

Foundations of Machine Learning

Support Vector Machines

Sep 2021

Vineeth N Balasubramanian



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

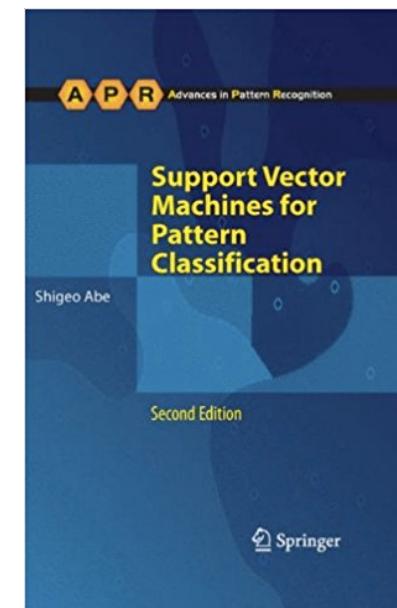
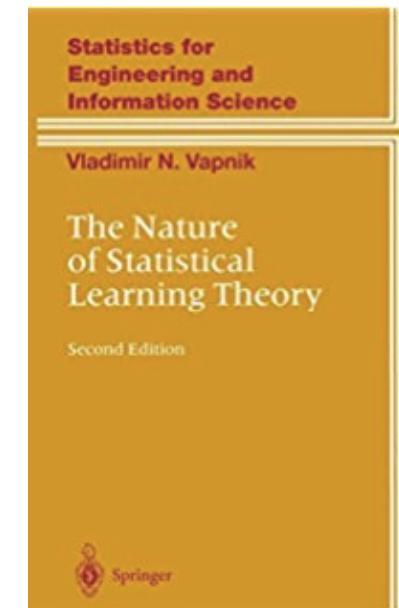
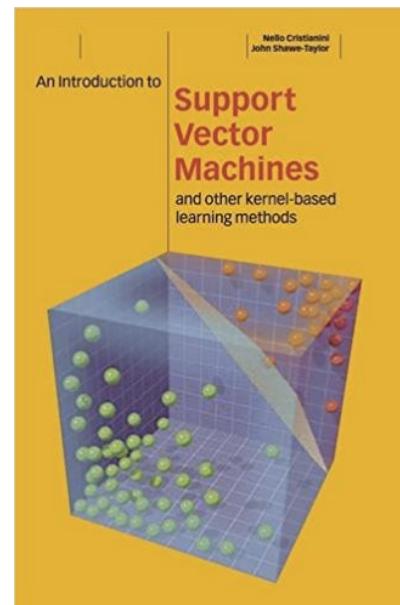
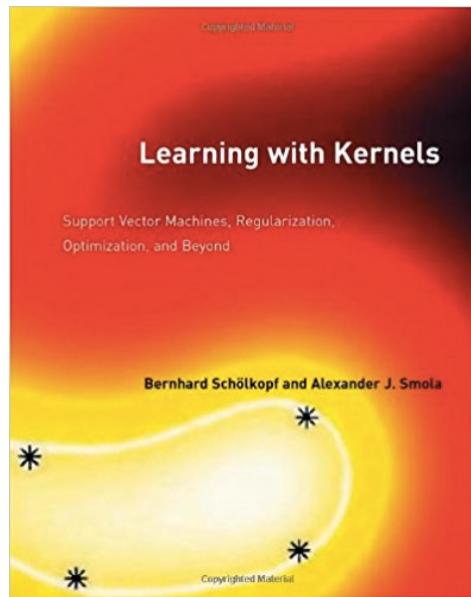
- A discriminative classifier
 - Non-parametric, Inductive
- SVM is inspired from statistical learning theory
- SVM was developed in 1992 by Vapnik, Guyon and Boser
- SVM became popular because of its success in handwritten digit recognition
- Has been one of the go-to methods in machine learning since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

Papers that introduced SVM in its current form

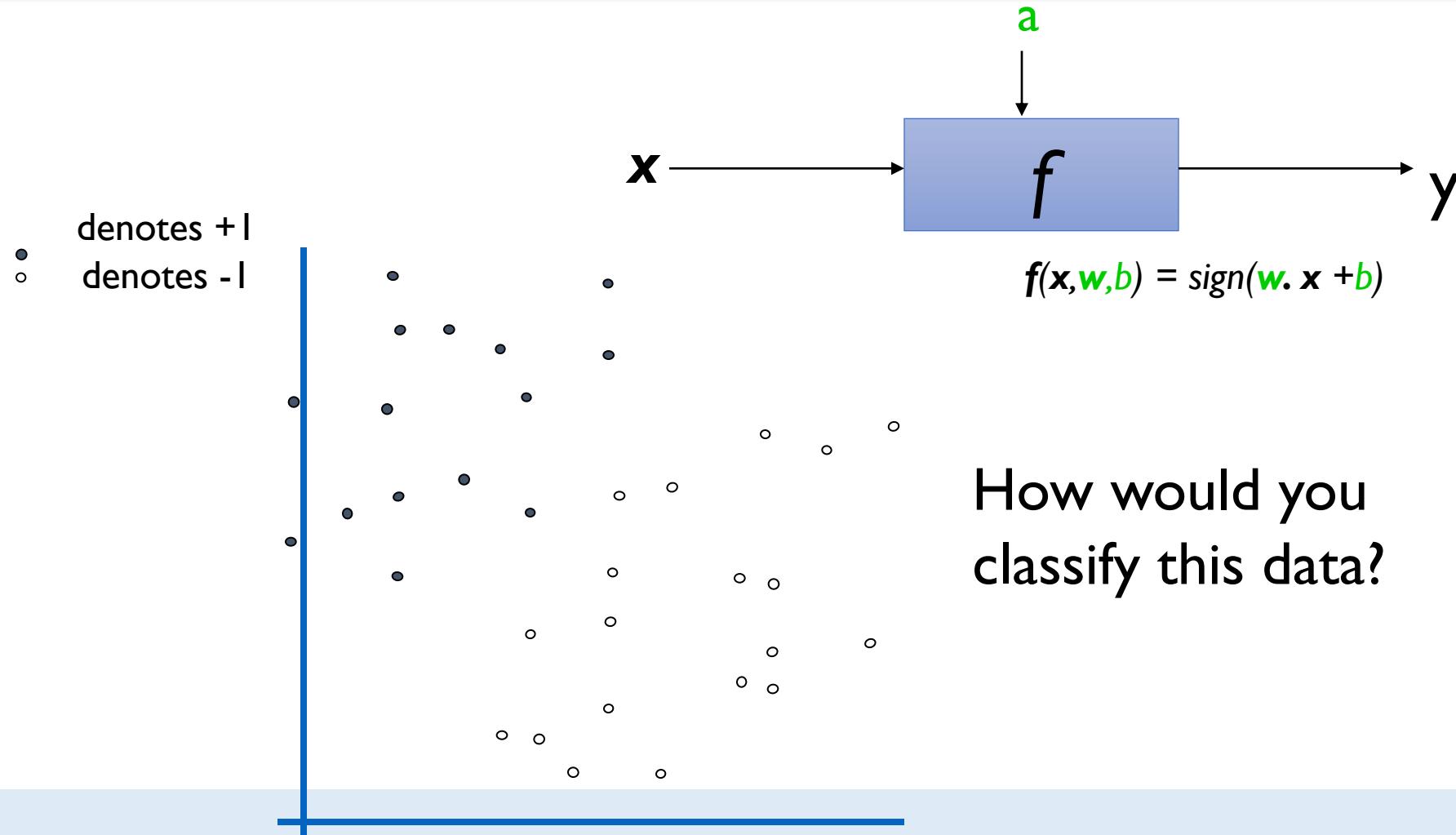
- Boser, B. E.; Guyon, I. M.; Vapnik, V. N. (1992). "A training algorithm for optimal margin classifiers". Proceedings of the fifth annual workshop on Computational learning theory – COLT '92.
- Cortes, C.; Vapnik, V. (1995). "Support-vector networks". Machine Learning. 20 (3): 273–297.

SVM: Overview and History

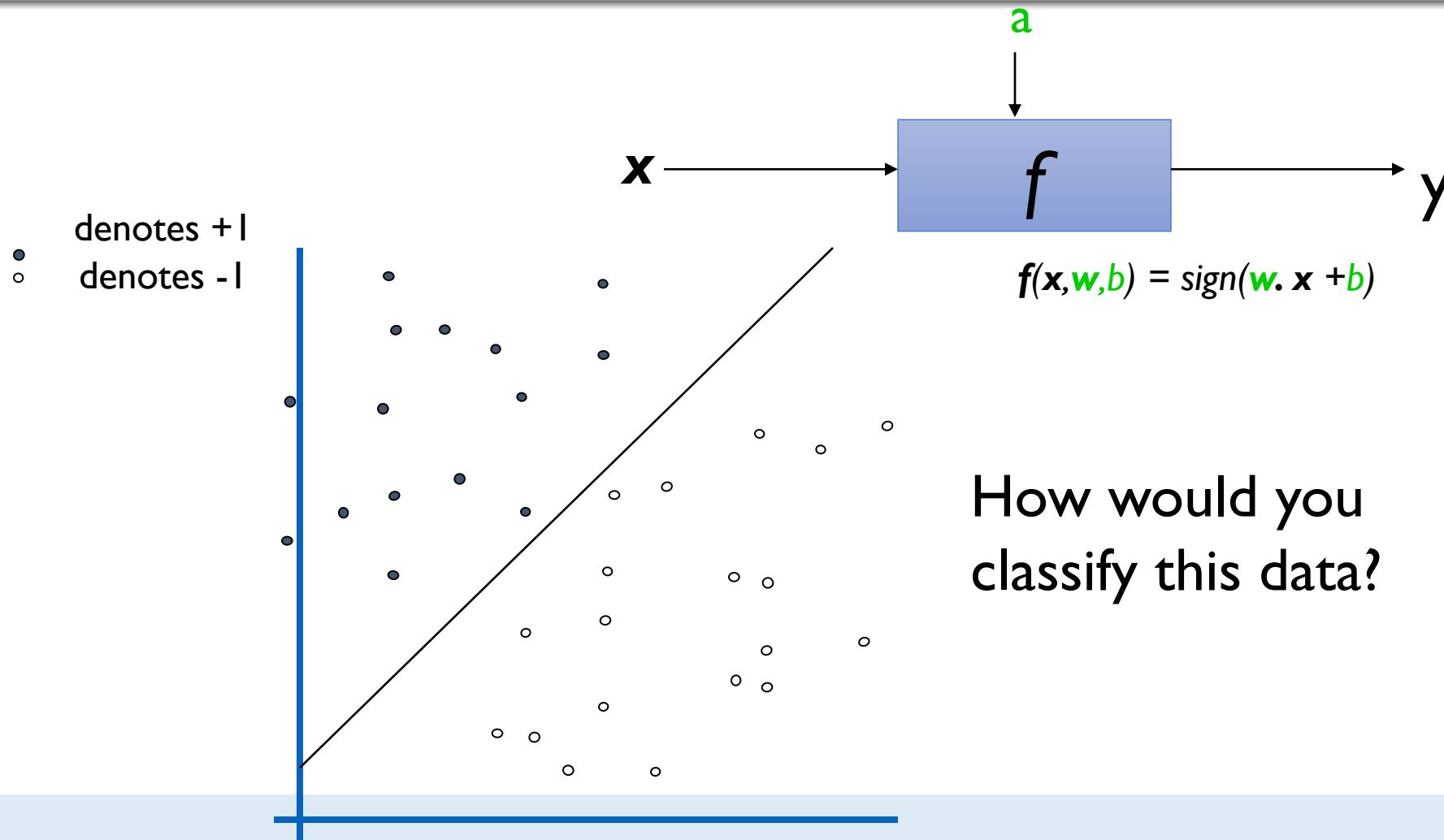
- Associated key words
 - Large-margin classifier, Max-margin classifier, Kernel methods, Reproducing kernel Hibert space, Statistical learning theory



