Introduction to Information Retrieval

http://informationretrieval.org

IIR 15-1: Support Vector Machines

Hinrich Schütze

Center for Information and Language Processing, University of Munich

2014-06-04

Overview

Outline

Rocchio, a simple vector space classifier

TrainRocchio($\mathbb{C}, \mathbb{D}$)

1 **for each** $c_j \in \mathbb{C}$

2 **do** $D_j \leftarrow \{d : \langle d, c_j \rangle \in \mathbb{D}\}$

3 $\quad \vec{\mu}_j \leftarrow \frac{1}{|D_j|} \sum_{d \in D_j} \vec{v}(d)$

4 **return** $\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}$

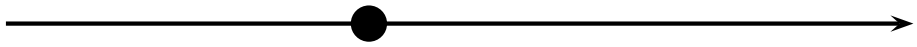
ApplyRocchio($\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}, d$)

1 **return** $\arg\min_j |\vec{\mu}_j - \vec{v}(d)|$

A linear classifier in 1D

- A linear classifier in 1D is a point described by the equation $w_1 d_1 = \theta$
- The point at θ / w_1
- Points (d_1) with $w_1 d_1 \geq \theta$ are in the class c.
- Points (d_1) with $w_1 d_1 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 1D

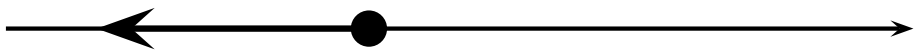


- A linear classifier in 1D is a point described by the equation $w_1 d_1 = \theta$
- The point at θ / w_1
- Points (d_1) with $w_1 d_1 \geq \theta$ are in the class c.
- Points (d_1) with $w_1 d_1 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 1D

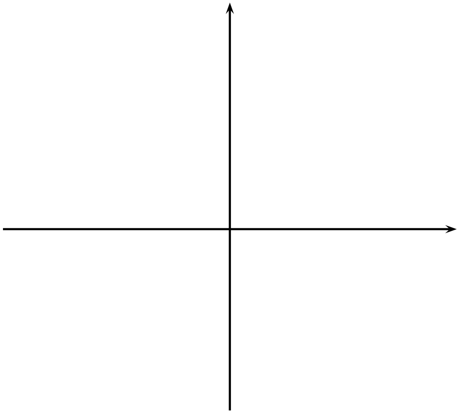
- A linear classifier in 1D is a point described by the equation $w_1 d_1 = \theta$
- The point at θ / w_1
- Points (d_1) with $w_1 d_1 \geq \theta$ are in the class c.
- Points (d_1) with $w_1 d_1 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 1D



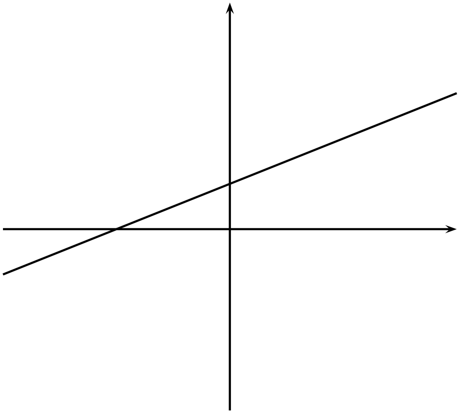
- A linear classifier in 1D is a point described by the equation $w_1 d_1 = \theta$
- The point at θ / w_1
- Points (d_1) with $w_1 d_1 \geq \theta$ are in the class c.
- Points (d_1) with $w_1 d_1 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 2D



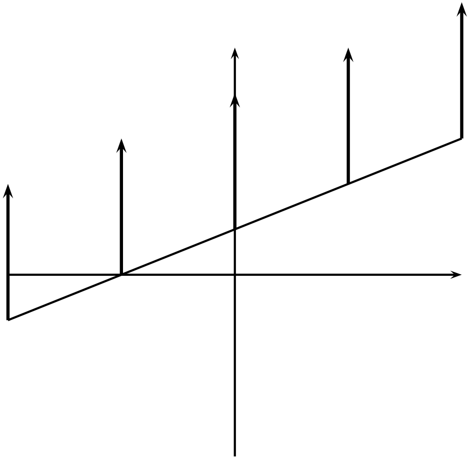
- A linear classifier in 2D is a line described by the equation $w_1 d_1 + w_2 d_2 = \theta$
- Example for a 2D linear classifier
- Points $(d_1 \;\; d_2)$ with $w_1 d_1 + w_2 d_2 \geq \theta$ are in the class c.
- Points $(d_1 \;\; d_2)$ with $w_1 d_1 + w_2 d_2 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 2D



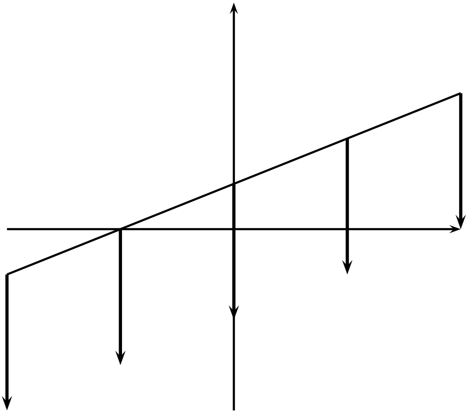
- A linear classifier in 2D is a line described by the equation $w_1 d_1 + w_2 d_2 = \theta$
- Example for a 2D linear classifier
- Points $(d_1 \; d_2)$ with $w_1 d_1 + w_2 d_2 \geq \theta$ are in the class c.
- Points $(d_1 \; d_2)$ with $w_1 d_1 + w_2 d_2 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 2D



