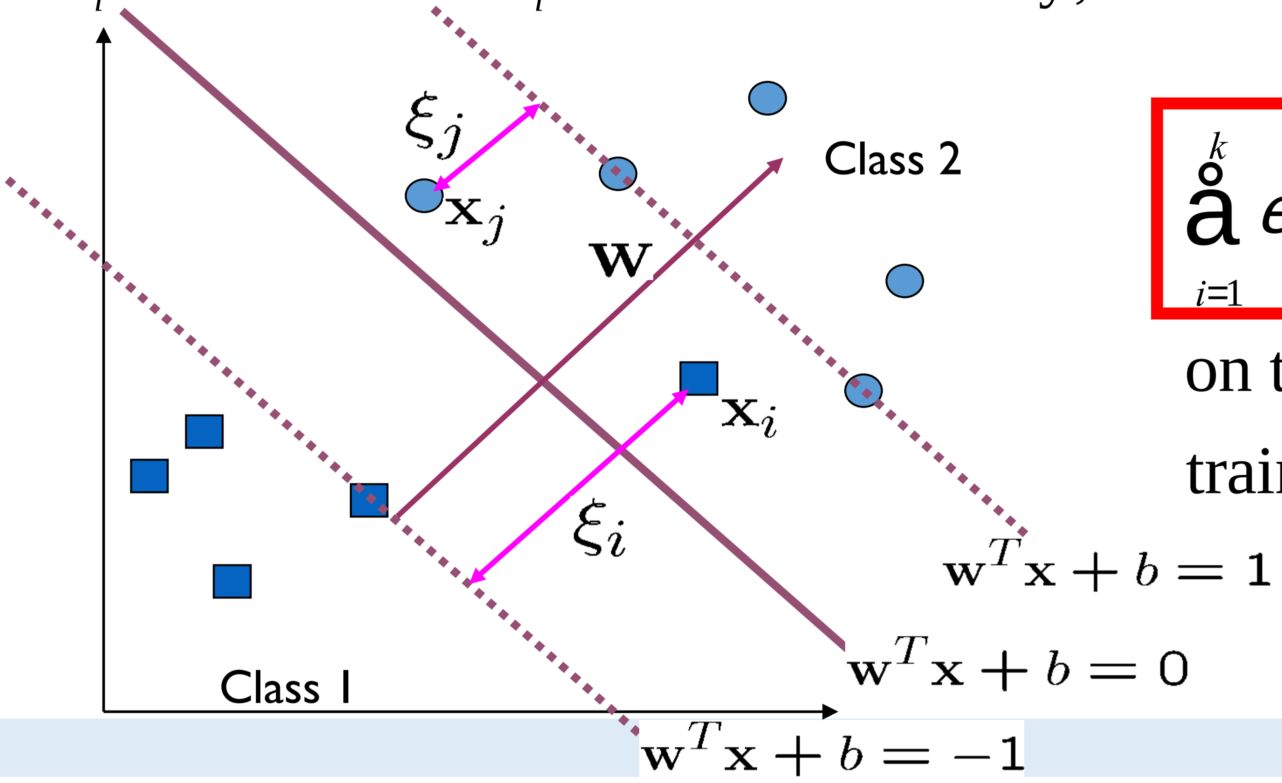


SVM for Noisy Data

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

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

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



$\boxed{\sum_{i=1}^{k} \varepsilon_i}$ 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: $0 \le \alpha_k \le c \quad \forall k$ $\quad \sum_{k=1}^{R} \alpha_k y_k = 0$

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

Once solved, we obtain w and b using:

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

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

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

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

Then classify with:

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

SVM Summary

SVM standard (primal) form:

Minimize: $\frac{1}{2}\|\vec{w}\|^2$
(w,b)

Such that: $y_i(\vec{w}\cdot\vec{x}_i - b) \geq 1$
(for all i)

Maximize γ = 2/||w||

SVM standard (dual) form:

Maximize: $\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j)$
(α_i)

Such that: $\sum_{i=1}^{n}\alpha_i y_i = 0 \qquad \alpha_i \geq 0$
(for all i)

Both yield the same solution

Only a function of "support vectors"

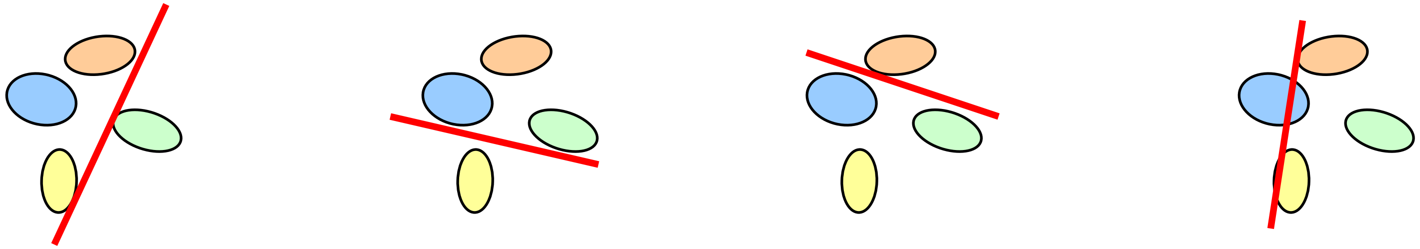
Slide credit: Nakul Verma, Columbia University