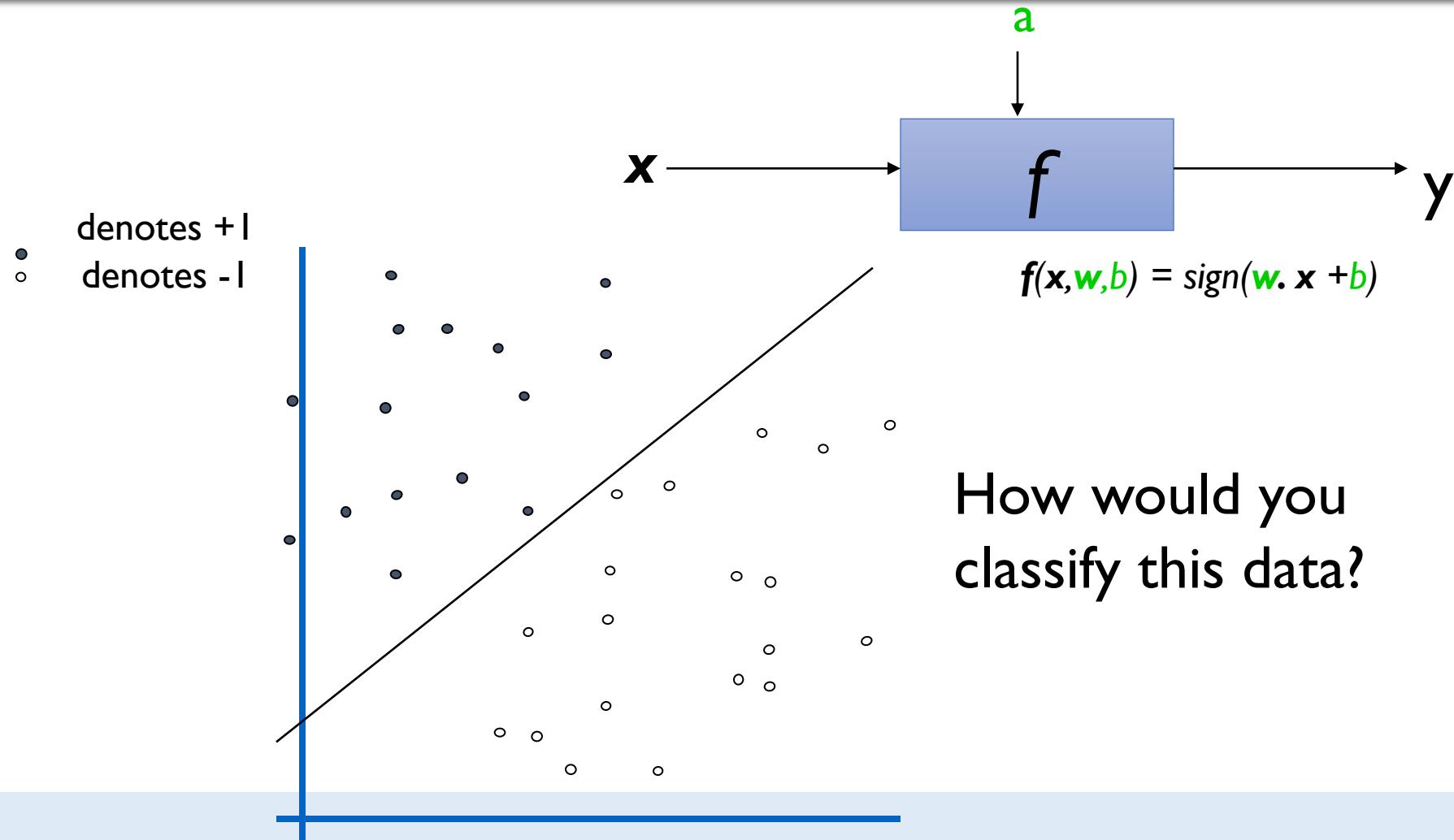
Linear Classifiers



Linear Classifiers

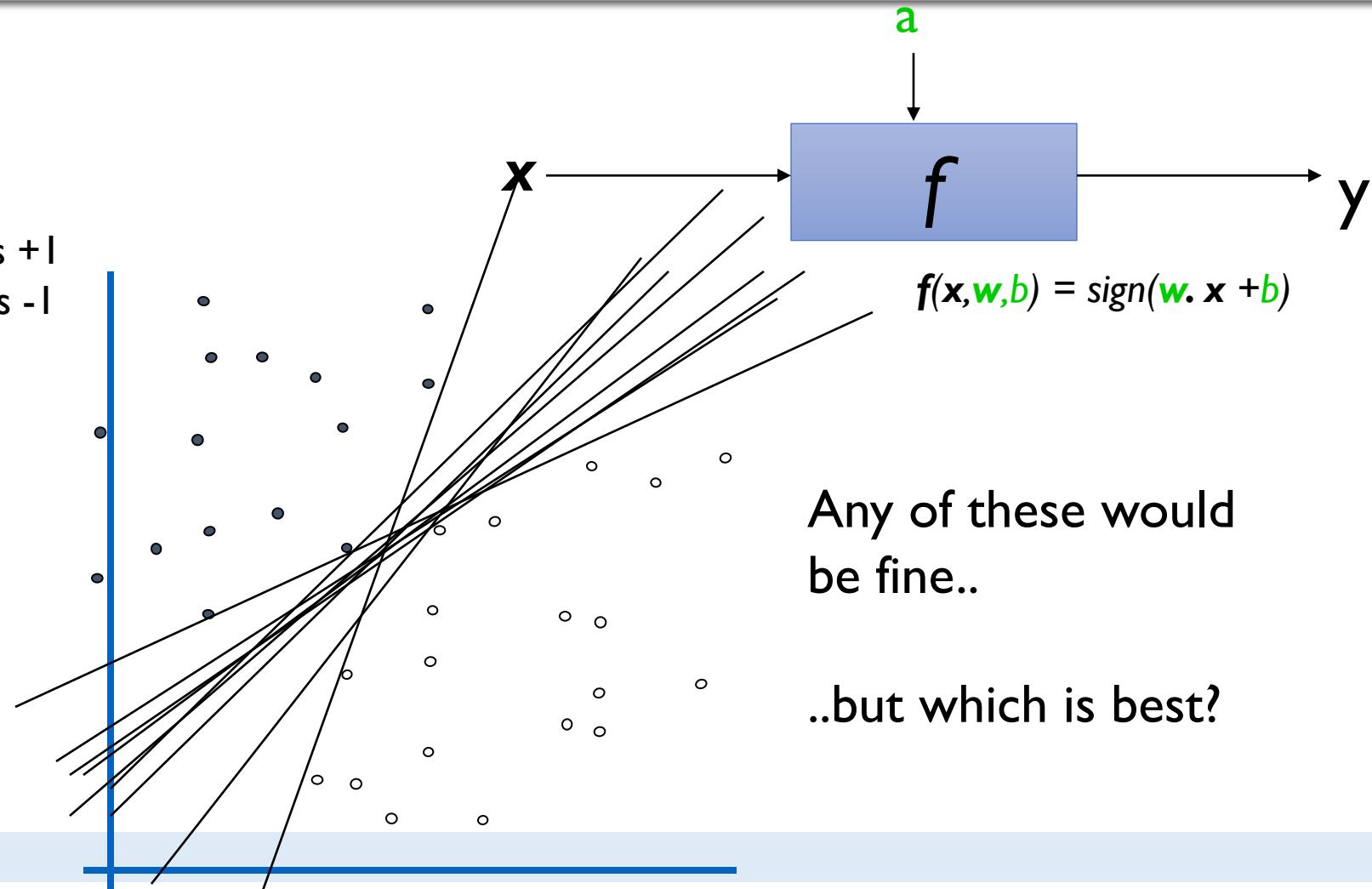


Linear Classifiers



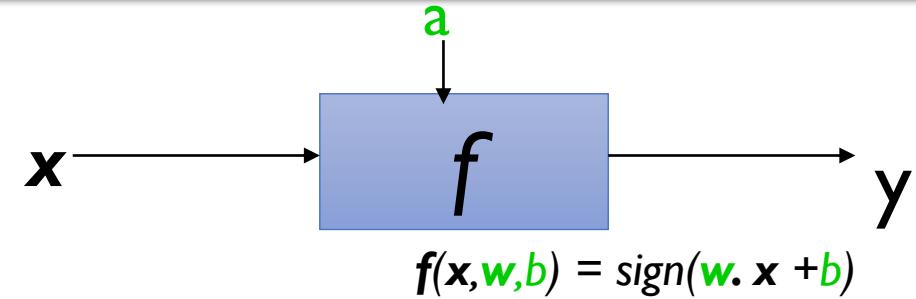
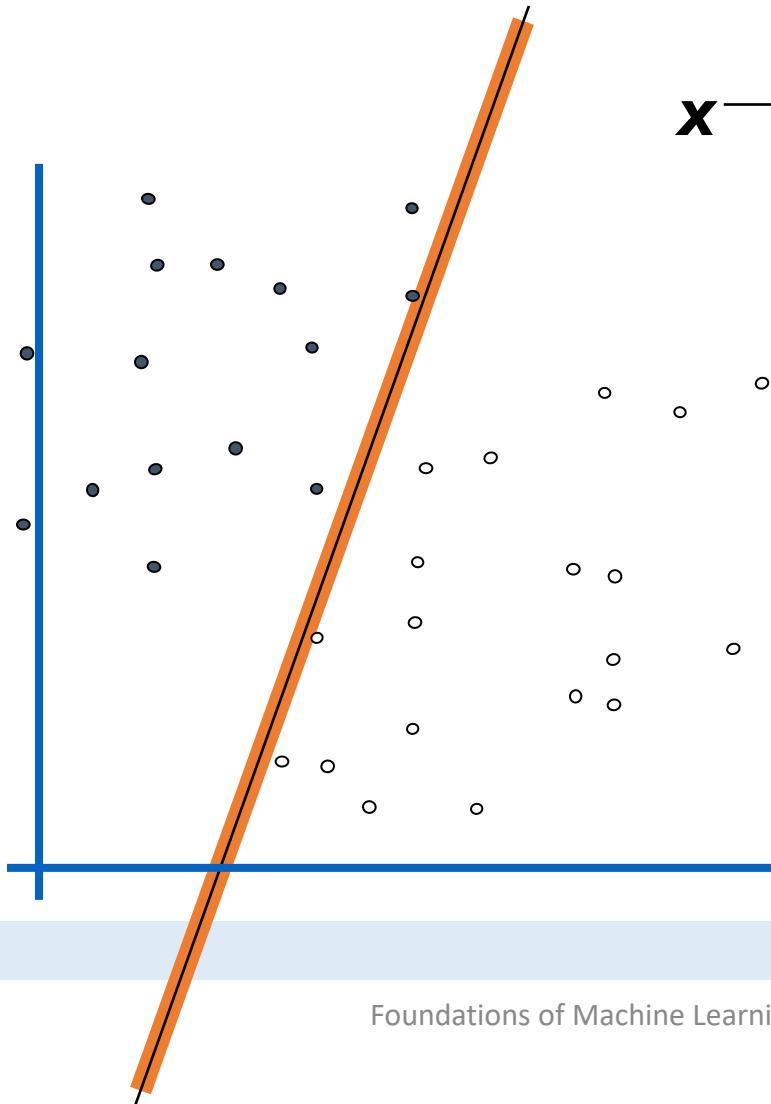
Linear Classifiers

- denotes +1
- denotes -1



Linear Classifiers

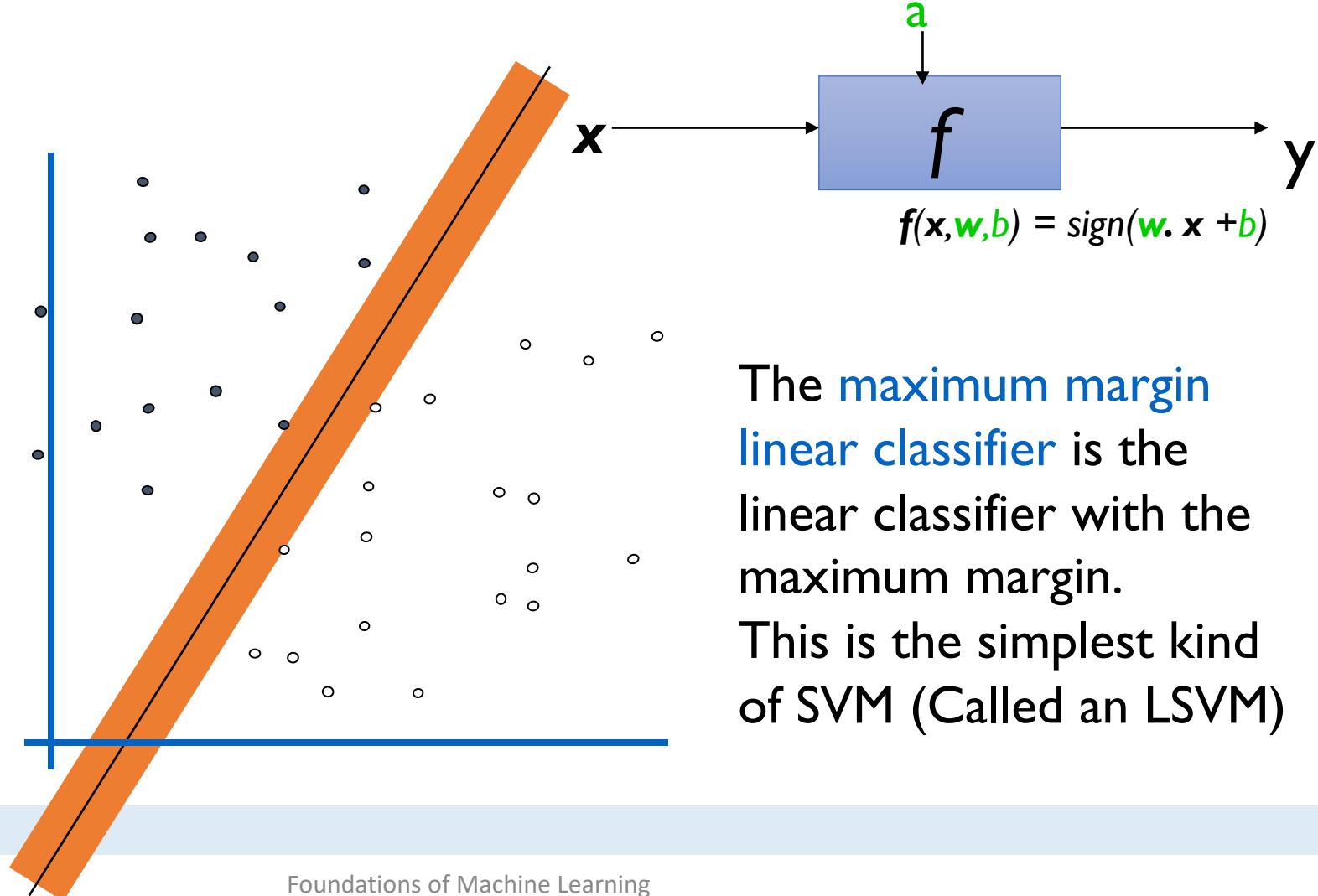
