University of Colorado **Boulder**

Department of Computer Science

CSCI 5622: Machine Learning

Chenhao Tan

Lecture 14: SVM

Slides adapted from Chris Ketelsen, Jordan Boyd-Graber, and Noah Smith

Administrivia

- HW3 is due on Friday
- Final project proposal is due on Friday

Outline

- A little bit of history
- Linear classifiers and margin
- Hard-margin SVM
- Soft-margin SVM

Outline

- A little bit of history
- Linear classifiers and margin
- Hard-margin SVM
- Soft-margin SVM

Outline for CSCI 5622
We've already covered stuff in blue!

- Problem formulations: classification, regression
- Supervised techniques: decision trees, nearest neighbors, perceptron, linear models, neural networks, support vector machine, kernel methods
- Unsupervised techniques: clustering, linear dimensionality reduction, topic modeling
- "Meta-techniques": ensembles, expectation-maximization, variational inference
- Understanding ML: limits of learning, practical issues, bias & fairness
- Recurring themes: (stochastic) gradient descent

History lesson

- 1962: Rosenblatt, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms
    - First neuron-based learning algorithm
    - “Could learning anything that you could program”

History lesson

- 1962: Rosenblatt, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms
    - First neuron-based learning algorithm
    - “Could learning anything that you could program”
- 1969: Minsky & Papert, Perceptron: An Introduction to Computational Geometry
    - First real complexity analysis
    - Showed, in principle, many things that perceptrons can’t learn to do
    - Shut down any interest in neural networks

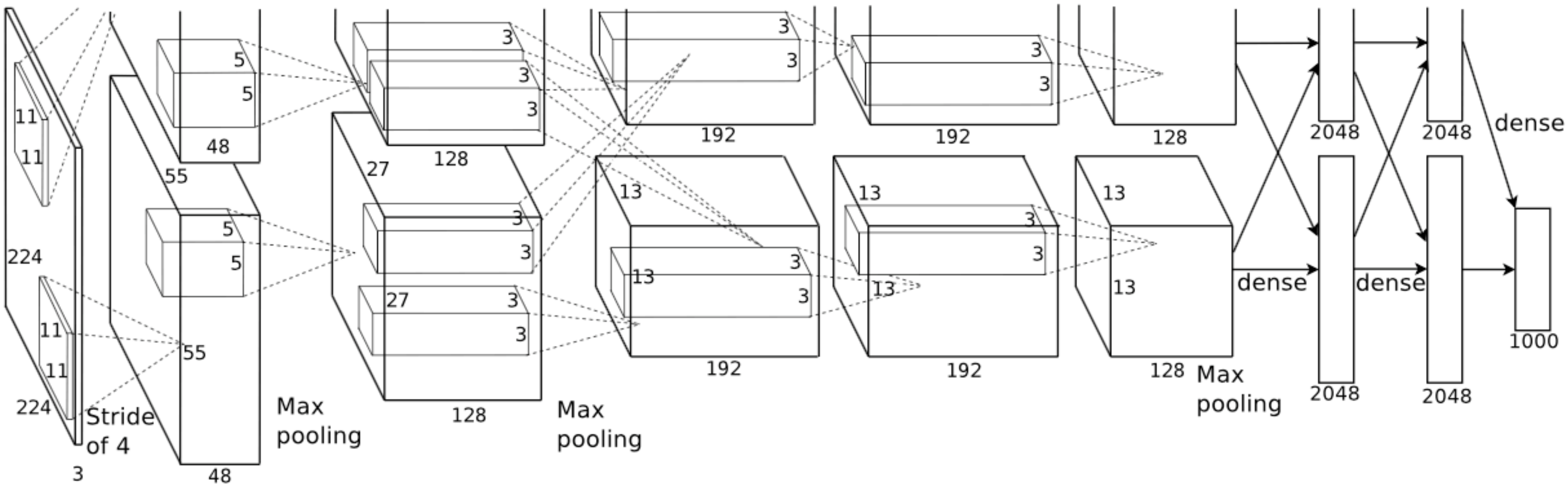
History lesson

- 1999-2005
  - Shift to Bayesian Methods
    - Best ideas from neural networks
    - Direct statistical computing
  - Support Vector Machines
    - Nice mathematical properties
    - Kernel tricks
  - A few people still playing with NNs
    - Bengio
    - Hinton
    - LeCun

History lesson

- 2005-2010
  - Core group continues to make improvements
  - Various tricks to make NNs practical
- 2010-present
  - BOOM!

AlexNet



Krizhevsky et al. [2012]

Question: Why? What made neural networks amazing again?

- Massive datasets
- Computing power
- Algorithmic improvements

History lesson

Machine learning has a short history, but seems cyclic.
What is next?

Outline

- A little bit of history
- Linear classifiers and margin
- Hard-margin SVM
- Soft-margin SVM

Linear classifiers

We have already seen several:

- Naïve Bayes
- Logistic regression
- Perceptron

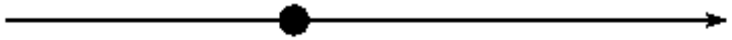
Definition: A linear classifier makes decisions by computing a linear combination of features of the form $\boldsymbol{w}^T\boldsymbol{x} + b$ and then classifies based on

$$\boldsymbol{w}^T\boldsymbol{x} + b \geq 0.$$

The decision boundary between the two classes is defined by $\boldsymbol{w}^T\boldsymbol{x} + b = 0$.

We estimate the weights and bias using the training data.

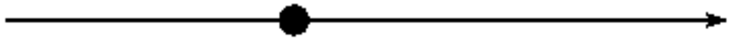
A linear classifier in 1D



- A linear classifier in 1D is a point x described by the equation $\boldsymbol{w}_1 \boldsymbol{x}_1 = -b$.

$w_1 x_1 + b = 0$

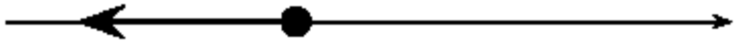
A linear classifier in 1D



- A linear classifier in 1D is a point x described by the equation $\boldsymbol{w}_1\boldsymbol{x}_1 = -b$.
- $\boldsymbol{x}_1 = -b/\boldsymbol{w}_1$

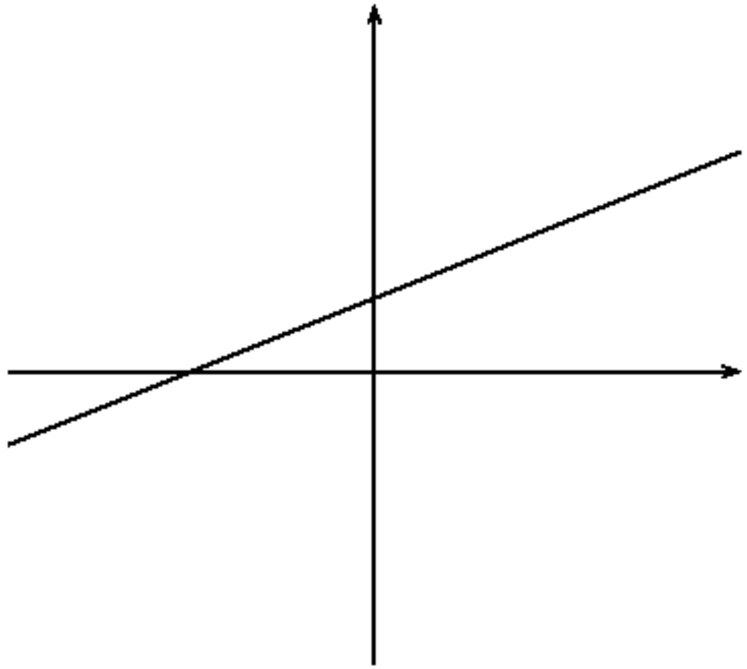
- A linear classifier in 1D is a point x described by the equation $\boldsymbol{w}_1\boldsymbol{x}_1 = -b$.
- $\boldsymbol{x}_1 = -b/\boldsymbol{w}_1$
- Points $(\boldsymbol{x}_1)$ with $\boldsymbol{w}_1\boldsymbol{x}_1 \geq -b$ are in the positive class.

A linear classifier in 1D



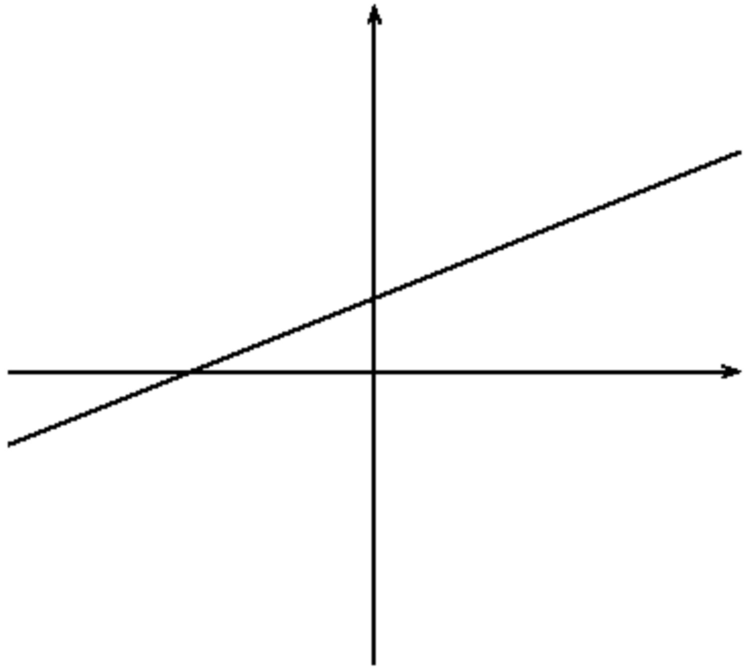
- A linear classifier in 1D is a point x described by the equation $\boldsymbol{w}_1 \boldsymbol{x}_1 = -b$.
- $\boldsymbol{x}_1 = -b/\boldsymbol{w}_1$
- Points $(\boldsymbol{x}_1)$ with $\boldsymbol{w}_1 \boldsymbol{x}_1 \geq -b$ are in the positive class.
- Points $(\boldsymbol{x}_1)$ with $\boldsymbol{w}_1 \boldsymbol{x}_1 < -b$ are in the negative class.