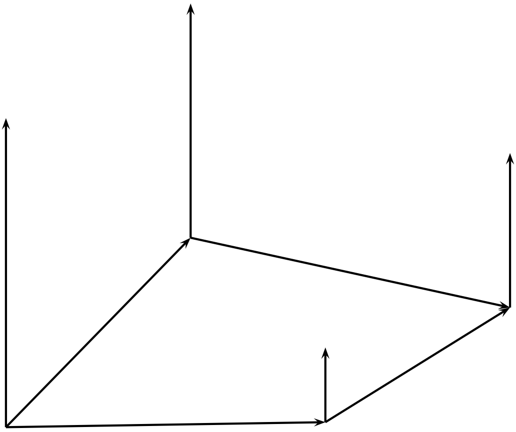
- A linear classifier in 2D is a line described by the equation $w_1 d_1 + w_2 d_2 = \theta$
- Example for a 2D linear classifier
- Points $(d_1 \ d_2)$ with $w_1 d_1 + w_2 d_2 \geq \theta$ are in the class c.
- Points $(d_1 \ d_2)$ with $w_1 d_1 + w_2 d_2 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 2D



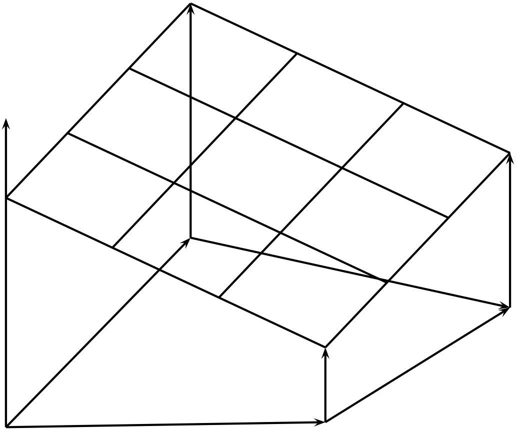
- A linear classifier in 2D is a line described by the equation $w_1 d_1 + w_2 d_2 = \theta$
- Example for a 2D linear classifier
- Points $(d_1 \; d_2)$ with $w_1 d_1 + w_2 d_2 \geq \theta$ are in the class c.
- Points $(d_1 \; d_2)$ with $w_1 d_1 + w_2 d_2 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 3D



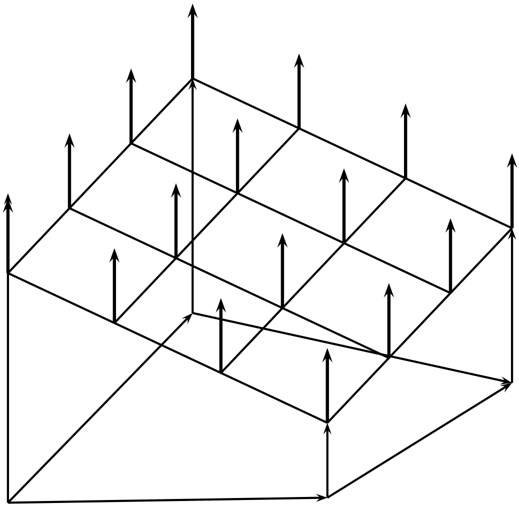
- A linear classifier in 3D is a plane described by the equation $w_1 d_1 + w_2 d_2 + w_3 d_3 = \theta$
- Example for a 3D linear classifier
- Points $(d_1 \ d_2 \ d_3)$ with $w_1 d_1 + w_2 d_2 + w_3 d_3 \geq \theta$ are in the class c.
- Points $(d_1 \ d_2 \ d_3)$ with $w_1 d_1 + w_2 d_2 + w_3 d_3 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 3D



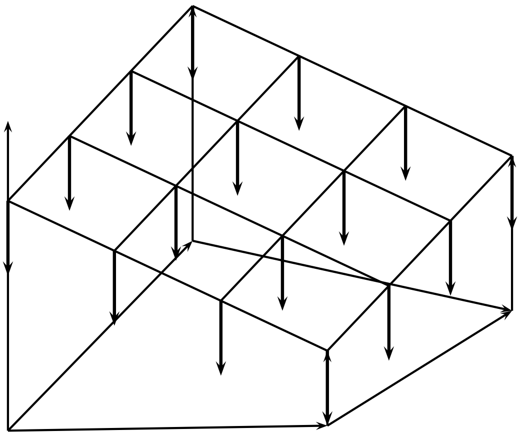
- A linear classifier in 3D is a plane described by the equation $w_1 d_1 + w_2 d_2 + w_3 d_3 = \theta$
- Example for a 3D linear classifier
- Points $(d_1\ d_2\ d_3)$ with $w_1 d_1 + w_2 d_2 + w_3 d_3 \geq \theta$ are in the class c.
- Points $(d_1\ d_2\ d_3)$ with $w_1 d_1 + w_2 d_2 + w_3 d_3 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 3D



- A linear classifier in 3D is a plane described by the equation $w_1 d_1 + w_2 d_2 + w_3 d_3 = \theta$
- Example for a 3D linear classifier
- Points $(d_1 \ d_2 \ d_3)$ with $w_1 d_1 + w_2 d_2 + w_3 d_3 \geq \theta$ are in the class c.
- Points $(d_1 \ d_2 \ d_3)$ with $w_1 d_1 + w_2 d_2 + w_3 d_3 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 3D



- A linear classifier in 3D is a plane described by the equation $w_1 d_1 + w_2 d_2 + w_3 d_3 = \theta$
- Example for a 3D linear classifier
- Points $(d_1 \; d_2 \; d_3)$ with $w_1 d_1 + w_2 d_2 + w_3 d_3 \geq \theta$ are in the class c.
- Points $(d_1 \; d_2 \; d_3)$ with $w_1 d_1 + w_2 d_2 + w_3 d_3 < \theta$ are in the complement class $\overline{c}$.