- denotes +1
- denotes -1



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

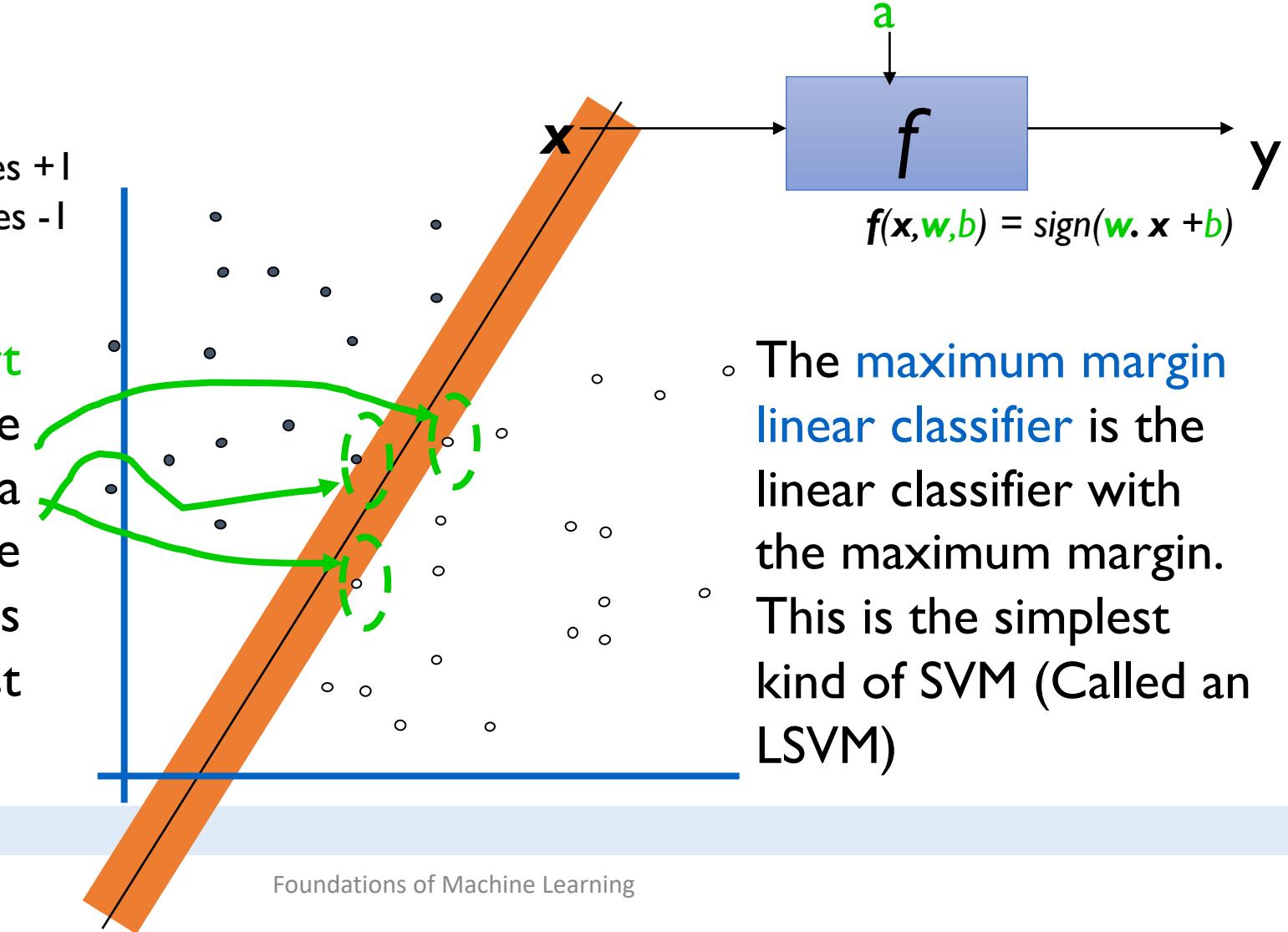
Linear Classifiers

- denotes +1
- denotes -1



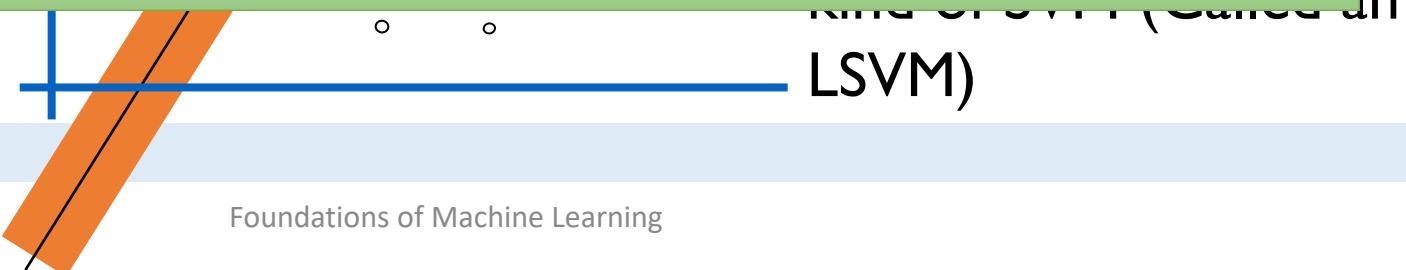
Maximum Margin Classifier

- denotes +1
 - denotes -1
- Support Vectors** are those data points that the margin pushes up against