Multi-class Classification with SVMs

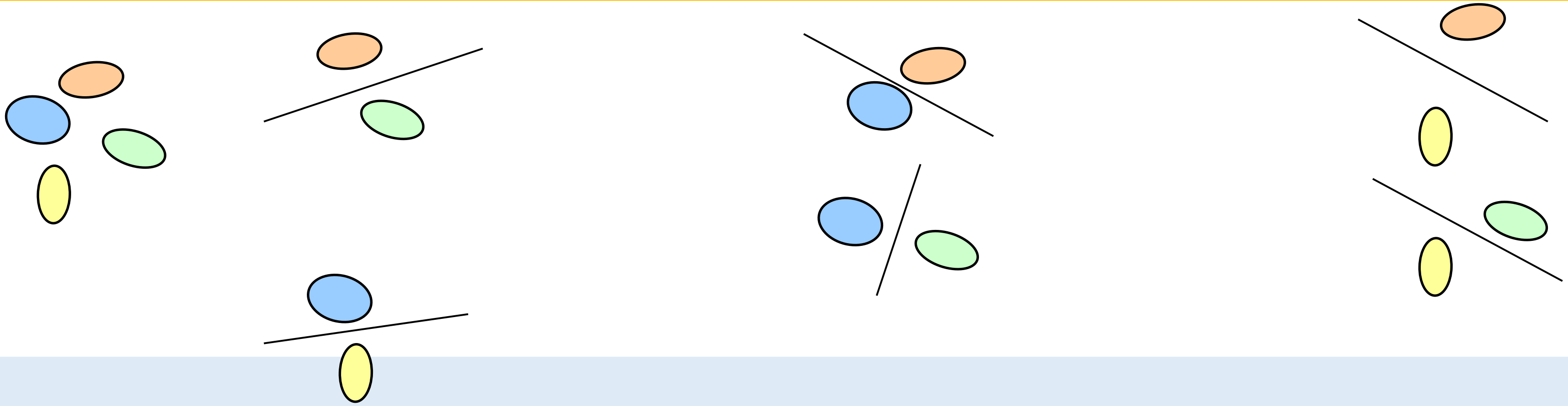
- SVMs can only handle two-class outputs.
- What can be done?
- Answer: with output arity N, learn N SVM's
  - SVM 1 learns "Output==1" vs "Output != 1"
  - SVM 2 learns "Output==2" vs "Output != 2"
  - :
  - SVM N learns "Output==N" vs "Output != N"
- Then to predict the output for a new input, just 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

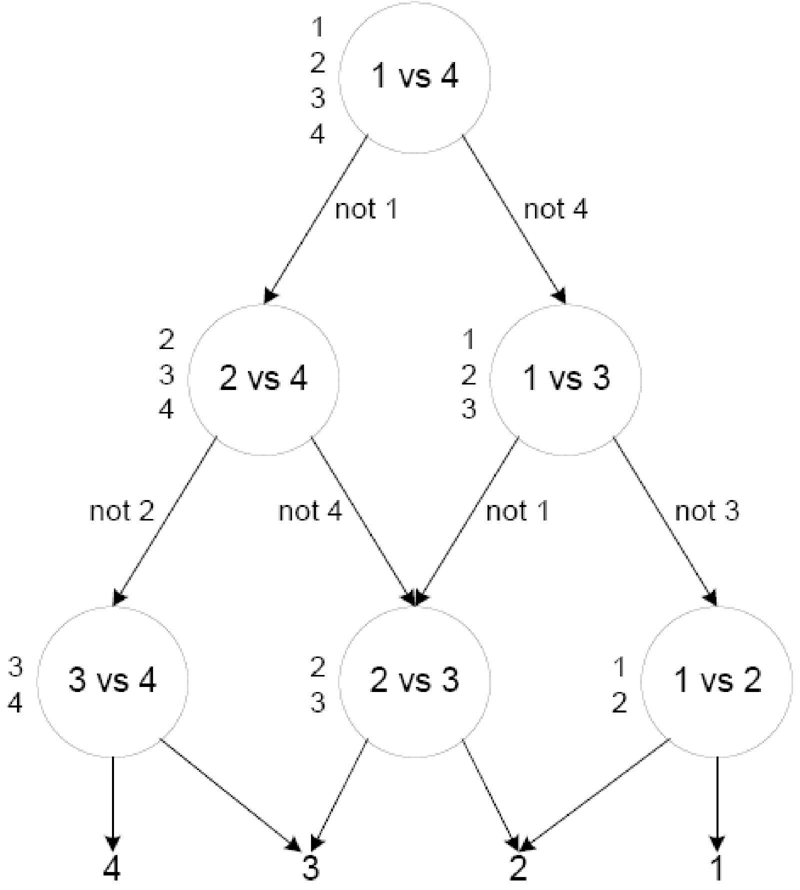
One- versus-all



One- versus-one



Tree-Structured SVM



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

Readings

- PRML, Bishop, Chapter 7 (7.1-7.3)
- "Introduction to Machine Learning" by Ethem Alpaydin, 2nd edition, Chapters 3 (3.1-3.4), Chapter 13 (13.1-13.9)
- Do read these!
  - https://www.svm-tutorial.com/2017/02/svms-overview-support-vector-machines/
  - https://www.svm-tutorial.com/2016/09/duality-lagrange-multipliers/
  - https://www.svm-tutorial.com/2017/10/support-vector-machines-succinctly-released/