Learning algorithms for vector space classification

- In terms of actual computation, there are two types of learning algorithms.
- (i) Simple learning algorithms that estimate the parameters of the classifier directly from the training data, often in one linear pass.
  - Naive Bayes, Rocchio, kNN are all examples of this.
- (ii) Iterative algorithms
  - Support vector machines
  - Perceptron (example available as PDF on website: http://cislmu.org)
- The best performing learning algorithms usually require iterative learning.

Linear classifiers: Discussion

- Many common text classifiers are linear classifiers: Naive Bayes, Rocchio, logistic regression, linear support vector machines etc.
- Each method has a different way of selecting the separating hyperplane
  - Huge differences in performance on test documents
- Can we get better performance with more powerful nonlinear classifiers?
- Not in general: A given amount of training data may suffice for estimating a linear boundary, but not for estimating a more complex nonlinear boundary.

Take-away today

Take-away today

- Support vector machines: State-of-the-art text classification methods (linear and nonlinear)

Take-away today

- Support vector machines: State-of-the-art text classification methods (linear and nonlinear)
- Introduction to SVMs

Take-away today

- Support vector machines: State-of-the-art text classification methods (linear and nonlinear)
- Introduction to SVMs
- Formalization

Take-away today

- Support vector machines: State-of-the-art text classification methods (linear and nonlinear)
- Introduction to SVMs
- Formalization
- Soft margin case for nonseparable problems

Take-away today

- Support vector machines: State-of-the-art text classification methods (linear and nonlinear)
- Introduction to SVMs
- Formalization
- Soft margin case for nonseparable problems
- Discussion: Which classifier should I use for my problem?

Overview

Outline

Support vector machines

Support vector machines

- Machine-learning research in the last two decades has improved classifier effectiveness.

Support vector machines

- Machine-learning research in the last two decades has improved classifier effectiveness.
- New generation of state-of-the-art classifiers: support vector machines (SVMs), boosted decision trees, regularized logistic regression, maximum entropy, neural networks, and random forests

Support vector machines

- Machine-learning research in the last two decades has improved classifier effectiveness.
- New generation of state-of-the-art classifiers: support vector machines (SVMs), boosted decision trees, regularized logistic regression, maximum entropy, neural networks, and random forests
- As we saw in IIR: Applications to IR problems, particularly text classification

What is a support vector machine – first take

What is a support vector machine – first take

- Vector space classification (similar to Rocchio, kNN, linear classifiers)

What is a support vector machine – first take

- Vector space classification (similar to Rocchio, kNN, linear classifiers)
- Difference from previous methods: large margin classifier

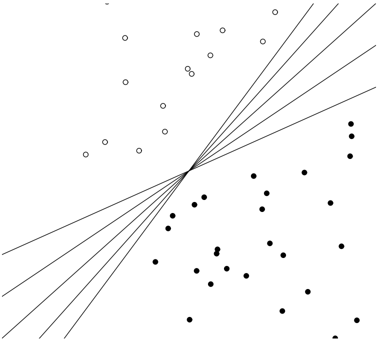
What is a support vector machine – first take

- Vector space classification (similar to Rocchio, kNN, linear classifiers)
- Difference from previous methods: large margin classifier
- We aim to find a separating hyperplane (decision boundary) that is maximally far from any point in the training data

What is a support vector machine – first take

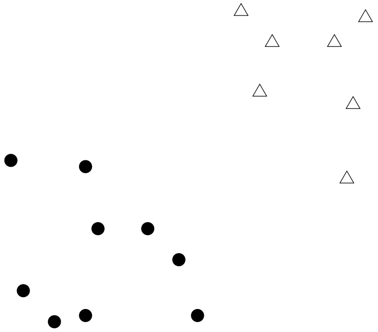
- Vector space classification (similar to Rocchio, kNN, linear classifiers)
- Difference from previous methods: large margin classifier
- We aim to find a separating hyperplane (decision boundary) that is maximally far from any point in the training data
- In case of non-linear-separability: We may have to discount some points as outliers or noise.

Which hyperplane?



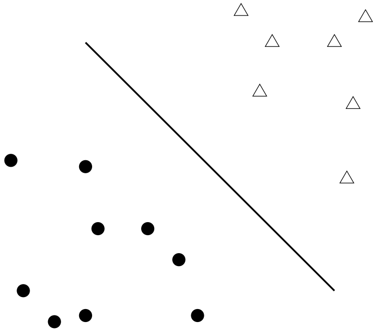
(Linear) Support Vector Machines

- binary classification problem



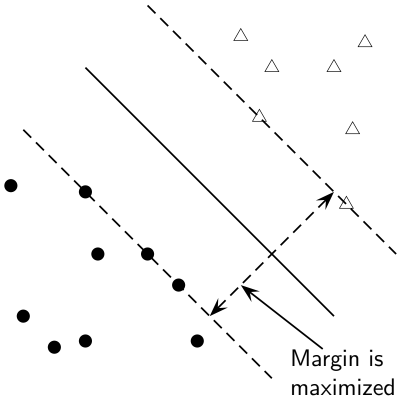
(Linear) Support Vector Machines

- binary classification problem
- Decision boundary is linear separator.