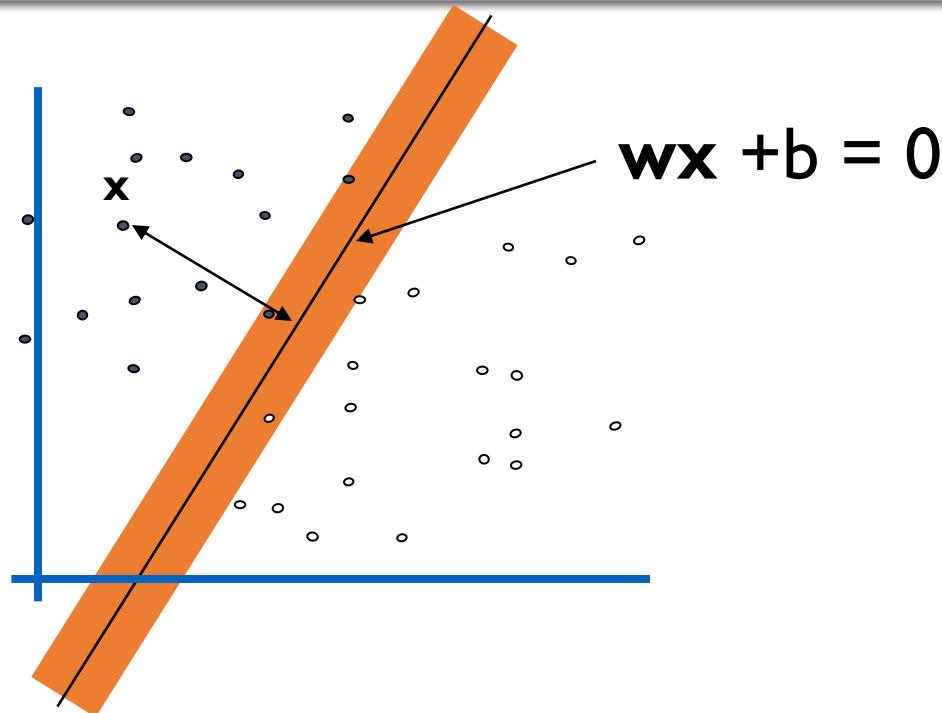
Why Maximum Margin?

- 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.



Estimating the Margin

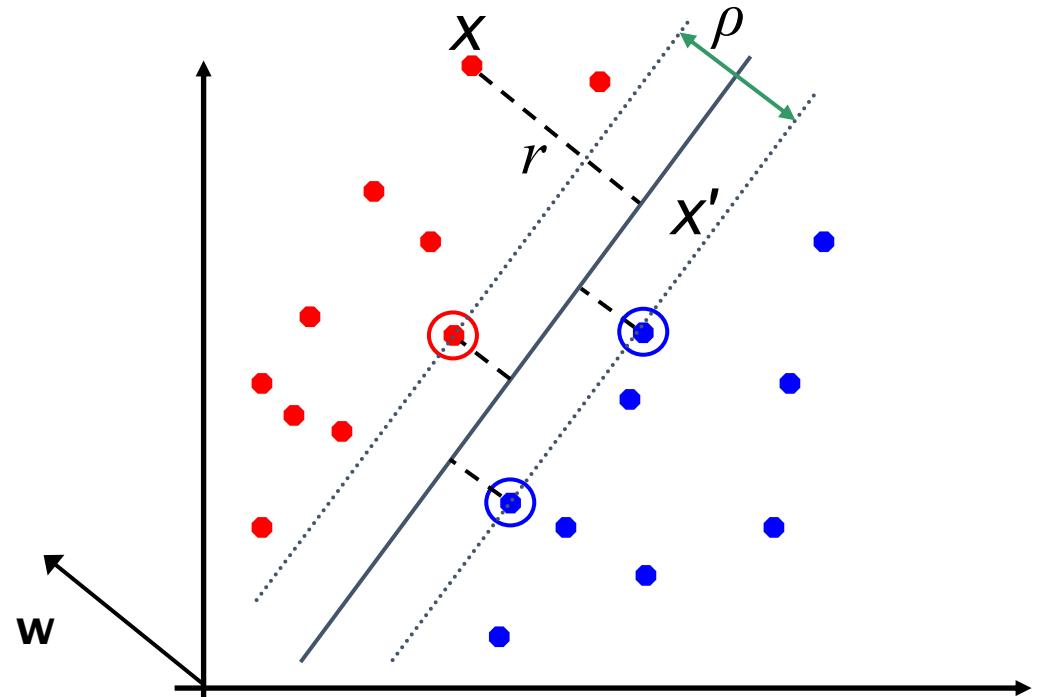
- denotes +1
- denotes -1



- What is the distance expression for a point \mathbf{x} to a line $\mathbf{w}\mathbf{x}+b=0$?

Estimating the Margin

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

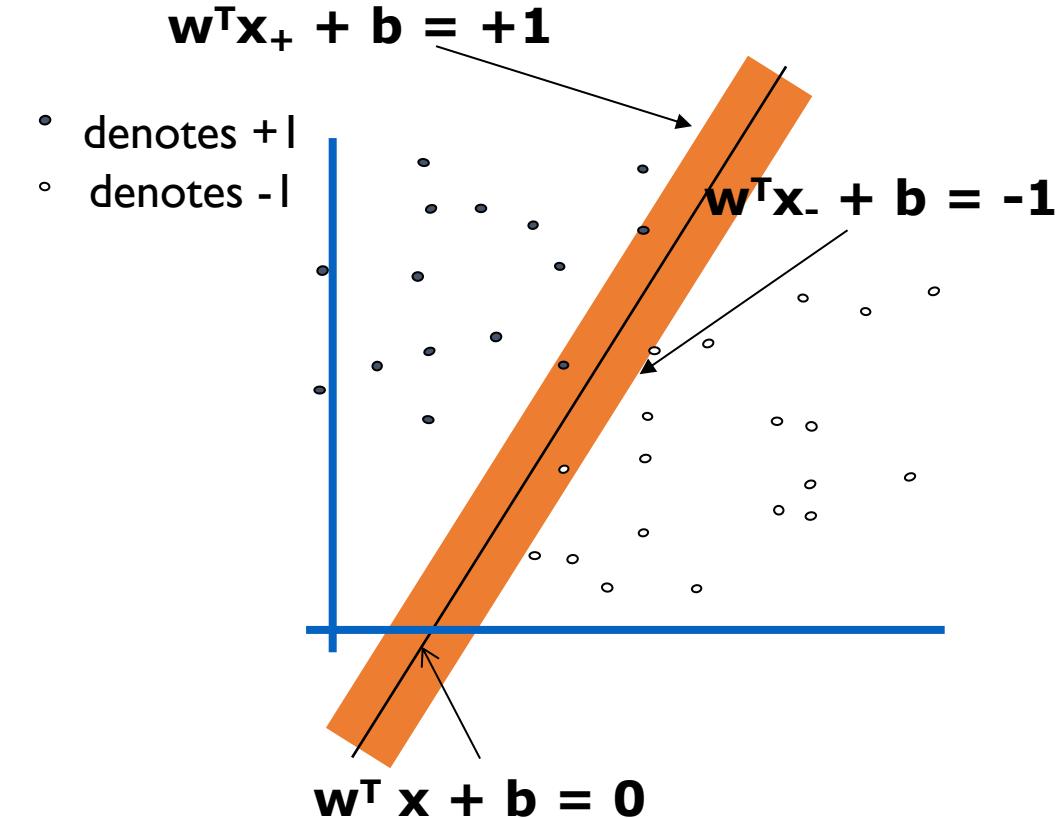


Derivation of finding r :

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

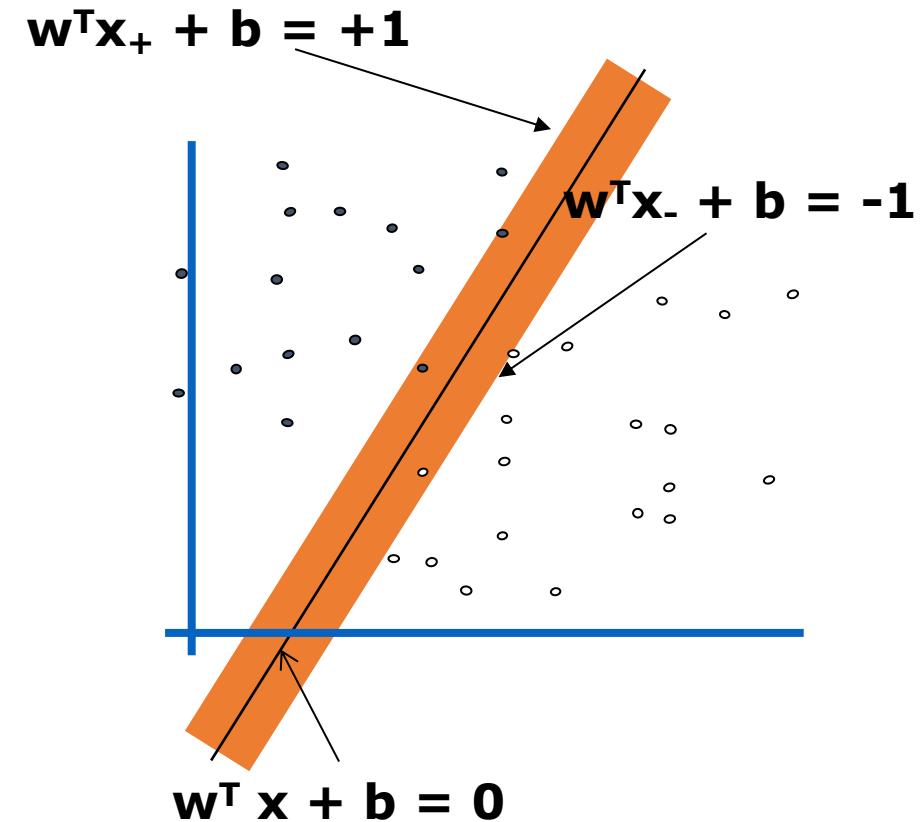
- Since $\mathbf{w}^T \mathbf{x} + b = 0$ and $c(\mathbf{w}^T \mathbf{x} + b) = 0$ define the same plane, we have the freedom to choose the normalization of \mathbf{w} (i.e. c)
- Let us choose normalization such that $\mathbf{w}^T \mathbf{x}_+ + b = +1$ and $\mathbf{w}^T \mathbf{x}_- + b = -1$ for the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