A linear classifier in 2D



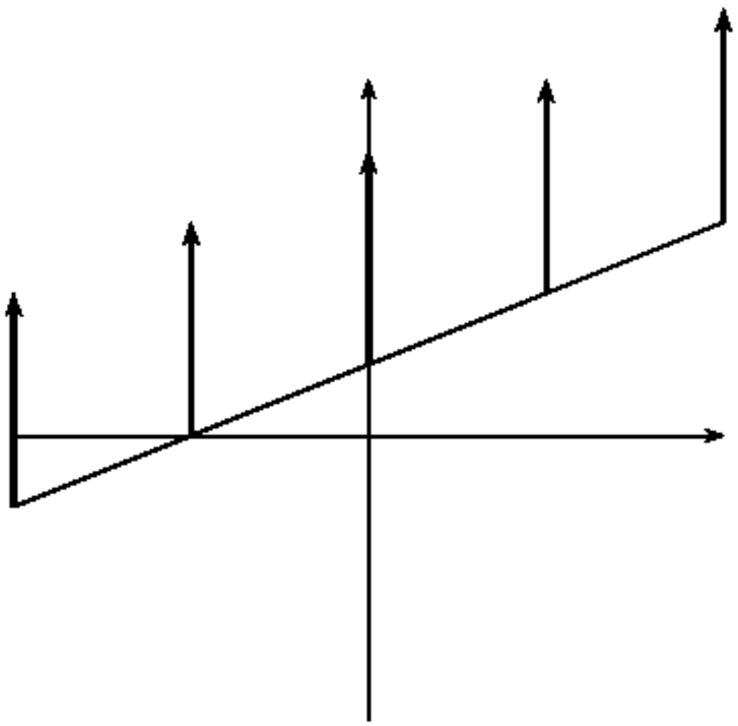
- A linear classifier in 2D is a line described by the equation $w_1 x_1 + w_2 x_2 = -b$.

A linear classifier in 2D



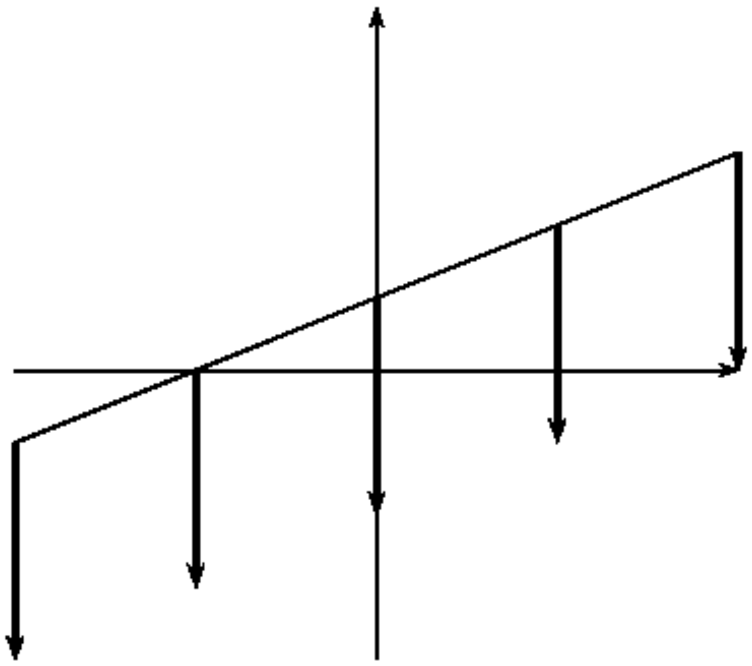
- A linear classifier in 2D is a line described by the equation $\boldsymbol{w}_1\boldsymbol{x}_1 + \boldsymbol{w}_2\boldsymbol{x}_2 = -b$.
- Example for a 2D linear classifier

A linear classifier in 2D



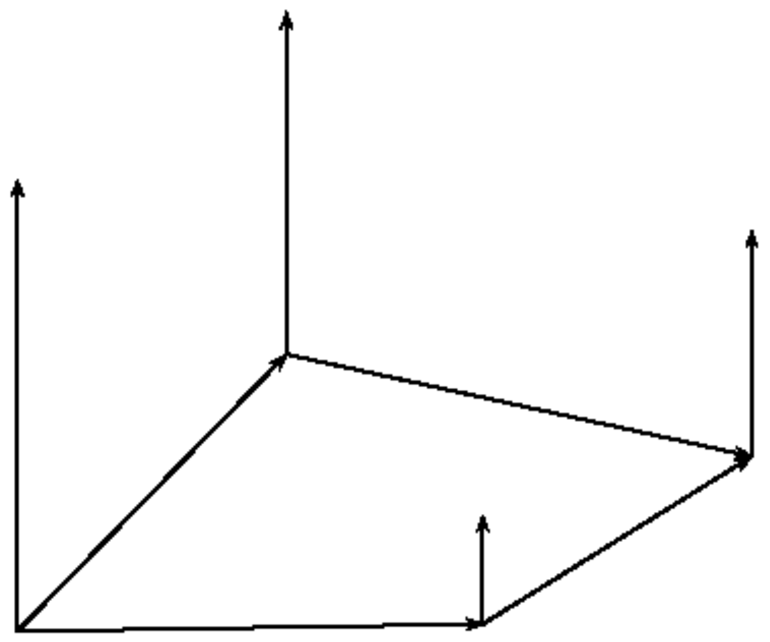
- A linear classifier in 2D is a line described by the equation $w_1 x_1 + w_2 x_2 = -b$.
- Example for a 2D linear classifier
- Points (x_1, x_2) with $w_1 x_1 + w_2 x_2 \geq -b$ are in the positive class.

A linear classifier in 2D



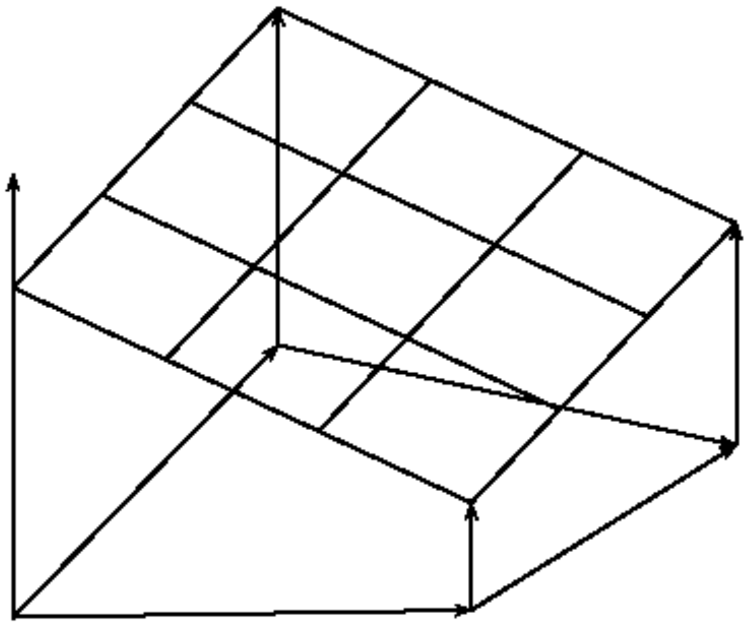
- A linear classifier in 2D is a line described by the equation $w_1 x_1 + w_2 x_2 = -b$.
- Example for a 2D linear classifier
- Points (x_1, x_2) with $w_1 x_1 + w_2 x_2 \geq -b$ are in the positive class.
- Points (x_1, x_2) with $w_1 x_1 + w_2 x_2 x_2 < -b$ are in the negative class.

A linear classifier in 3D



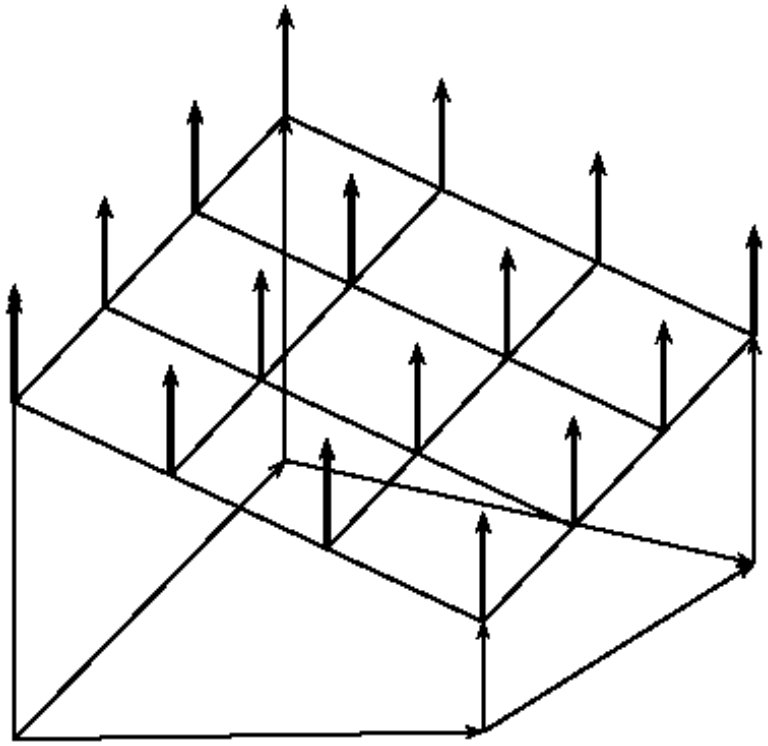
- A linear classifier in 3D is a plane described by the equation $\boldsymbol{w}_1\boldsymbol{x}_1 + \boldsymbol{w}_2\boldsymbol{x}_2 + \boldsymbol{w}_3\boldsymbol{x}_3 = -b$.