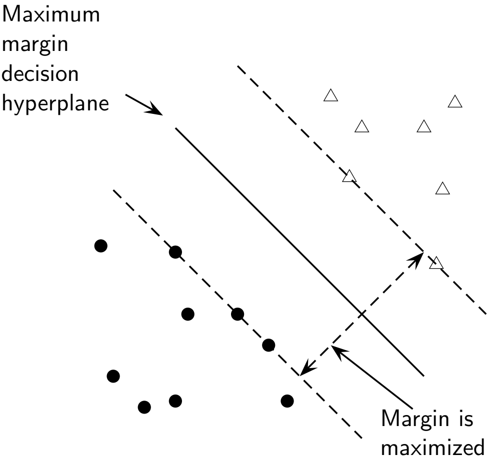
(Linear) Support Vector Machines

- binary classification problem
- Decision boundary is linear separator.
- criterion: being maximally far away from any data point $\rightarrow$ determines classifier margin



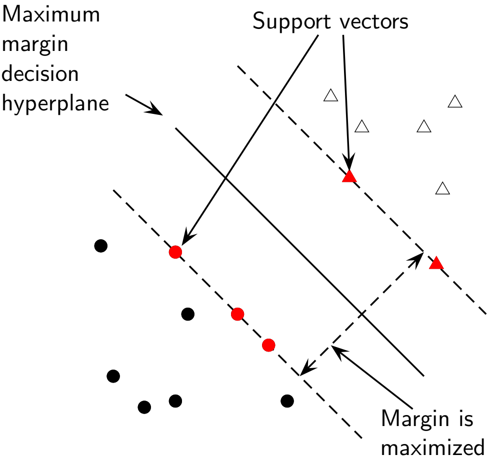
(Linear) Support Vector Machines

- binary classification problem
- Decision boundary is linear separator.
- criterion: being maximally far away from any data point → determines classifier margin



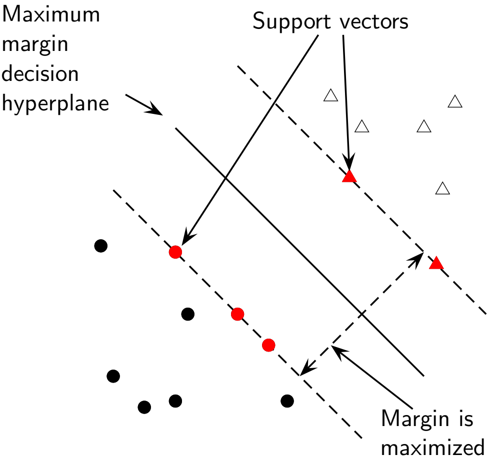
(Linear) Support Vector Machines

- binary classification problem
- Decision boundary is linear separator.
- criterion: being maximally far away from any data point → determines classifier margin
- Vectors on margin lines are called support vectors



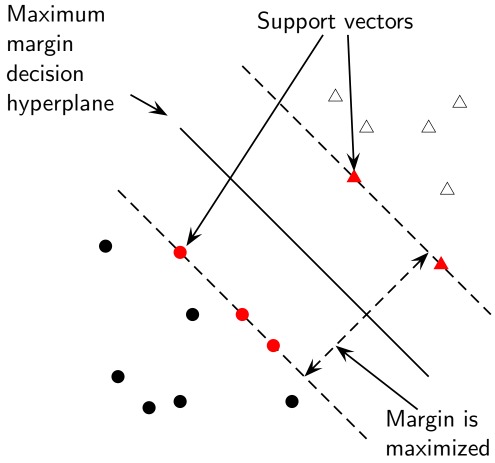
(Linear) Support Vector Machines

- binary classification problem
- Decision boundary is linear separator.
- criterion: being maximally far away from any data point → determines classifier margin
- Vectors on margin lines are called support vectors
- Set of support vectors are a complete specification of classifier



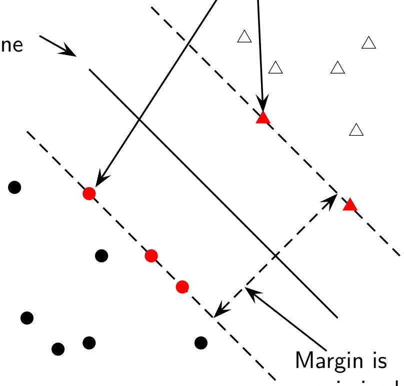
Why maximize the margin?

Points near the decision surface are uncertain classification decisions.



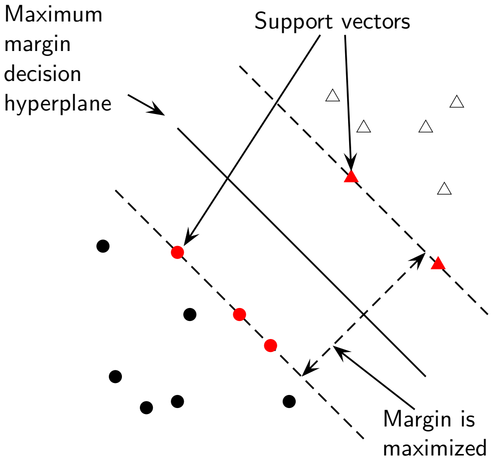
Why maximize the margin?

Points near the decision surface are uncertain classification decisions. A classifier with a large margin makes no low certainty classification decisions (on the training set).