Estimating the Margin

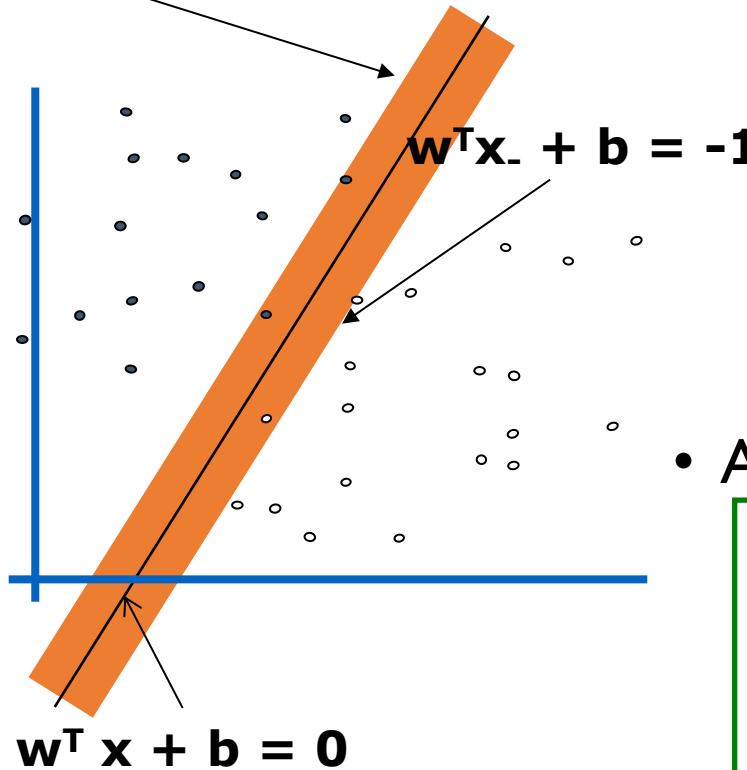
- Since $\mathbf{w}^T \mathbf{x} + b = 0$ and $c(\mathbf{w}^T \mathbf{x} + b) = 0$ define the same plane, we have the freedom to choose the normalization of \mathbf{w} (i.e. c)
- Let us choose normalization such that $\mathbf{w}^T \mathbf{x}_+ + b = +1$ and $\mathbf{w}^T \mathbf{x}_- + b = -1$ for the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

$$\mathbf{w}^T \mathbf{x}_+ + b = +1$$



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

Find \mathbf{w} and b such that

$$\rho = \frac{2}{\|\mathbf{w}\|} \text{ is maximized; and for all } \{(\mathbf{x}_i, y_i)\}$$

$$\mathbf{w}^T \mathbf{x}_i + b \geq 1 \text{ if } y_i = +1; \quad \mathbf{w}^T \mathbf{x}_i + b \leq -1 \quad \text{if } y_i = -1$$

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

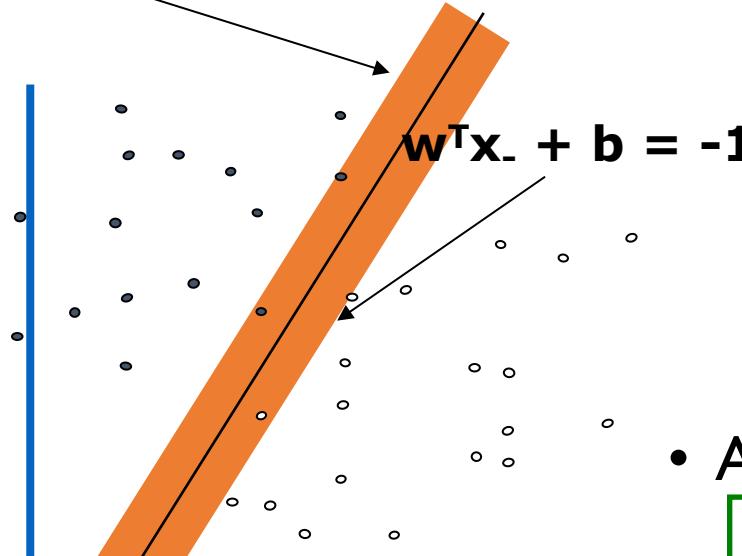
Find \mathbf{w} and b such that

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

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

Maximizing the Margin

$$\mathbf{w}^T \mathbf{x}_+ + b = +1$$



How to solve?

Quadratic Programming

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

Find \mathbf{w} and b such that

$$\rho = \frac{2}{\|\mathbf{w}\|} \text{ is maximized; and for all } \{(\mathbf{x}_i, y_i)\}$$

$$\mathbf{w}^T \mathbf{x}_i + b \geq 1 \text{ if } y_i = +1; \quad \mathbf{w}^T \mathbf{x}_i + b \leq -1 \text{ if } y_i = -1$$

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

Find \mathbf{w} and b such that

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

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

Using Lagrange Multipliers

Consider the augmented function:

$$L(\vec{x}, \vec{\lambda}) := f(\vec{x}) + \sum_{i=1}^n \lambda_i g_i(\vec{x})$$

(Lagrange function)

(Lagrange variables,
or dual variables)

Observation:

For **any** feasible \vec{x} and **all** $\lambda_i \geq 0$, we have $L(\vec{x}, \vec{\lambda}) \leq f(\vec{x})$

$$\Rightarrow \max_{\lambda_i \geq 0} L(\vec{x}, \vec{\lambda}) \leq f(\vec{x})$$

So, the optimal value to the constrained optimization:

$$p^* := \min_{\vec{x}} \max_{\lambda_i \geq 0} L(\vec{x}, \vec{\lambda})$$

The problem becomes
unconstrained in \vec{x} !

Slide credit: Nakul Verma, Columbia University

Optimization problem:

Minimize: $f(\vec{x})$

Such that:
(for all i) $g_i(\vec{x}) \leq 0$

Duality

Optimal value: $p^* = \min_{\vec{x}} \max_{\lambda_i \geq 0} L(\vec{x}, \vec{\lambda})$
(also called the primal)

Now, consider the function: $\min_{\vec{x}} L(\vec{x}, \vec{\lambda})$

Observation:

Since, for **any** feasible x and **all** $\lambda_i \geq 0$:

$$p^* \geq \min_{\vec{x}} L(\vec{x}, \vec{\lambda})$$

Thus:

$$d^* := \max_{\lambda_i \geq 0} \min_{\vec{x}'} L(\vec{x}', \vec{\lambda}) \leq p^*$$

(also called the dual)

Optimization problem:

Minimize: $f(\vec{x})$
Such that: $g_i(\vec{x}) \leq 0$
(for all i)

Lagrange function:

$$L(\vec{x}, \vec{\lambda}) := f(\vec{x}) + \sum_{i=1}^n \lambda_i g_i(\vec{x})$$

Duality

Theorem (weak Lagrangian duality):

$$d^* \leq p^*$$

(also called the minimax inequality)

$$p^* - d^* \quad (\text{called the duality gap})$$

Under what conditions can we achieve equality?

Optimization problem:

Minimize: $f(\vec{x})$

Such that: $g_i(\vec{x}) \leq 0$
(for all i)

Lagrange function:

$$L(\vec{x}, \vec{\lambda}) := f(\vec{x}) + \sum_{i=1}^n \lambda_i g_i(\vec{x})$$

Primal:

$$p^* = \min_{\vec{x}} \max_{\lambda_i \geq 0} L(\vec{x}, \vec{\lambda})$$

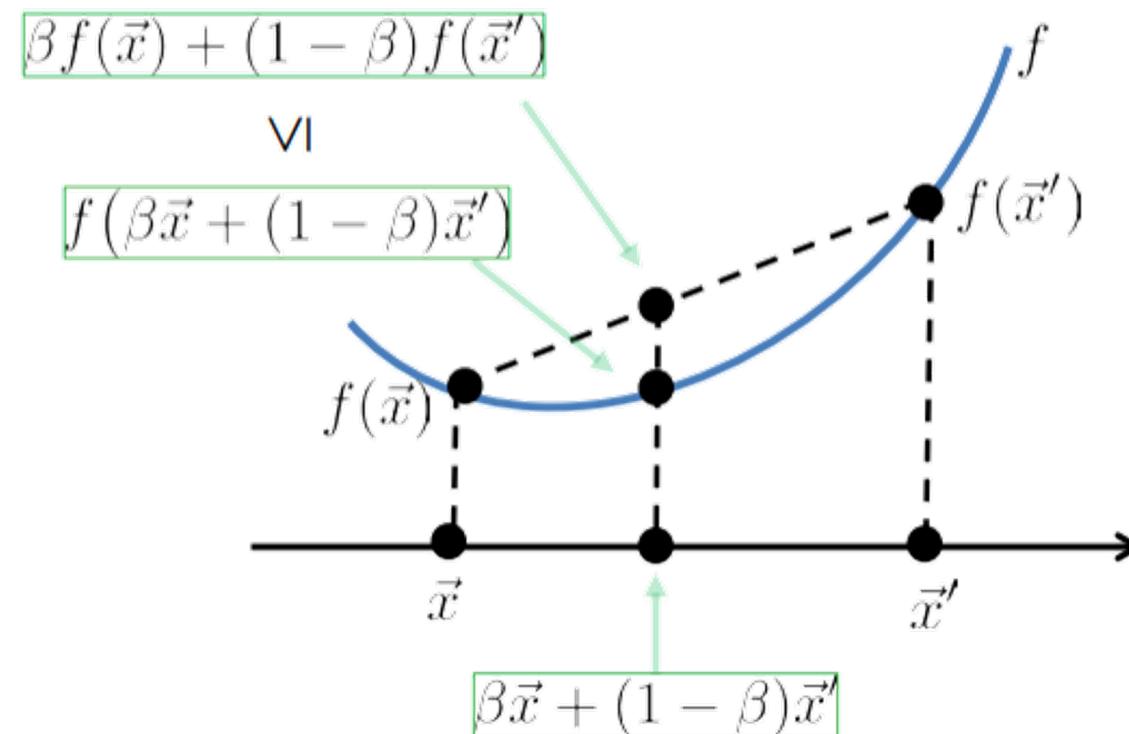
Dual:

$$d^* := \max_{\lambda_i \geq 0} \min_{\vec{x}} L(\vec{x}, \vec{\lambda})$$

Convexity

A function $f: \mathbf{R}^d \rightarrow \mathbf{R}$ is called convex iff for any two points x, x' and $\beta \in [0,1]$

$$f(\beta\vec{x} + (1 - \beta)\vec{x}') \leq \beta f(\vec{x}) + (1 - \beta)f(\vec{x}')$$