A linear classifier in 3D



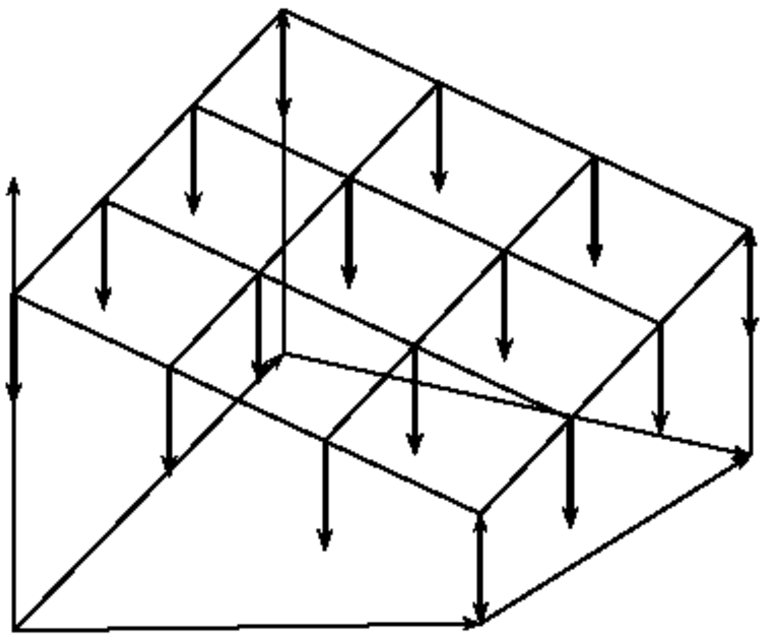
- A linear classifier in 3D is a plane described by the equation $\boldsymbol{w}_1\boldsymbol{x}_1 + \boldsymbol{w}_2\boldsymbol{x}_2 + \boldsymbol{w}_3\boldsymbol{x}_3 = -b$.
- Example for a 3D linear classifier

A linear classifier in 3D



- A linear classifier in 3D is a plane described by the equation $w_1x_1 + w_2x_2 + w_3x_3 = -b$.
- Example for a 3D linear classifier
- Points (x_1, x_2, x_3) with $w_1x_1 + w_2x_2 + w_3x_3 \geq -b$ are in the class c.

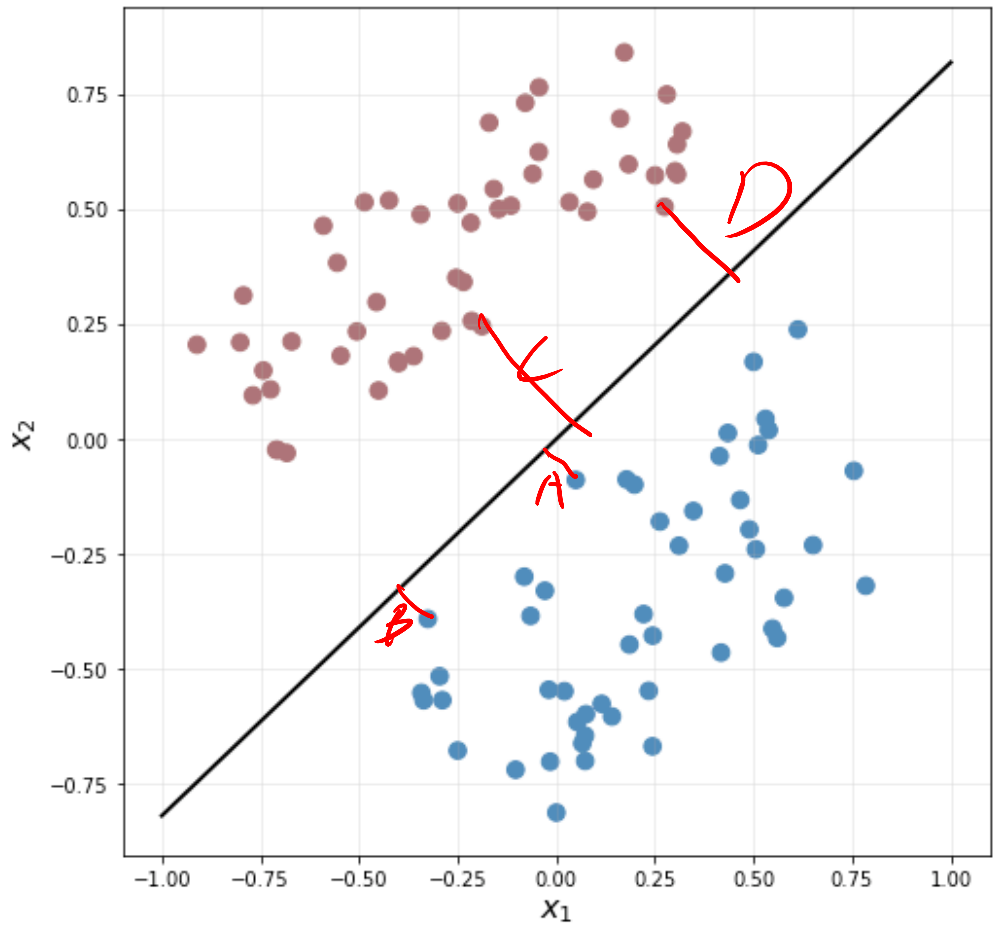
A linear classifier in 3D



- A linear classifier in 3D is a plane described by the equation $w_1x_1 + w_2x_2 + w_3x_3 = -b$.
- Example for a 3D linear classifier
- Points (x_1, x_2, x_3) with $w_1x_1 + w_2x_2 + w_3x_3 \geq -b$ are in the class c.
- Points (x_1, x_2, x_3) with $w_1x_1 + w_2x_2 + w_3x_3 < -b$ are in the complement class $\overline{c}$.

Pictorial definition of margin

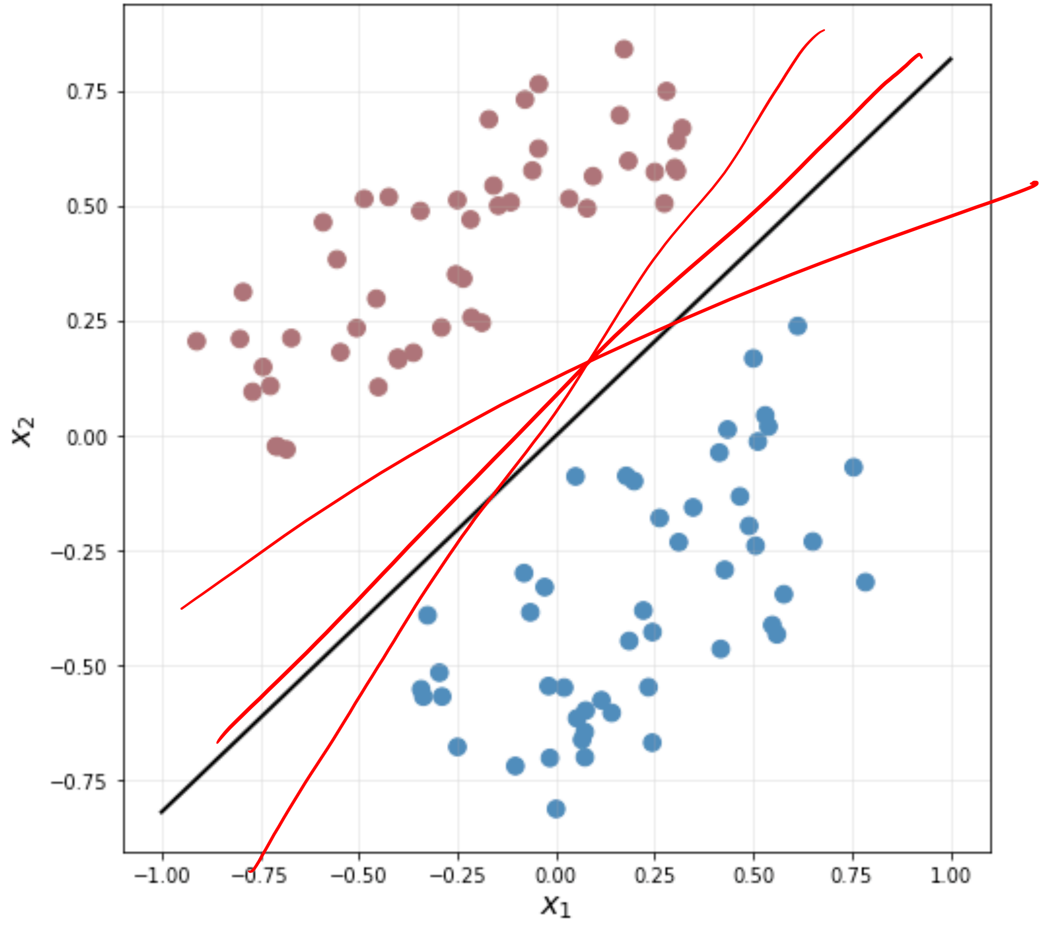
Assuming that the dataset is separable, margin is defined as the smallest distance from any data point to the decision boundary.



Pictorial definition of margin

Assuming that the dataset is separable, margin is defined as the smallest distance from any data point to the decision boundary.

What is the margin of the classifier on the right?



Outline

- A little bit of history
- Linear classifiers and margin
- Hard-margin SVM
- Soft-margin SVM

Which hyperplane?