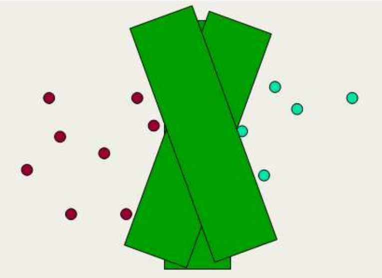


Why maximize the margin?

Points near the decision surface are uncertain classification decisions. A classifier with a large margin makes no low certainty classification decisions (on the training set). Gives classification safety margin with respect to errors and random variation

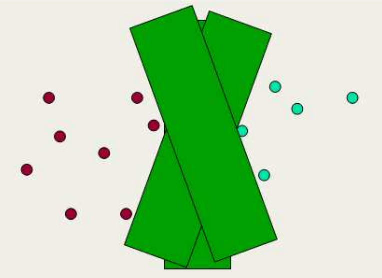


Why maximize the margin?



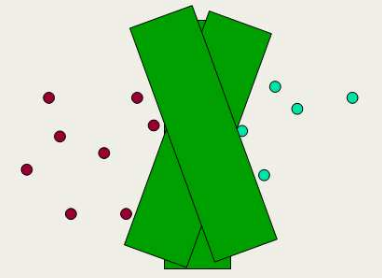
- SVM classification = large margin around decision boundary

Why maximize the margin?



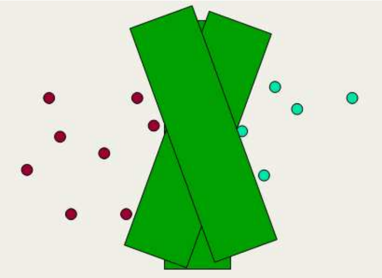
- SVM classification = large margin around decision boundary
- We can think of the margin as a "fat separator" – a fatter version of our regular decision hyperplane.

Why maximize the margin?



- SVM classification = large margin around decision boundary
- We can think of the margin as a "fat separator" – a fatter version of our regular decision hyperplane.
- unique solution

Why maximize the margin?



- SVM classification = large margin around decision boundary
- We can think of the margin as a "fat separator" – a fatter version of our regular decision hyperplane.
- unique solution
- increased ability to correctly generalize to test data

Separating hyperplane: Recap

Hyperplane

An n-dimensional generalization of a plane (point in 1-D space, line in 2-D space, ordinary plane in 3-D space).

Decision hyperplane

Can be defined by:

- intercept term b (we were calling this θ before)
- normal vector $\vec{w}$ (weight vector) which is perpendicular to the hyperplane

All points $\vec{x}$ on the hyperplane satisfy:

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

Notation: Different conventions for linear separator

Notation: Different conventions for linear separator

- $\vec{w}^{\mathsf{T}}\vec{x} + b = 0$

Notation: Different conventions for linear separator

- $\vec{w}^{\mathsf{T}}\vec{x} + b = 0$
  - Used in SVM literature

Notation: Different conventions for linear separator

- $\vec{w}^{\mathsf{T}}\vec{x} + b = 0$
  - Used in SVM literature
- $\vec{w}^{\mathsf{T}}\vec{x} = 0$

Notation: Different conventions for linear separator

- $\vec{w}^{\mathsf{T}}\vec{x} + b = 0$
  - Used in SVM literature
- $\vec{w}^{\mathsf{T}}\vec{x} = 0$
  - Often used in perceptron literature, folds threshold into vector by adding a constant dimension (set to 1 or -1 for all vectors)

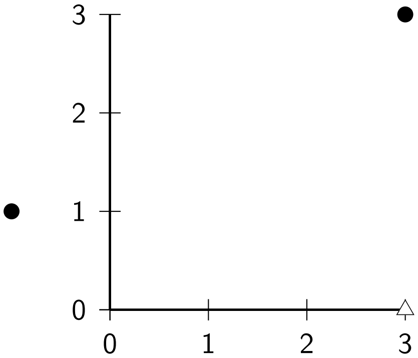
Notation: Different conventions for linear separator

- $\vec{w}^{\mathsf{T}}\vec{x} + b = 0$
  - Used in SVM literature
- $\vec{w}^{\mathsf{T}}\vec{x} = 0$
  - Often used in perceptron literature, folds threshold into vector by adding a constant dimension (set to 1 or -1 for all vectors)
- $\sum_{i=1}^{M} w_i d_i = \theta$

Notation: Different conventions for linear separator

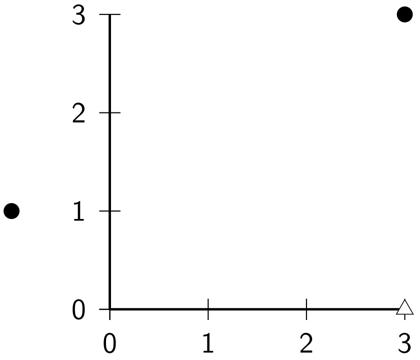
- $\vec{w}^{\mathsf{T}}\vec{x} + b = 0$
  - Used in SVM literature
- $\vec{w}^{\mathsf{T}}\vec{x} = 0$
  - Often used in perceptron literature, folds threshold into vector by adding a constant dimension (set to 1 or -1 for all vectors)
- $\sum_{i=1}^{M} w_i d_i = \theta$
  - "Spelled out" version we used in the last chapter for linear separators