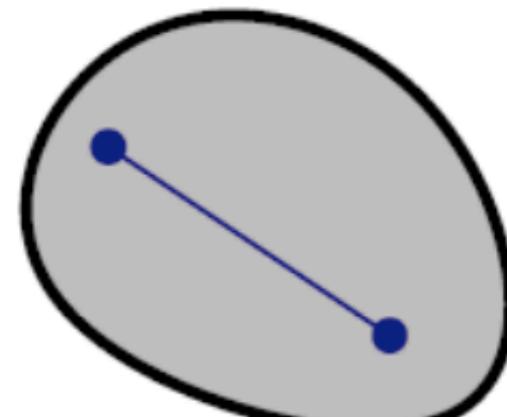
Slide credit: Nakul Verma, Columbia University

Conve xity

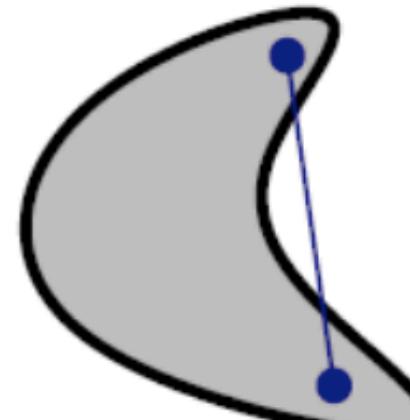
A set $S \subset \mathbf{R}^d$ is called convex iff for any two points $x, x' \in S$ and any $\beta \in [0,1]$

$$\beta\vec{x} + (1 - \beta)\vec{x}' \in S$$

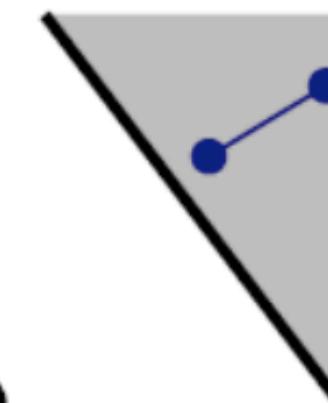
Examples:



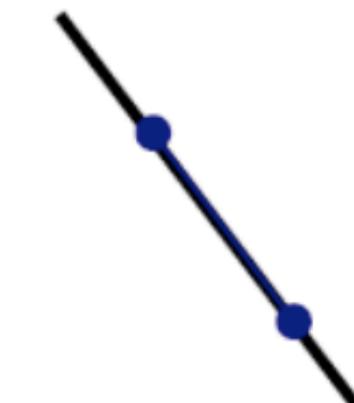
convex



not convex



convex



convex

Slide credit: Nakul Verma, Columbia University

Convex Optimization

A constrained optimization

$$\underset{\vec{x} \in \mathbf{R}^d}{\text{minimize}} \quad f(\vec{x}) \quad (\text{objective})$$

$$\text{subject to: } g_i(\vec{x}) \leq 0 \quad \text{for } 1 \leq i \leq n \quad (\text{constraints})$$

is called convex a convex optimization problem

If:

the objective function $f(\vec{x})$ is convex function, and

the feasible set induced by the constraints g_i is a convex set

Why do we care?

We and find the optimal solution for convex problems efficiently!

Slide credit: Nakul Verma, Columbia University



Back to Duality

Theorem (weak Lagrangian duality):

$$d^* \leq p^*$$

Theorem (strong Lagrangian duality):

If f is convex and for a feasible point x^*

$$g_i(\vec{x}^*) < 0, \text{ or}$$

$g_i(\vec{x}^*) \leq 0$ when g is affine

Then $d^* = p^*$

Slater's condition

Optimization problem:

Minimize: $f(\vec{x})$

Such that: $g_i(\vec{x}) \leq 0$
(for all i)

Lagrange function:

$$L(\vec{x}, \vec{\lambda}) := f(\vec{x}) + \sum_{i=1}^n \lambda_i g_i(\vec{x})$$

Primal:

$$p^* = \min_{\vec{x}} \max_{\lambda_i \geq 0} L(\vec{x}, \vec{\lambda})$$

Dual:

$$d^* := \max_{\lambda_i \geq 0} \min_{\vec{x}} L(\vec{x}, \vec{\lambda})$$

Back to Duality

Observations:

- object function is **convex**
- the constraints are **affine**, inducing a polytope constraint set.

So, SVM is a convex optimization problem
(in fact a **quadratic program**)

Moreover, **strong duality holds**.

Let's examine the dual... the Lagrangian is.

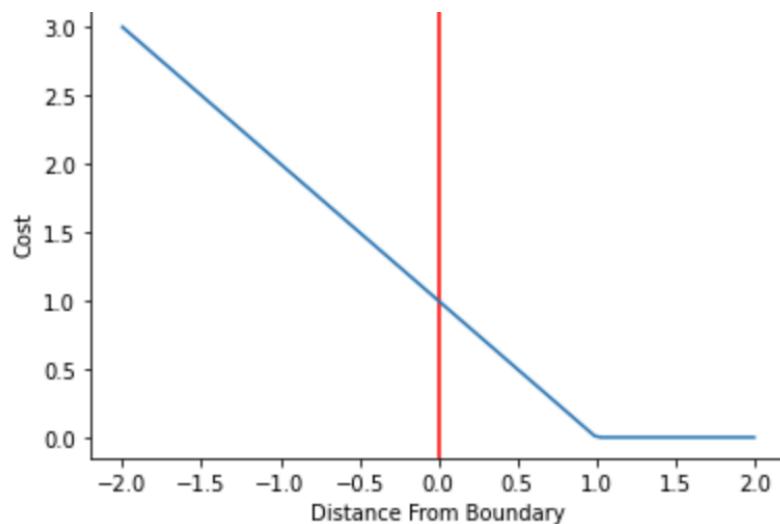
$$L(\vec{w}, b, \vec{\alpha}) = \frac{1}{2} \|\vec{w}\|^2 + \sum_{i=1}^n \alpha_i (1 - y_i (\vec{w} \cdot \vec{x}_i + b))$$

Also known as hinge loss

Slide credit: Nakul Verma, Columbia University



Why hinge loss?



Looks like a hinge

$$L(\vec{w}, b, \vec{x}) = \frac{1}{2} \|\vec{w}\|^2 + \sum_{i=1}^n \alpha_i (1 - y_i (\vec{w} \cdot \vec{x}_i + b))$$

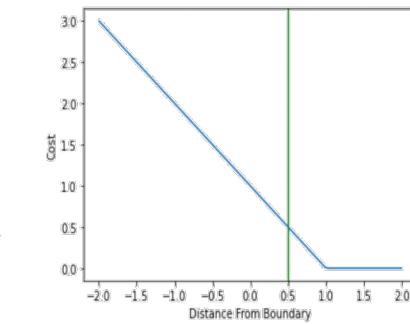
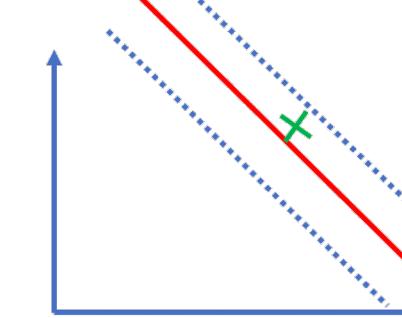
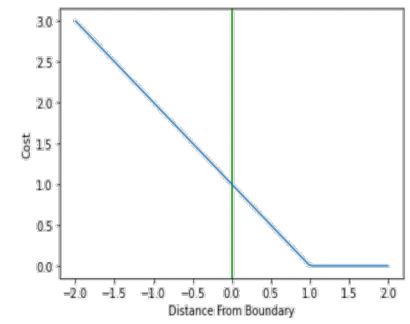
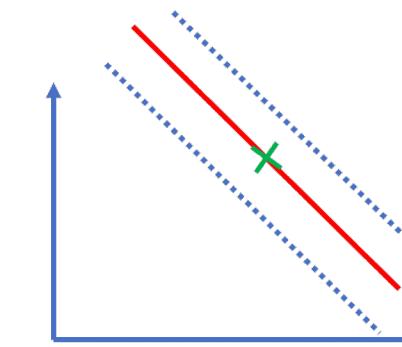
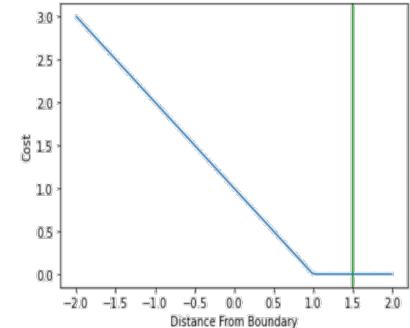
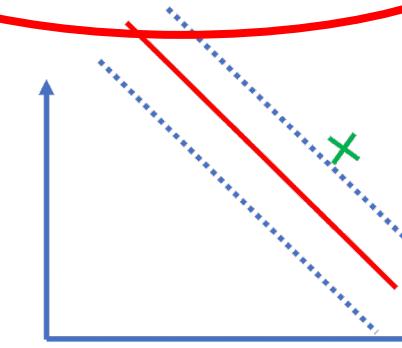
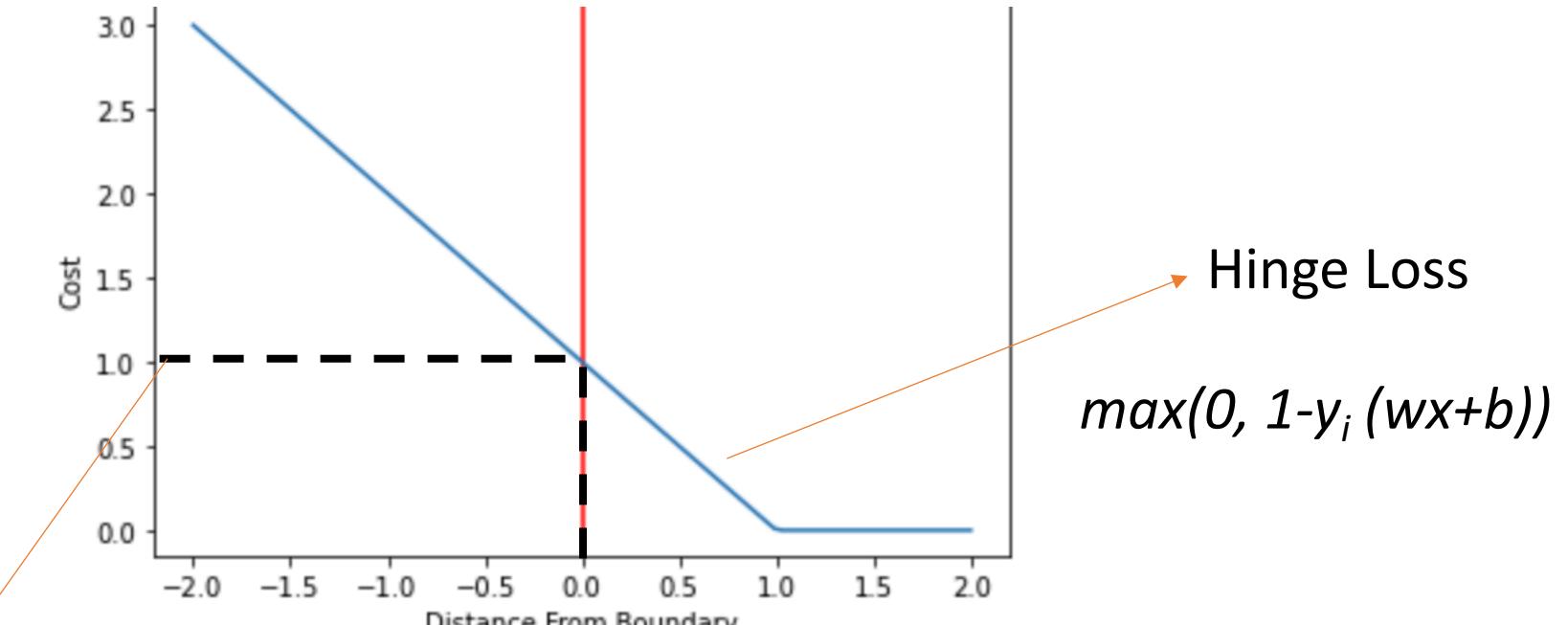