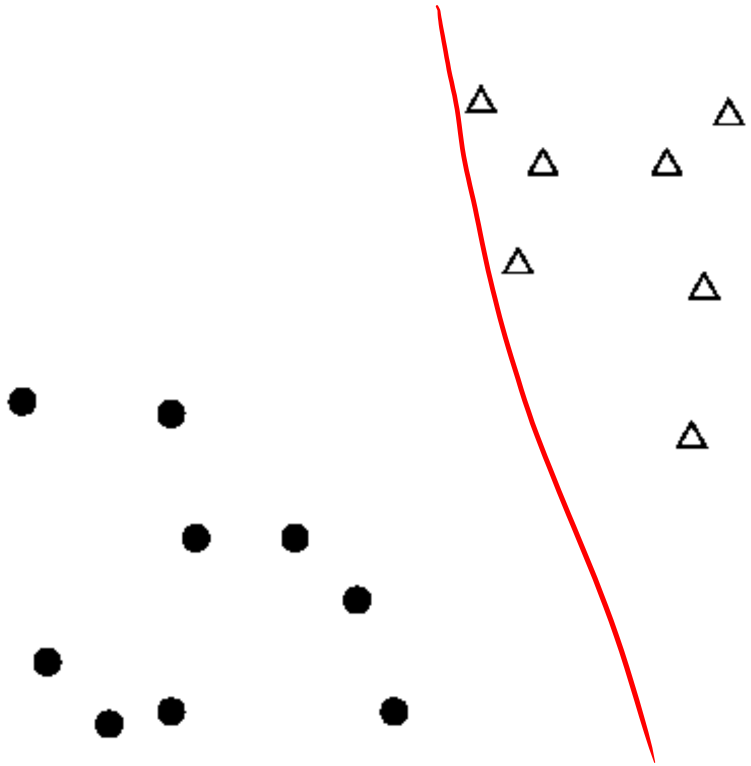
- For linearly separable training sets: there are **infinitely** many separating hyperplanes.
- They all separate the training set perfectly . . .
- . . . but they behave differently on test data.
- Error rates on new data are low for some, high for others.
- How do we find a low-error separator?

SVMs: A kind of large-margin classifier

Find a decision boundary between two classes that is maximally far from any point in the training data (possibly discounting some points as outliers or noise).

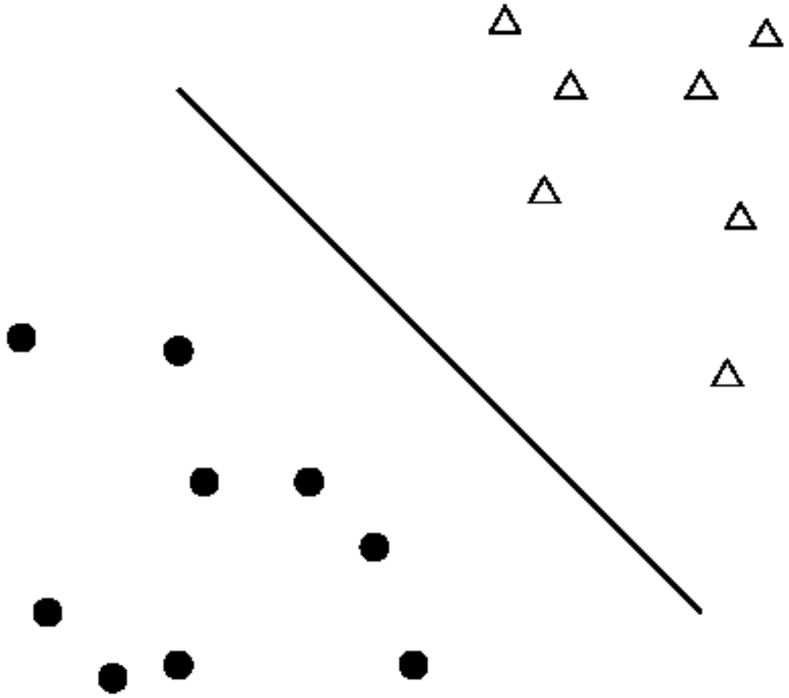
Support Vector Machines

- 2-class training data



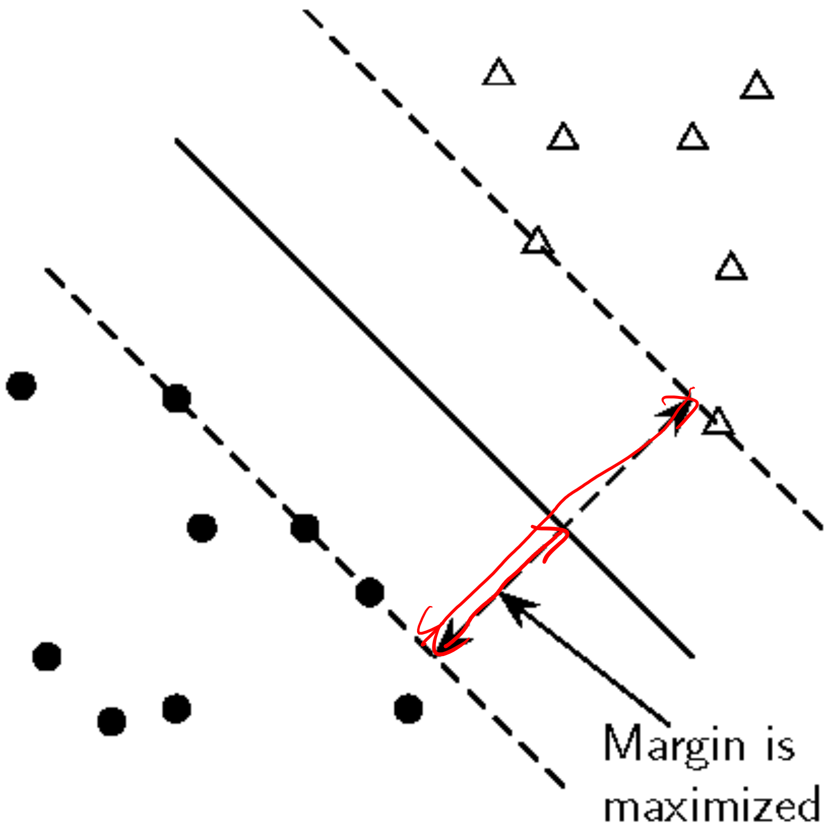
Support Vector Machines

- 2-class training data
- decision boundary $\rightarrow$ **linear separator**



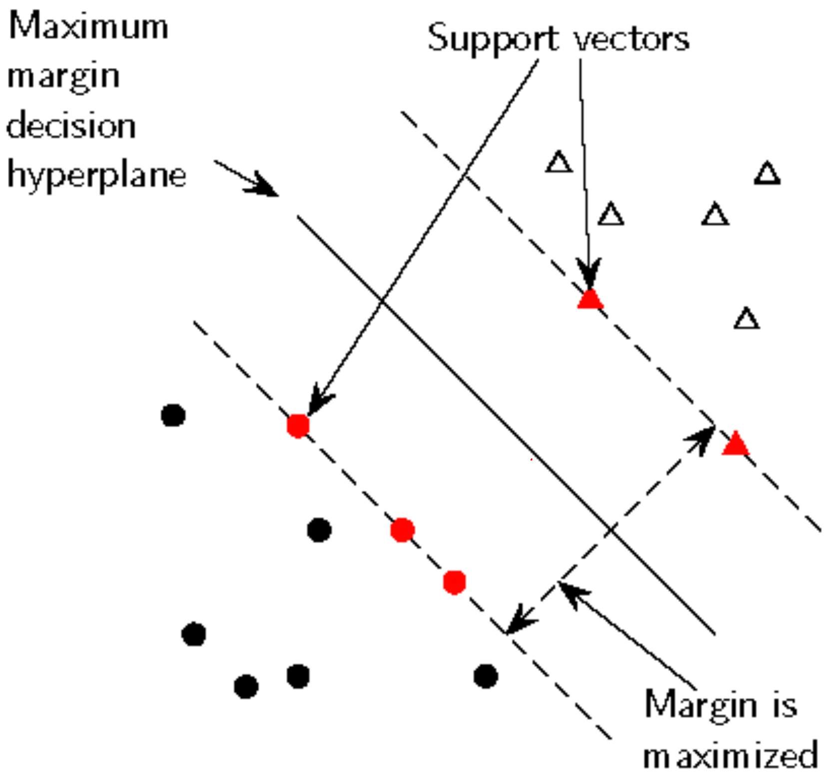
Support Vector Machines

- 2-class training data
- decision boundary → **linear separator**
- criterion: being maximally far away from any data point → determines classifier **margin**



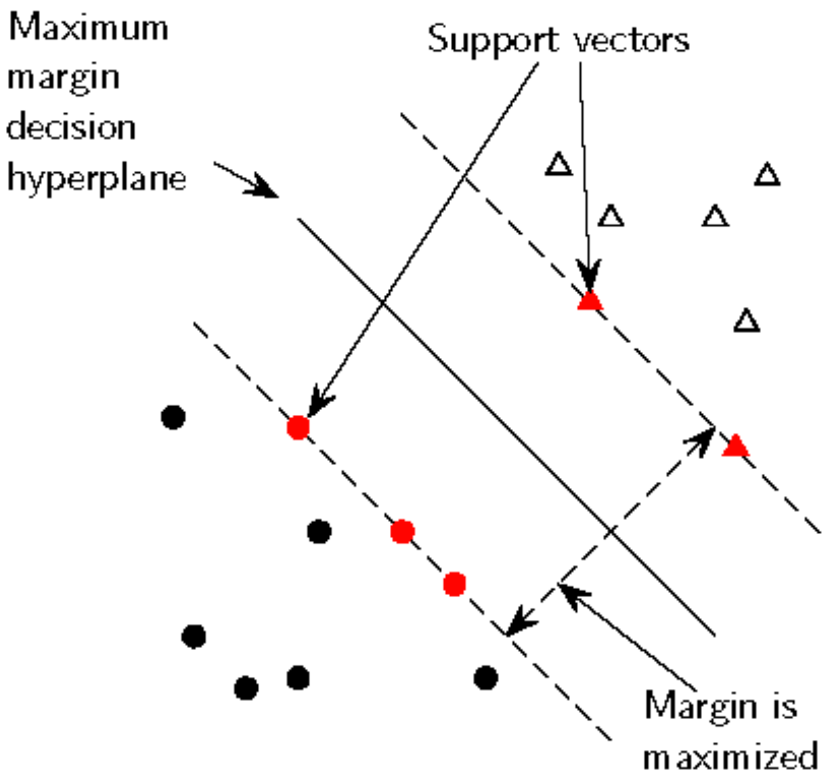
Support Vector Machines

- 2-class training data
- decision boundary $\rightarrow$ **linear separator**
- criterion: being maximally far away from any data point $\rightarrow$ determines classifier **margin**
- linear separator position defined by **support vectors**
- other points have no impact on the decision boundary



Why maximize the margin?