Exercise



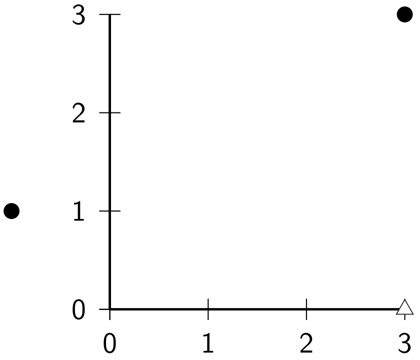
Draw the maximum margin separator. Which vectors are the support vectors? Coordinates of dots: (3,3), (-1,1). Coordinates of triangle: (3,0)

Exercise



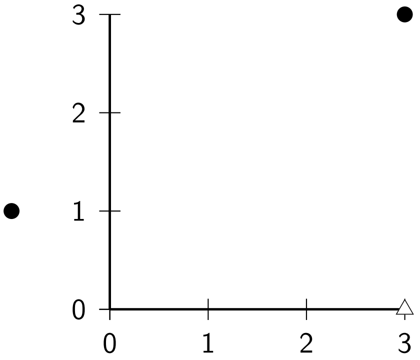
Draw the maximum margin separator. Which vectors are the support vectors? Coordinates of dots: (3,3), (-1,1). Coordinates of triangle: (3,0)

Exercise



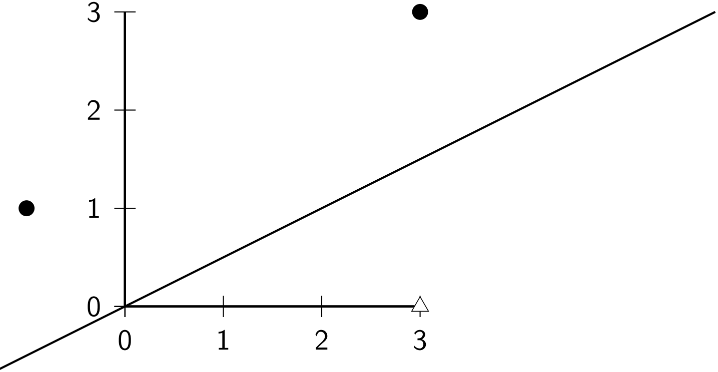
Draw the maximum margin separator. Which vectors are the support vectors? Coordinates of dots: (3,3), (-1,1). Coordinates of triangle: (3,0)

Exercise



Draw the maximum margin separator. Which vectors are the support vectors? Coordinates of dots: (3,3), (-1,1). Coordinates of triangle: (3,0)

Exercise



Draw the maximum margin separator. Which vectors are the support vectors? Coordinates of dots: (3,3), (-1,1). Coordinates of triangle: (3,0)

Outline

Formalization of SVMs

Training set

Consider a binary classification problem:

- $\vec{x}_i$ are the input vectors
- y_i are the labels

Formalization of SVMs

Training set

Consider a binary classification problem:

- $\vec{x}_i$ are the input vectors
- y_i are the labels

For SVMs, the two classes are $y_i = +1$ and $y_i = -1$.

Formalization of SVMs

Training set

Consider a binary classification problem:

- $\vec{x}_i$ are the input vectors
- y_i are the labels

For SVMs, the two classes are $y_i = +1$ and $y_i = -1$.

The linear classifier is then:

$$f(\vec{x}) = \text{sign}(\vec{w}^{\mathsf{T}}\vec{x} + b)$$

Formalization of SVMs

Training set

Consider a binary classification problem:

- $\vec{x}_i$ are the input vectors
- y_i are the labels

For SVMs, the two classes are $y_i = +1$ and $y_i = -1$.

The linear classifier is then:

$$f(\vec{x}) = \text{sign}(\vec{w}^{\mathsf{T}}\vec{x} + b)$$

A value of -1 indicates one class, and a value of $+1$ the other class.

Functional margin of a point

Functional margin of a point

SVM makes its decision based on the score $\vec{w}^{\mathsf{T}}\vec{x} + b$.

Functional margin of a point

SVM makes its decision based on the score $\vec{w}^{\mathsf{T}}\vec{x} + b$.
Clearly, the larger $|\vec{w}^{\mathsf{T}}\vec{x} + b|$ is, the more confidence we can have that the decision is correct.

Functional margin of a point

SVM makes its decision based on the score $\vec{w}^{\mathsf{T}}\vec{x} + b$.
Clearly, the larger $|\vec{w}^{\mathsf{T}}\vec{x} + b|$ is, the more confidence we can have that the decision is correct.

Functional margin of a point

SVM makes its decision based on the score $\vec{w}^{\mathsf{T}}\vec{x} + b$.
Clearly, the larger $|\vec{w}^{\mathsf{T}}\vec{x} + b|$ is, the more confidence we can have that the decision is correct.

Functional margin

- The functional margin of the vector $\vec{x}_i$ w.r.t the hyperplane $\langle \vec{w}, b \rangle$ is: $y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b)$

Functional margin of a point

SVM makes its decision based on the score $\vec{w}^{\mathsf{T}}\vec{x} + b$.
Clearly, the larger $|\vec{w}^{\mathsf{T}}\vec{x} + b|$ is, the more confidence we can have that the decision is correct.

Functional margin

- The functional margin of the vector $\vec{x}_i$ w.r.t the hyperplane $\langle \vec{w}, b \rangle$ is: $y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b)$
- The functional margin of a data set w.r.t a decision surface is twice the functional margin of any of the points in the data set with minimal functional margin

Functional margin of a point

SVM makes its decision based on the score $\vec{w}^{\mathsf{T}}\vec{x} + b$.
Clearly, the larger $|\vec{w}^{\mathsf{T}}\vec{x} + b|$ is, the more confidence we can have that the decision is correct.