Image source: <https://programmatically.com/understanding-hinge-loss-and-the-svm-cost-function/>

Why hinge loss?

0-1 Loss

$$1[y \neq (wx+b)]$$



Hinge Loss
 $\max(0, 1-y_i(wx+b))$

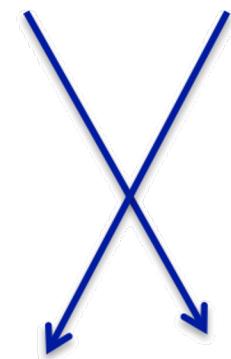
Hinge loss upper bounds 0/1 loss!

It is the tightest convex upper bound on the 0/1 loss

SVM Dual

(Primal)

$$\min_{\vec{w}, b} \max_{\vec{\alpha} \geq 0} \frac{1}{2} \|\vec{w}\|^2 - \sum_j \alpha_j [(\vec{w} \cdot \vec{x}_j + b) y_j - 1]$$



Swap min and max

(Dual)

$$\max_{\vec{\alpha} \geq 0} \min_{\vec{w}, b} \frac{1}{2} \|\vec{w}\|^2 - \sum_j \alpha_j [(\vec{w} \cdot \vec{x}_j + b) y_j - 1]$$

Slater's condition from convex optimization guarantees that these two optimization problems are equivalent!

Slide credit: David Sontag, MIT

Solving using KKT conditions

$$\max_{\vec{\alpha} \geq 0} \min_{\vec{w}, b} \frac{1}{2} \|\vec{w}\|^2 - \sum_i \alpha_j [(\vec{w} \cdot \vec{x}_j + b) y_j - 1]$$

Can solve for optimal w, b as function of α :

$$\frac{\partial L}{\partial w} = w - \sum_j \alpha_j y_j x_j \rightarrow w = \sum_j \alpha_j y_j x_j$$

$$\frac{\partial L}{\partial b} = - \sum_j \alpha_j y_j \rightarrow \sum_j \alpha_j y_j = 0$$

Karush-Kuhn-Tucker
Conditions

Substituting these values back in (and simplifying), we obtain:

$$\max_{\vec{\alpha} \geq 0, \sum_j \alpha_j y_j = 0} \sum_j \alpha_j - \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j (\vec{x}_i \cdot \vec{x}_j)$$

Sums over all training examples scalars dot product

Slide credit: David Sontag, MIT

Solving using KKT conditions

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:

$$\alpha_k \geq 0 \quad \forall k \quad \sum_{k=1}^R \alpha_k y_k = 0$$

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.

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

$$y_i(\mathbf{x}_i \bullet \mathbf{w} + b) - 1 = 0$$

$$b = -y_i(\mathbf{x}_i \bullet \mathbf{w}) - 1$$

Then classify with:

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

Solving Convex Optimization Problems

- Every local optima is a global optima in a convex optimization problem.
- Example convex problems:
 - Linear programs, quadratic programs,
 - Conic programs, semi-definite program.
- Several solvers exist to find the optima:
 - CVX, SeDuMi, C-SALSA, ...
- We can use a simple ‘descent-type’ algorithm for finding the minima!

Slide credit: Nakul Verma, Columbia University



Gradient Descent

Theorem (Gradient Descent):

Given a smooth function $f : \mathbf{R}^d \rightarrow \mathbf{R}$

Then, for any $\vec{x} \in \mathbf{R}^d$ and $\vec{x}' := \vec{x} - \eta \nabla_x f(\vec{x})$

For sufficiently small $\eta > 0$, we have: $f(\vec{x}') \leq f(\vec{x})$

Can derive a **simple algorithm** (the projected Gradient Descent):

Initialize \vec{x}^0

for $t = 1, 2, \dots$ do

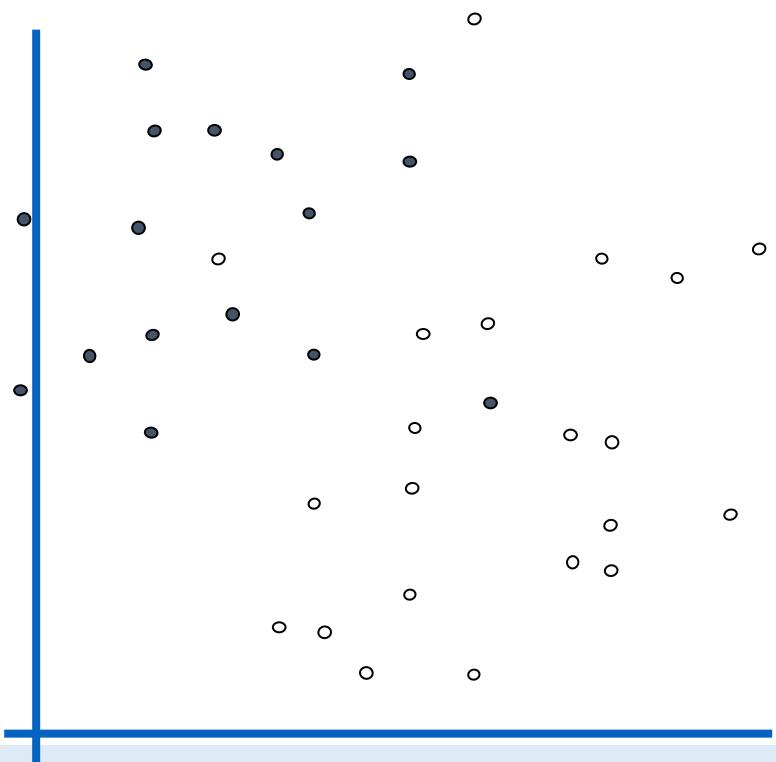
$$\vec{x}^{t+1} := \vec{x}^{t-1} - \eta \nabla_x f(\vec{x}^{t-1}) \quad (\text{step in the gradient direction})$$

$$\vec{x}^t := \Pi_{g_i}(\vec{x}^{t+1}) \quad (\text{project back onto the constraints})$$

terminate when no progress can be made, ie, $|f(\vec{x}^t) - f(\vec{x}^{t-1})| \leq \epsilon$

Non-separable Data

- denotes +1
- denotes -1



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

From hard-margin SVMs to
soft-margin SVMs...

SVM for Noisy Data (C-SVM, Soft-Margin SVM)

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

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

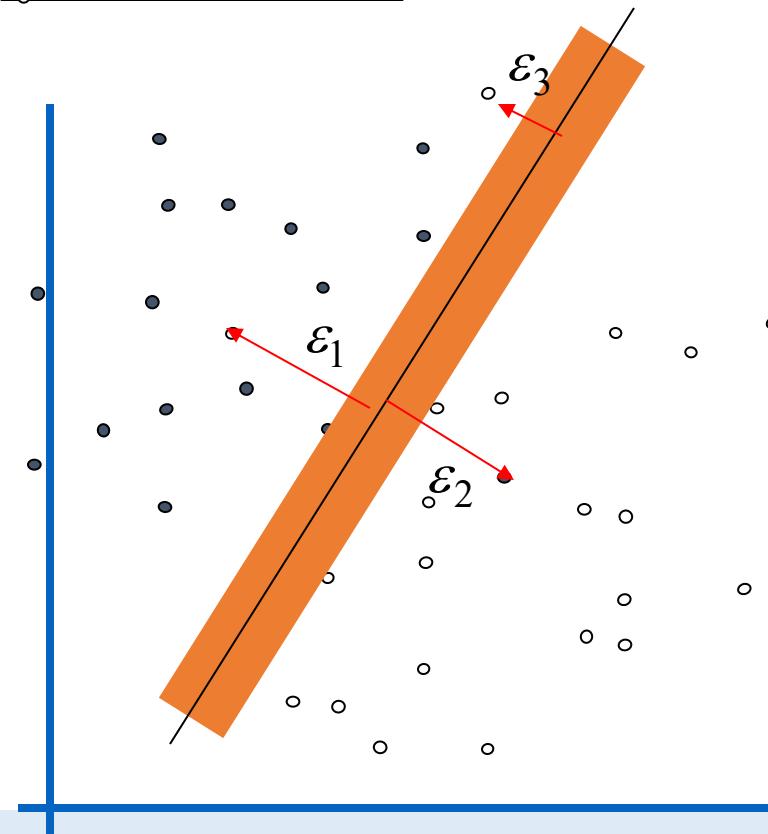
$$y_2(\vec{w} \cdot \vec{x}_2 + b) \geq 1 - \varepsilon_2, \varepsilon_2 \geq 0$$

...