- Points near decision surface $\rightarrow$ uncertain classification decisions
- A classifier with a large margin is always confident
- Gives classification safety margin (measurement or variation)
- Increased ability to correctly generalize to test data



Equation

- Equation of a hyperplane

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$$

$kw \cdot x + kb = 0 \quad k \neq 0$

- Distance of a point to hyperplane

$$\frac{|\boldsymbol{w} \cdot \boldsymbol{x} + b|}{||\boldsymbol{w}||}$$

- The margin ρ is given by

$|wx+b| = 1$

$$\rho \equiv \min_{(\boldsymbol{x},y)\in S} \frac{|\boldsymbol{w} \cdot \boldsymbol{x} + b|}{||\boldsymbol{w}||} \equiv \frac{1}{||\boldsymbol{w}||}$$

Equation

- Equation of a hyperplane

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$$

$k \cdot w \cdot x + kb = 0$

- Distance of a point to hyperplane

$$\frac{|w \cdot x + b|}{\sqrt{\sum_i w_i^2}}$$

$$\frac{|\boldsymbol{w} \cdot \boldsymbol{x} + b|}{||\boldsymbol{w}||}$$

$\max_{w,b} \frac{1}{||w||} \Leftrightarrow \min ||w|| \Leftrightarrow \min ||w||^2$

s.t. $y_i(w \cdot x + b) \geq 1$

$\min \frac{1}{2}||w||^2$

s.t. $y_i(w \cdot x + b) \geq 1$

- The margin ρ is given by

$$\rho \equiv \min_{(\boldsymbol{x},\boldsymbol{y}) \in S} \frac{|\boldsymbol{w} \cdot \boldsymbol{x} + b|}{||\boldsymbol{w}||} \equiv \frac{1}{||\boldsymbol{w}||}$$

This is because for any point on the marginal hyperplane, we can let $|\boldsymbol{w} \cdot x + b| = 1$, and we would like to maximize the margin ρ.

Optimization Problem

$y = \begin{cases} 1 & wx+b > 0 \\ -1 & wx+b < 0 \end{cases}$

We want to find a weight vector w and bias b that optimize

$$\min_{w,b} \frac{1}{2}||w||^2$$

Σw_i^2

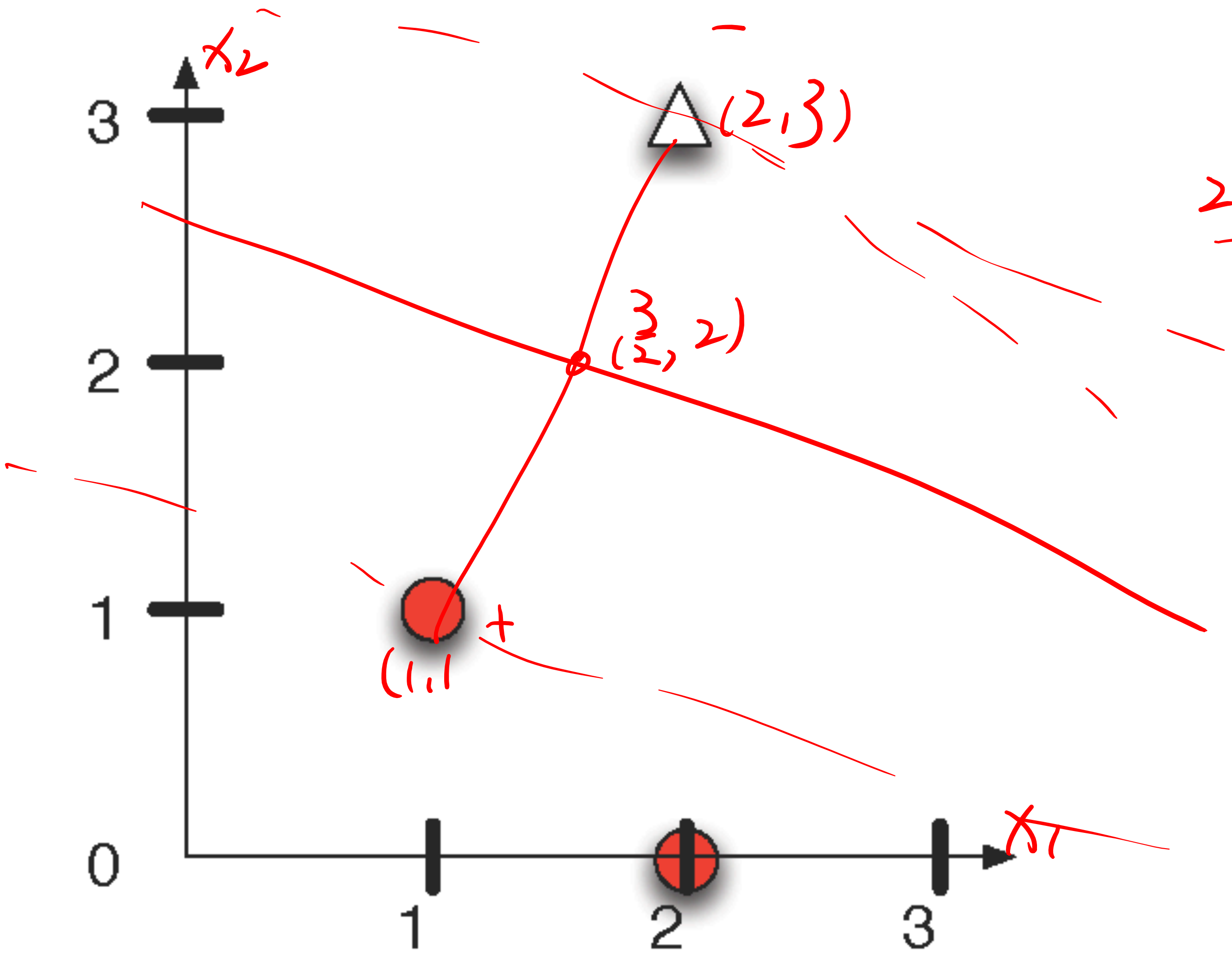
subject to $y_i(w \cdot x_i + b) \geq 1, \forall i \in [1, n]$.

$y_i \in \{-1, 1\}$ $\quad y_i(wx+b) = 1$

$wx+b = 0$

b

Find the maximum margin hyperplane



$w_1x_1 + w_2x_2 + b = 0$

$2x_1 + 4x_2 - 11 = 0$

$w_1 = 2 \quad w_2 = 4$

$b = -11$

$w_1 + w_2 + b = 1$

$2w_1 + 3w_2 + b = -1$

-5

5

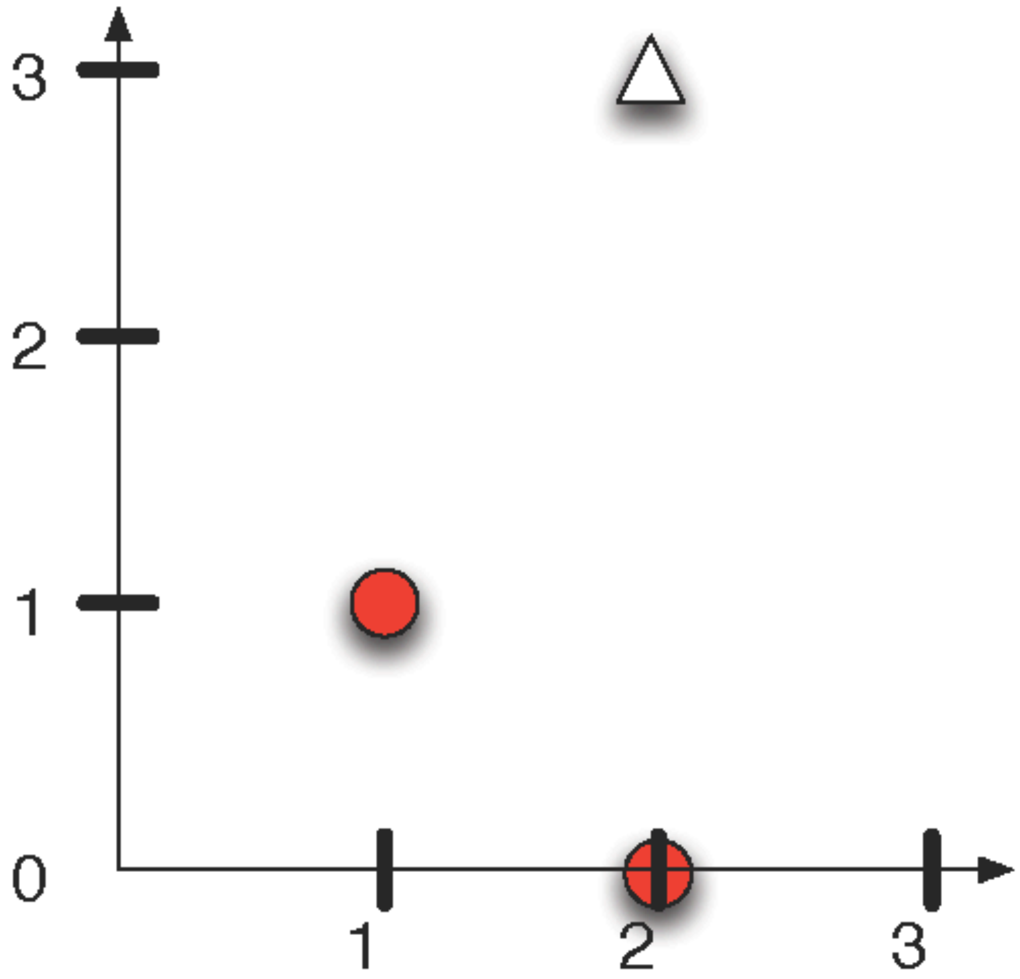
$\frac{3}{2}w_1 + 2w_2 + b = 0$

$w_1 = -\frac{2}{5}$

$w_2 = -\frac{4}{5}$

$b = \frac{11}{5}$

Find the maximum margin hyperplane



Which are the support vectors?