Functional margin

- The functional margin of the vector $\vec{x}_i$ w.r.t the hyperplane $\langle \vec{w}, b \rangle$ is: $y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b)$
- The functional margin of a data set w.r.t a decision surface is twice the functional margin of any of the points in the data set with minimal functional margin
- Factor 2 comes from measuring across the whole width of the margin.

Functional margin of a point

SVM makes its decision based on the score $\vec{w}^{\mathsf{T}}\vec{x} + b$.
Clearly, the larger $|\vec{w}^{\mathsf{T}}\vec{x} + b|$ is, the more confidence we can have that the decision is correct.

Functional margin

- The functional margin of the vector $\vec{x}_i$ w.r.t the hyperplane $\langle \vec{w}, b \rangle$ is: $y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b)$
- The functional margin of a data set w.r.t a decision surface is twice the functional margin of any of the points in the data set with minimal functional margin
- Factor 2 comes from measuring across the whole width of the margin.

Problem: We can increase functional margin by scaling $\vec{w}$ and b.

Functional margin of a point

SVM makes its decision based on the score $\vec{w}^T\vec{x} + b$.
Clearly, the larger $|\vec{w}^T\vec{x} + b|$ is, the more confidence we can have that the decision is correct.

Functional margin

- The functional margin of the vector $\vec{x}_i$ w.r.t the hyperplane $\langle \vec{w}, b \rangle$ is: $y_i(\vec{w}^T\vec{x}_i + b)$
- The functional margin of a data set w.r.t a decision surface is twice the functional margin of any of the points in the data set with minimal functional margin
- Factor 2 comes from measuring across the whole width of the margin.

Problem: We can increase functional margin by scaling $\vec{w}$ and b.
$\rightarrow$ We need to place some constraint on the size of $\vec{w}$.

Geometric margin

Geometric margin

Geometric margin of the classifier: maximum width of the band that can be drawn separating the support vectors of the two classes.

Geometric margin

Geometric margin of the classifier: maximum width of the band that can be drawn separating the support vectors of the two classes.

To compute the geometric margin, we need to compute the distance of a vector $\vec{x}$ from the hyperplane:

Geometric margin

Geometric margin of the classifier: maximum width of the band that can be drawn separating the support vectors of the two classes.

To compute the geometric margin, we need to compute the distance of a vector $\vec{x}$ from the hyperplane:

$$r = y\frac{\vec{w}^{\mathsf{T}}\vec{x} + b}{|\vec{w}|}$$

Geometric margin

Geometric margin of the classifier: maximum width of the band that can be drawn separating the support vectors of the two classes.

To compute the geometric margin, we need to compute the distance of a vector $\vec{x}$ from the hyperplane:

$$r = y\frac{\vec{w}^{\mathsf{T}}\vec{x} + b}{|\vec{w}|}$$

(why? we will see that this is so graphically in a few moments)

Geometric margin

Geometric margin of the classifier: maximum width of the band that can be drawn separating the support vectors of the two classes.

To compute the geometric margin, we need to compute the distance of a vector $\vec{x}$ from the hyperplane:

$$r = y\frac{\vec{w}^{\mathsf{T}}\vec{x} + b}{|\vec{w}|}$$

(why? we will see that this is so graphically in a few moments)

Distance is of course invariant to scaling:

Geometric margin

Geometric margin of the classifier: maximum width of the band that can be drawn separating the support vectors of the two classes.

To compute the geometric margin, we need to compute the distance of a vector $\vec{x}$ from the hyperplane:

$$r = y\frac{\vec{w}^{\mathsf{T}}\vec{x} + b}{|\vec{w}|}$$

(why? we will see that this is so graphically in a few moments)

Distance is of course invariant to scaling: if we replace $\vec{w}$ by $5\vec{w}$ and b by $5b$, then the distance is the same because it is normalized by the length of $\vec{w}$.

Optimization problem solved by SVMs

Optimization problem solved by SVMs

Assume canonical “functional margin” distance

Optimization problem solved by SVMs

Assume canonical "functional margin" distance

Assume that every data point has at least distance 1 from the hyperplane, then:

Optimization problem solved by SVMs

Assume canonical "functional margin" distance

Assume that every data point has at least distance 1 from the hyperplane, then:

$$y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b) \geq 1$$

Optimization problem solved by SVMs

Assume canonical "functional margin" distance

Assume that every data point has at least distance 1 from the hyperplane, then:

$$y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b) \geq 1$$

Since each example's distance from the hyperplane is $r_i = y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b)/|\vec{w}|$, the margin is $\rho = 2/|\vec{w}|$.

Optimization problem solved by SVMs

Assume canonical "functional margin" distance

Assume that every data point has at least distance 1 from the hyperplane, then:

$$y_i(\vec{w}^\mathsf{T}\vec{x}_i + b) \geq 1$$

Since each example's distance from the hyperplane is $r_i = y_i(\vec{w}^\mathsf{T}\vec{x}_i + b)/|\vec{w}|$, the margin is $\rho = 2/|\vec{w}|$.

We want to maximize this margin.

Optimization problem solved by SVMs

Assume canonical "functional margin" distance

Assume that every data point has at least distance 1 from the hyperplane, then:

$$y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b) \geq 1$$

Since each example's distance from the hyperplane is $r_i = y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b)/|\vec{w}|$, the margin is $\rho = 2/|\vec{w}|$.

We want to maximize this margin.