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

Balance the trade off between margin and classification errors

- denotes +1
- denotes -1

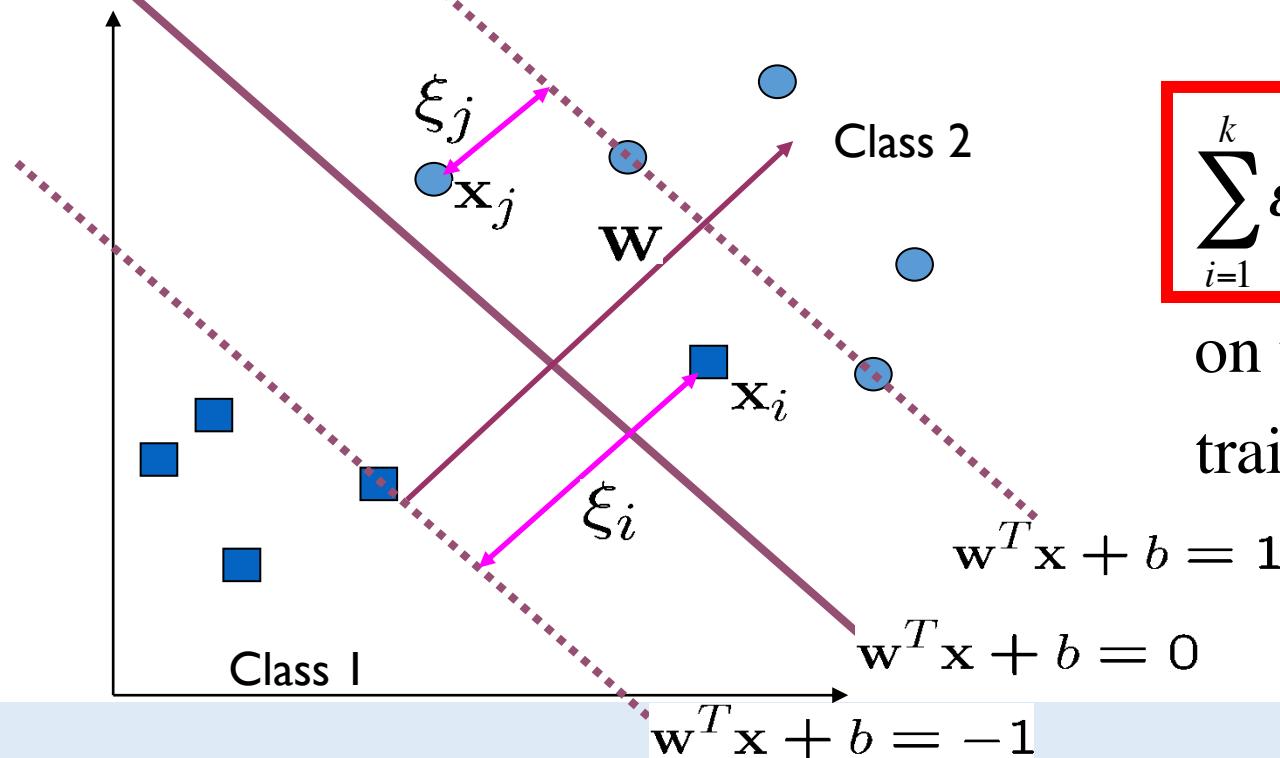


SVM for Noisy Data (C-SVM, Soft-Margin SVM)

$\varepsilon_i \geq 1 \Leftrightarrow y_i(\mathbf{w}^T \mathbf{x}_i + b) < 0$, i.e., misclassification

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

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



$\sum_{i=1}^k \varepsilon_i$ is an upper bound

on the number of
training errors.

SVM for Noisy Data (C-SVM, Soft-Margin SVM)

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

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

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

α_i β_i

Lagrangian multipliers

The diagram consists of two mathematical expressions. The top expression is $y_i(x_i \bullet w + b) - 1 + \varepsilon_i \geq 0$, where w and b are underlined with blue arrows pointing to the terms α_i and β_i in the bottom expression. The bottom expression is $\varepsilon_i \geq 0$. To the right of the bottom expression, the text "Lagrangian multipliers" is written next to a blue double-headed arrow between α_i and β_i .

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

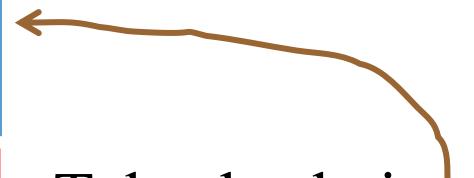
SVM for Noisy Data (C-SVM, Soft-Margin SVM)

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

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

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

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



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

Karush-Kuhn-Tucker Conditions

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

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

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

SVM for Noisy Data (C-SVM, Soft-Margin SVM)

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:

$$0 \leq \alpha_k \leq c \quad \forall k$$

$$\sum_{k=1}^R \alpha_k y_k = 0$$

Compare this with the hard-margin SVM dual – what is different?

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:

$$\alpha_k \geq 0 \quad \forall k$$

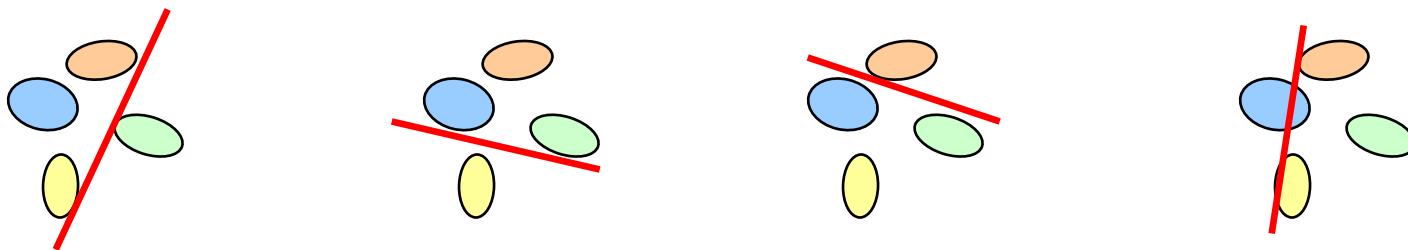
$$\sum_{k=1}^R \alpha_k y_k = 0$$

Multi-class Classification with SVMs

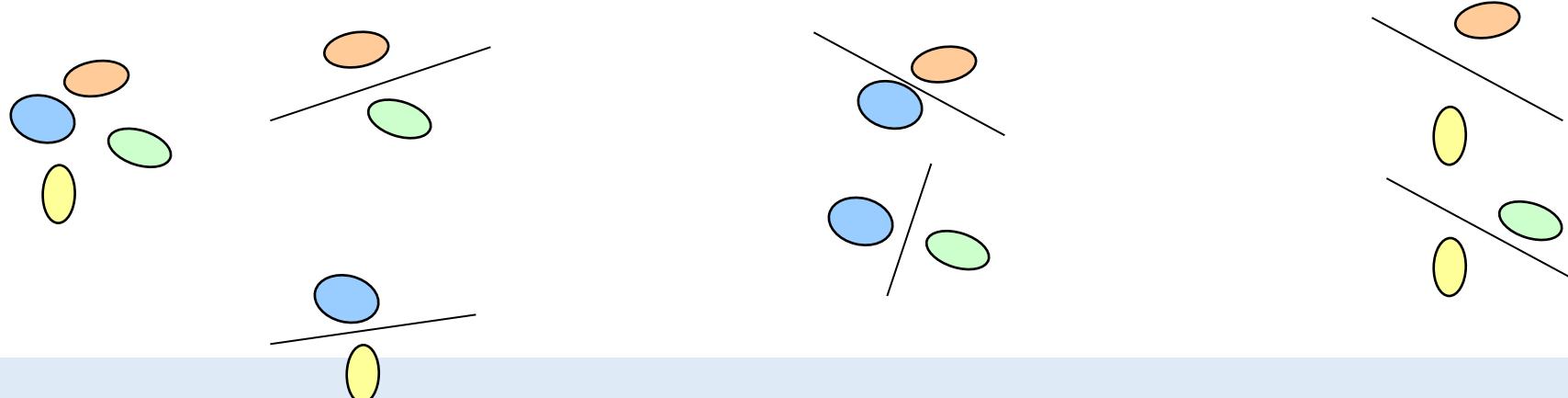
- SVMs can only handle two-class outputs.
- What can be done?
- Answer: with output arity N, learn N SVM's
 - SVM 1 learns "Output==1" vs "Output != 1"
 - SVM 2 learns "Output==2" vs "Output != 2"
 - :
 - SVM N learns "Output==N" vs "Output != N"
- Then to predict the output for a new input, just predict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

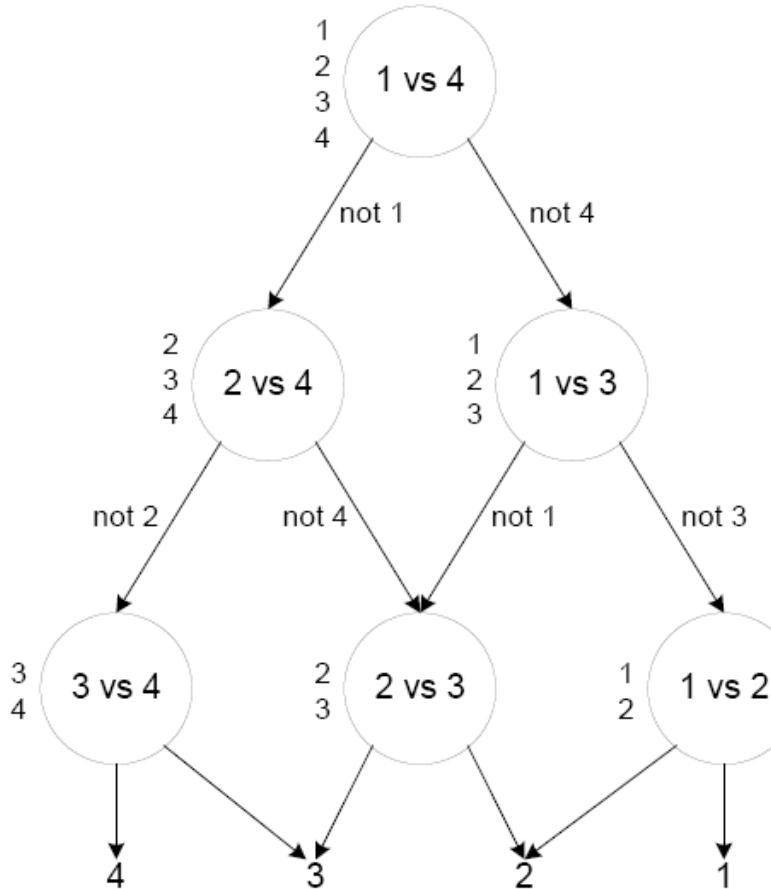
One- versus-all



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/>