Walk through example: building an SVM over the data shown

Working geometrically:

- Set up system of equations

Walk through example: building an SVM over the data shown

Working geometrically:

- Set up system of equations

$$
\begin{aligned}
w_1 + w_2 + b &= -1 \\
\frac{3}{2}w_1 + 2w_2 + b &= 0 \\
2w_1 + 3w_2 + b &= +1
\end{aligned}
$$

Walk through example: building an SVM over the data shown

Working geometrically:

- Set up system of equations

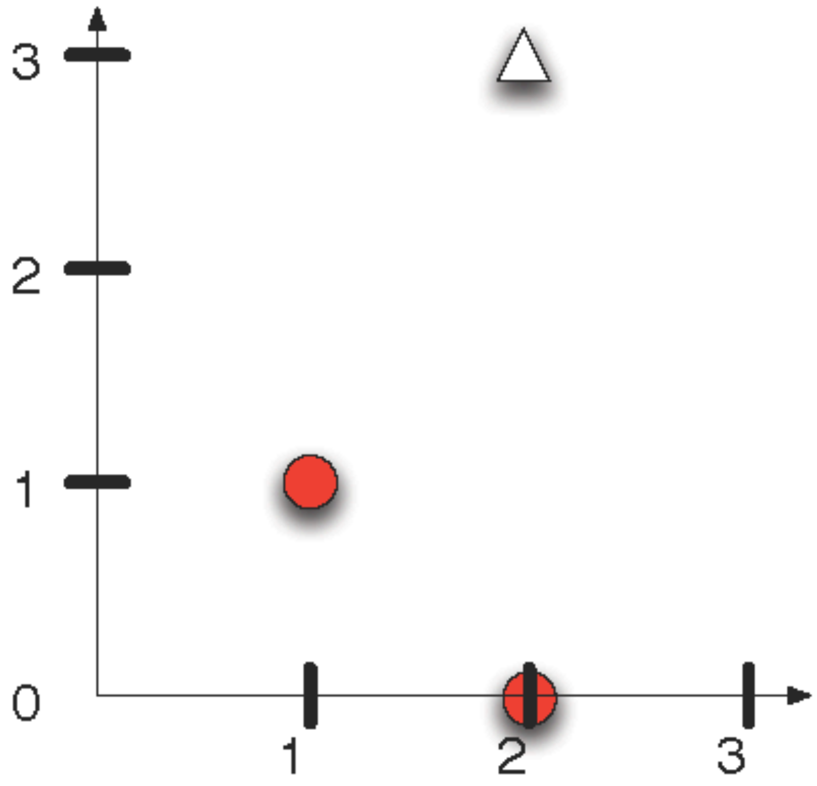
$$w_1 + w_2 + b = -1$$

$$\frac{3}{2}w_1 + 2w_2 + b = 0$$

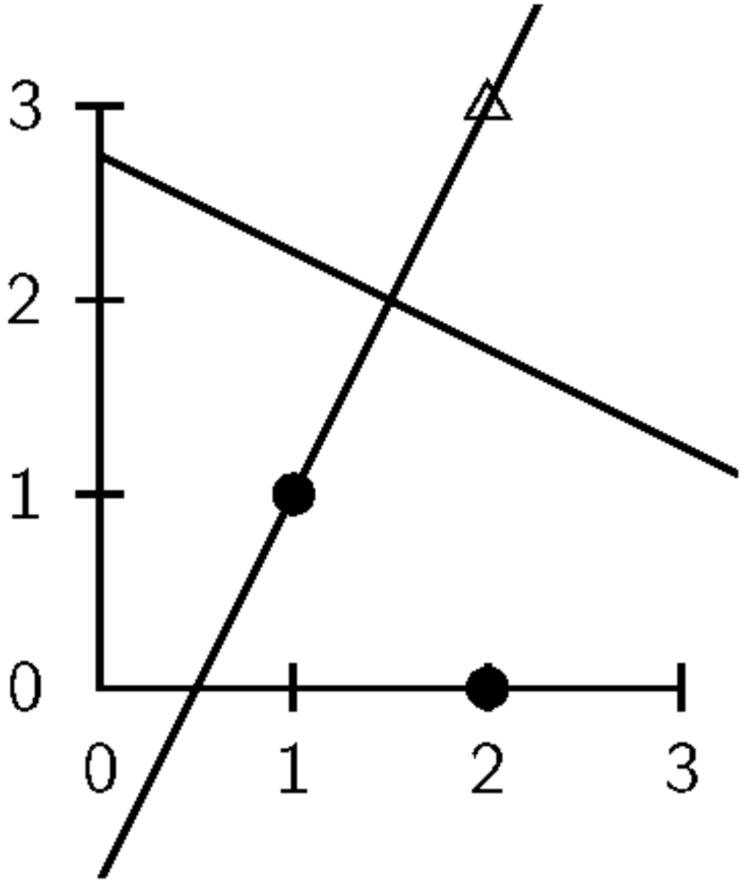
$$2w_1 + 3w_2 + b = +1$$

The SVM decision boundary is:

$$0 = \frac{2}{5}x + \frac{4}{5}y - \frac{11}{5}$$

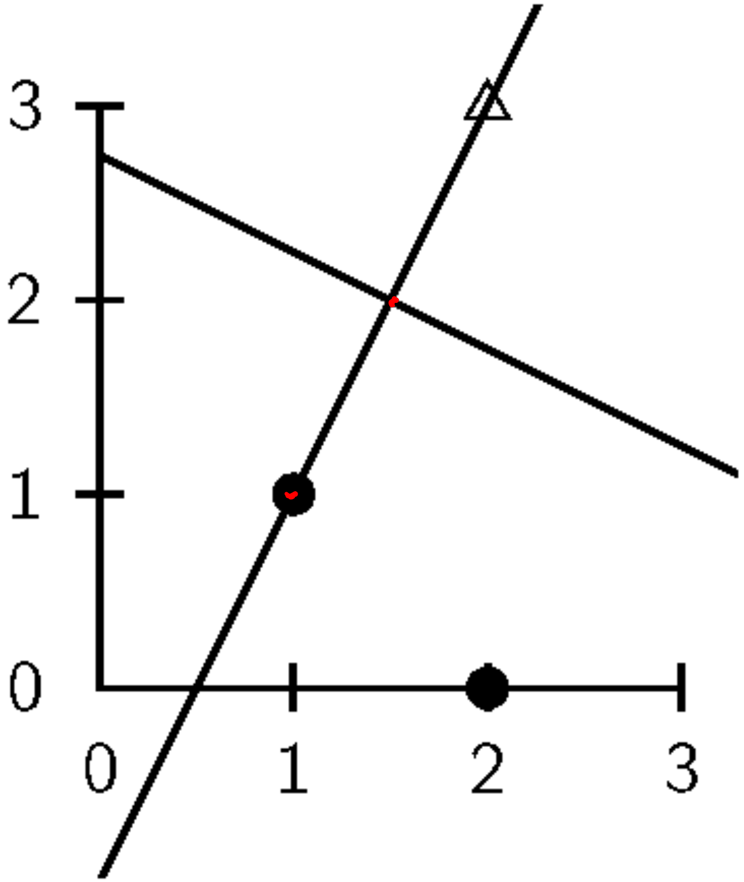


Canonical form



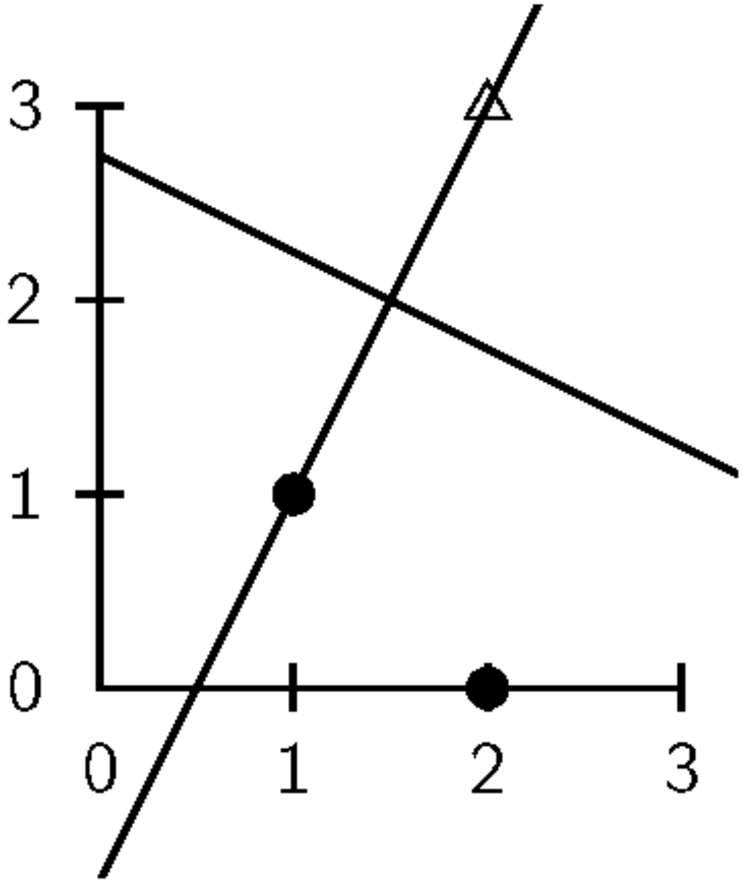
$$w_1 x_1 + w_2 x_2 + b = 0$$

Canonical form



$$.4x_1 + .8x_2 - 2.2 = 0$$

Canonical form



$$.4x_1 + .8x_2 - 2.2 = 0$$

- $.4 \cdot 1 + .8 \cdot 1 - 2.2 = -1$
- $.4 \cdot \frac{3}{2} + .8 \cdot 2 - 2.2 = 0$
- $.4 \cdot 2 + .8 \cdot 3 - 2.2 = +1$

What’s the margin?

- Distance to closest point

What’s the margin?

- Distance to closest point

$$\sqrt{\left(\frac{3}{2}-1\right)^2+(2-1)^2}=\frac{\sqrt{5}}{2}$$

What's the margin?

- Distance to closest point

$$\sqrt{\left(\frac{3}{2}-1\right)^2+(2-1)^2}=\frac{\sqrt{5}}{2}$$

- Margin computed from the weight vector

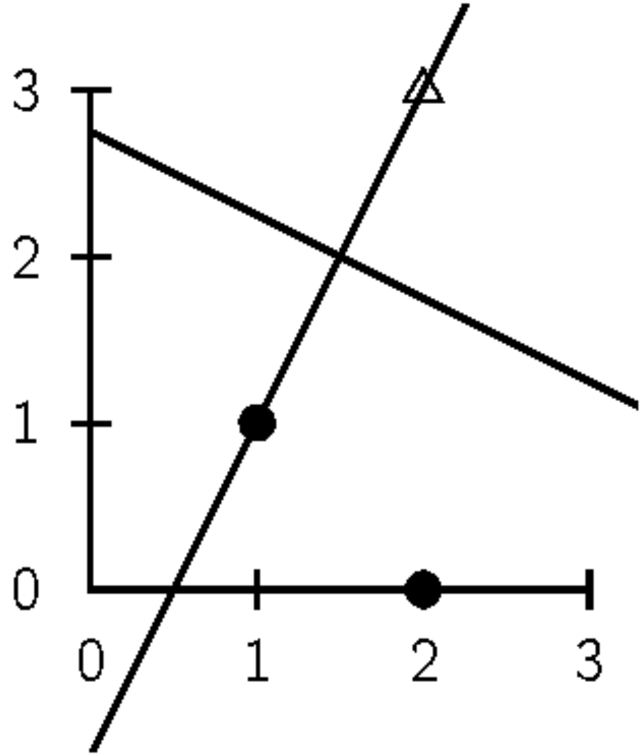
What's the margin?

- Distance to closest point

$$\sqrt{\left(\frac{3}{2}-1\right)^2+(2-1)^2}=\frac{\sqrt{5}}{2}$$

- Margin computed from the weight vector

$$\frac{1}{||w||}=\frac{1}{\sqrt{\left(\frac{2}{5}\right)^2+\left(\frac{4}{5}\right)^2}}=\frac{1}{\sqrt{\frac{20}{25}}}=\frac{5}{\sqrt{5}\sqrt{4}}=\frac{\sqrt{5}}{2}$$



$$\frac{2}{5}x+\frac{4}{5}y-\frac{11}{5}=0$$

Theoretical evidence that suggests SVMs will Work

- Leave-one-out error

Leave One Out Error (sketch)

Leave one out error is the error by using one point as your test set (averaged over all such points).

$$\hat{R}_{LOO} = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1} \left[h_{s-\{x_i\}} \neq y_i \right] \quad (1)$$

Leave One Out Error (sketch)

Leave one out error is the error by using one point as your test set (averaged over all such points).

$$\hat{R}_{LOO} = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\left[h_{s-\{x_i\}} \neq y_i\right] \tag{1}$$

This serves as an unbiased estimate of generalization error for samples of size $m-1$:

Leave One Out Error (sketch)

Leave-one-out error is bounded by the number of support vectors.

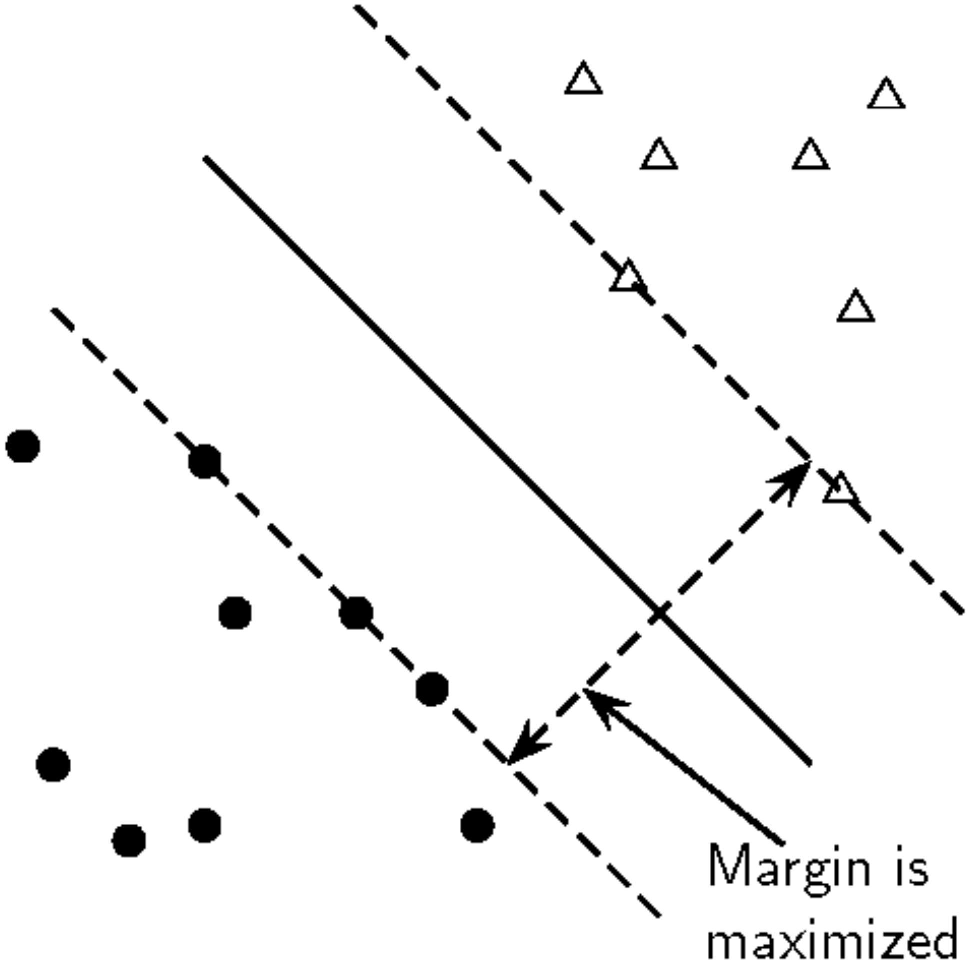
$$\mathbb{E}_{S\sim D^{m-1}}\left[R(h_s)\right] \leq \mathbb{E}_{S\sim D^{m}}\left[\frac{N_{SV}(S)}{m}\right] \tag{2}$$

Consider the held out error for x_i.

- If x_i was not a support vector, the answer doesn't change.
- If x_i was a support vector, it could change the answer; this is when we can have an error.

There are N_{SV} support vectors and thus N_{SV} possible errors.

Pictorial proof



Outline

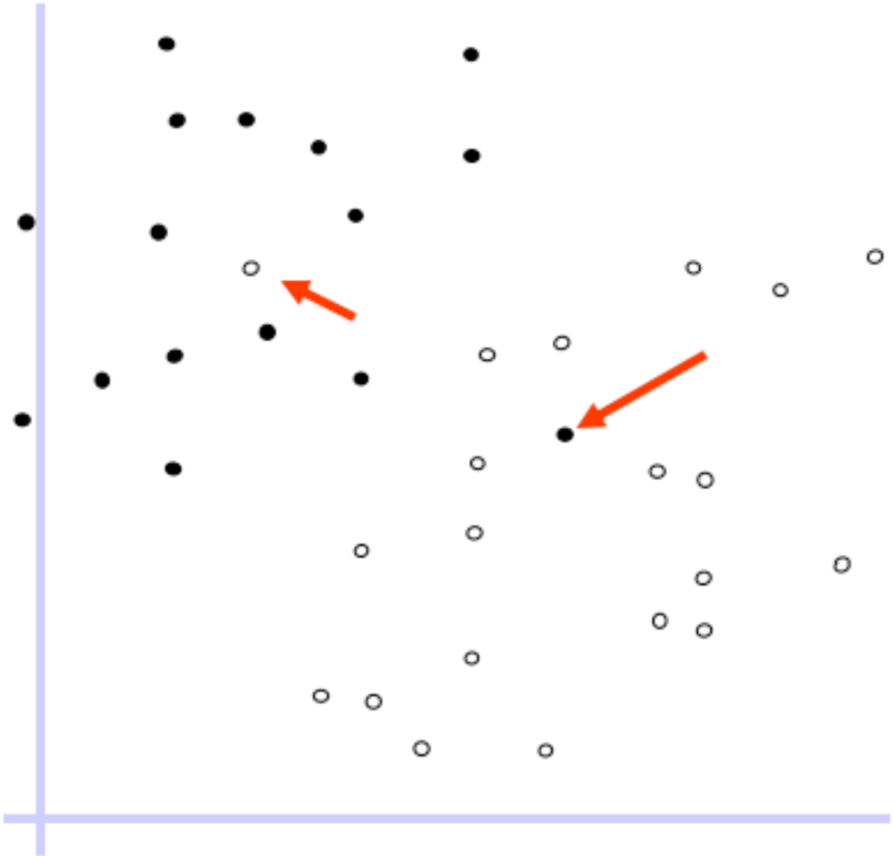
- A little bit of history
- Linear classifiers and margin
- Hard-margin SVM
- Soft-margin SVM

Objective function for hard-margin SVM

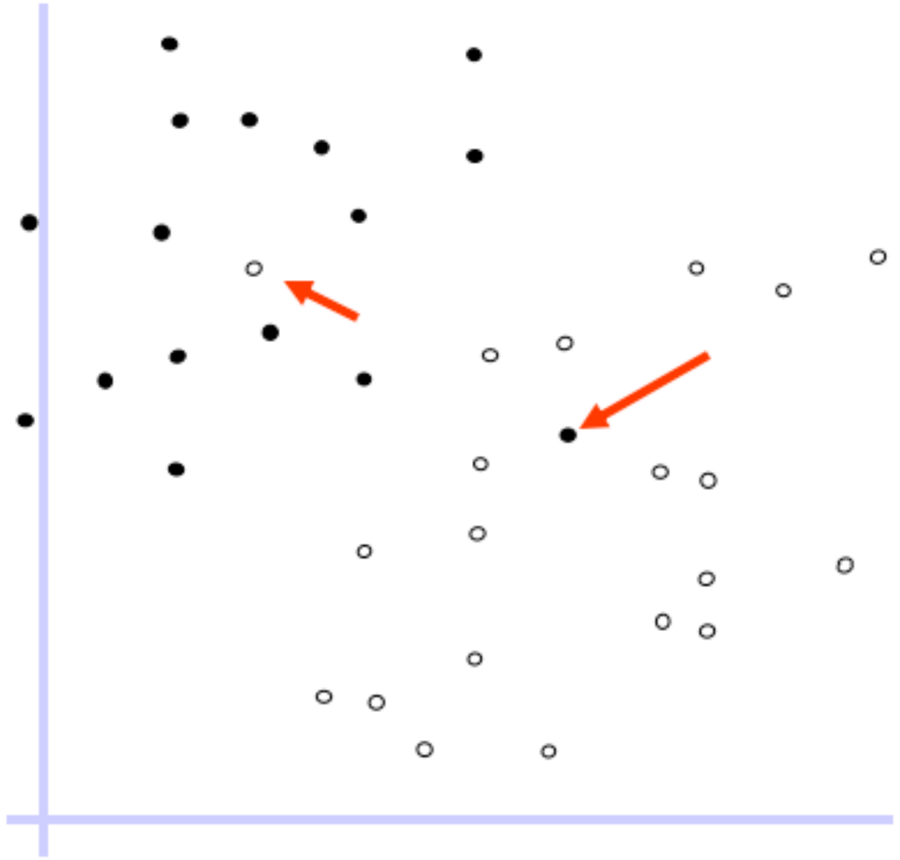
$$\min_{\boldsymbol{w}, b} \frac{1}{2} ||\boldsymbol{w}||^2$$

subject to $y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1, i \in [1, m]$

Can SVMs Work Here?

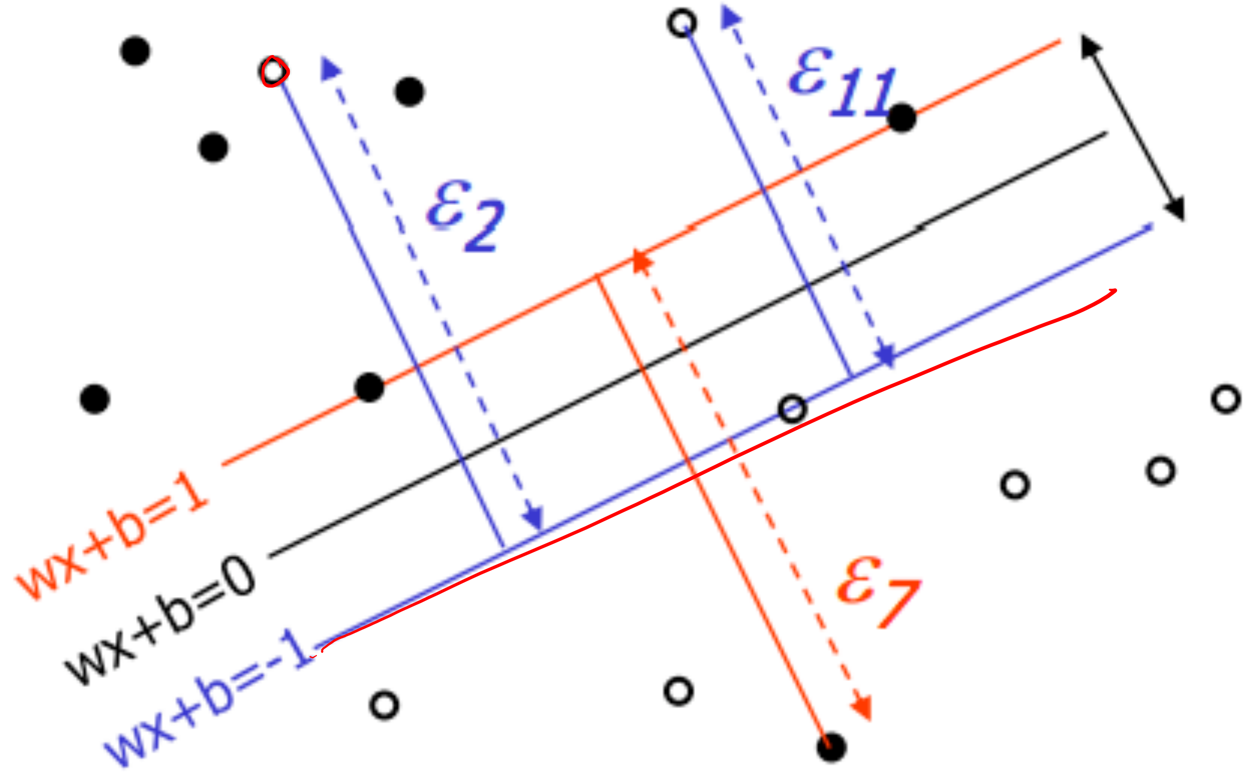


Can SVMs Work Here?



$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1$$

Trick: Allow for a few bad apples



Hard-margin objective function

$$\min_{\boldsymbol{w},b} \frac{1}{2}||\boldsymbol{w}||^2 \quad + C \sum_i \xi_i$$

subject to $y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1, i \in [1, m]$

$- \xi_i$

$\xi_i \geq 0$

Relaxing the constraint

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1 - \xi_i$$

- $\xi_i = 0$ means at least one margin on correct side of decision boundary
- $\xi_i = 1/2$ means at least one-half margin on correct side of decision boundary
- $\xi_i = 2$ means at least one margin on wrong side of decision boundary

New objective function

$$\min_{\boldsymbol{w},b,\xi} \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_i \xi_i$$

subject to

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1 - \xi_i,\ i \in [1, m]$$

$$\xi_i \geq 0,\ i \in [1, m]$$

New objective function

$$\min_{\boldsymbol{w},b,\xi} \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_i \xi_i$$

subject to

$$y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i + b) \geq 1 - \xi_i,\ i \in [1,m]$$
$$\xi_i \geq 0,\ i \in [1,m]$$

- Standard margin

New objective function

$$\min_{\boldsymbol{w},b,\xi} \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_i \xi_i$$

subject to

$$y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i + b) \geq 1 - \xi_i,\ i \in [1,m]$$
$$\xi_i \geq 0,\ i \in [1,m]$$

- Standard margin
- How wrong a point is (slack variables)

New objective function

$$\min_{\boldsymbol{w},b,\xi} \frac{1}{2}||\boldsymbol{w}||^2 + C \sum_i \xi_i$$

subject to

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1 - \xi_i, i \in [1, m]$$
$$\xi_i \geq 0, i \in [1, m]$$

- Standard margin
- How wrong a point is (slack variables)
- Tradeoff between margin and slack variables

$\min \frac{1}{2C} ||W||^2 + \sum_i \max(0, 1 - y_i(wx_i + b))$

What is the role of C?

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_i \xi_i$$

$\frac{1}{2C} ||W||^2 + \sum_i \xi_i$

subject to

$$\begin{cases} y_i(w \cdot x_i + b) \geq 1 - \xi_i, i \in [1, m] \\ \xi_i \geq 0, i \in [1, m] \end{cases}$$

$\xi_i = \begin{cases} 1 - y_i(w \cdot x_i + b) & \text{otherwise} \\ 0 & y_i(w \cdot x_i + b) \geq 1 \end{cases}$

$\xi_i = \max(0, 1 - y_i(wx_i + b))$

A. $C \uparrow \Rightarrow$ decrease bias, decrease variance

B. $C \uparrow \Rightarrow$ decrease bias, increase variance

C. $C \uparrow \Rightarrow$ increase bias, decrease variance

D. $C \uparrow \Rightarrow$ increase bias, increase variance