That is, we want to find $\vec{w}$ and b such that:

Optimization problem solved by SVMs

Assume canonical "functional margin" distance

Assume that every data point has at least distance 1 from the hyperplane, then:

$$y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b) \geq 1$$

Since each example's distance from the hyperplane is $r_i = y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b)/|\vec{w}|$, the margin is $\rho = 2/|\vec{w}|$.

We want to maximize this margin.

That is, we want to find $\vec{w}$ and b such that:

- For all $(\vec{x}_i, y_i) \in \mathbb{D}$, $y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b) \geq 1$

Optimization problem solved by SVMs

Assume canonical "functional margin" distance

Assume that every data point has at least distance 1 from the hyperplane, then:

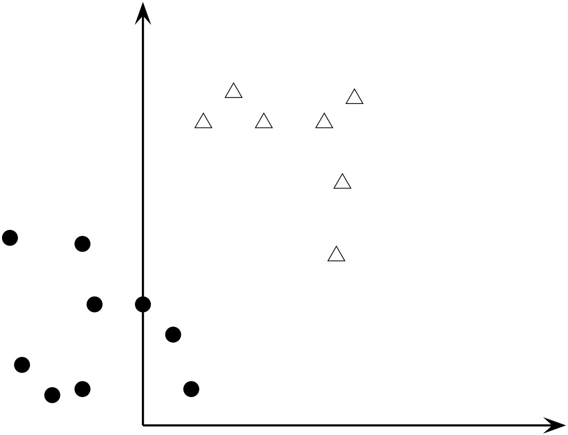
$$y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b) \geq 1$$

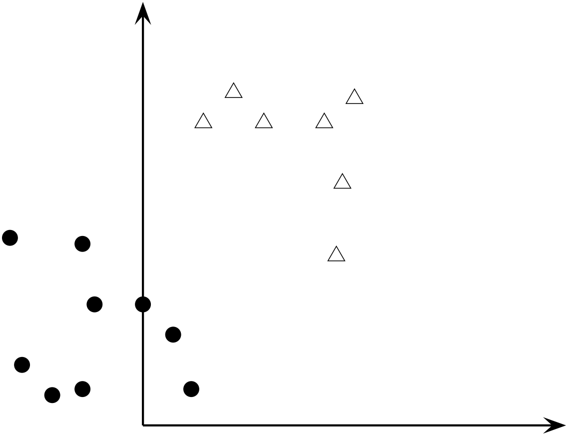
Since each example's distance from the hyperplane is $r_i = y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b)/|\vec{w}|$, the margin is $\rho = 2/|\vec{w}|$.

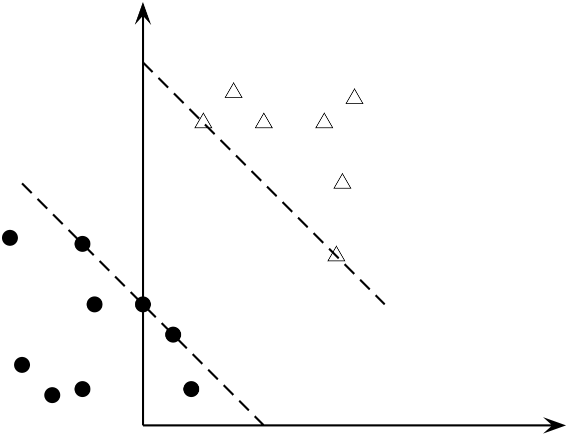
We want to maximize this margin.

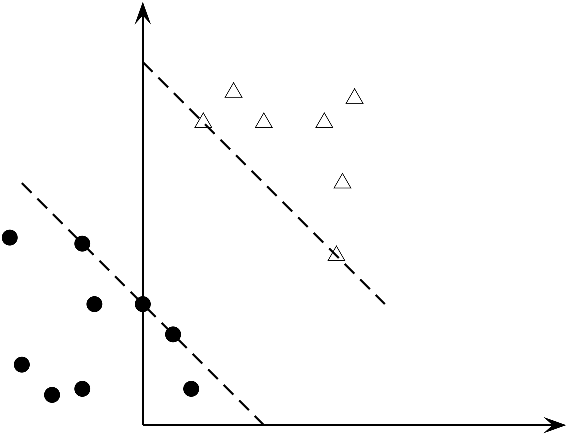
That is, we want to find $\vec{w}$ and b such that:

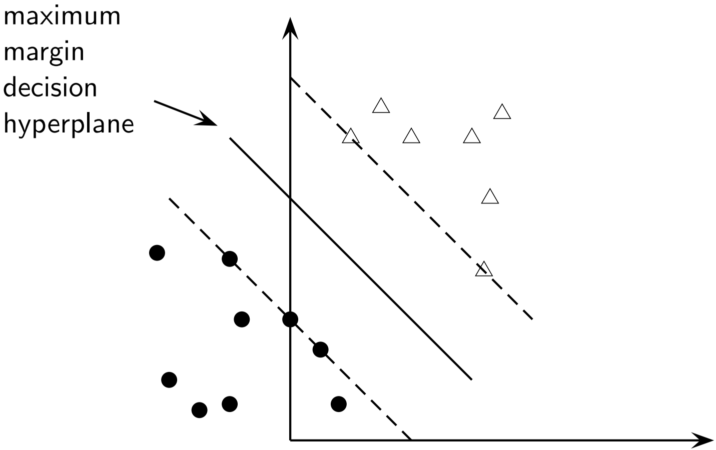
- For all $(\vec{x}_i, y_i) \in \mathbb{D}$, $y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b) \geq 1$
- $\rho = 2/|\vec{w}|$ is maximized



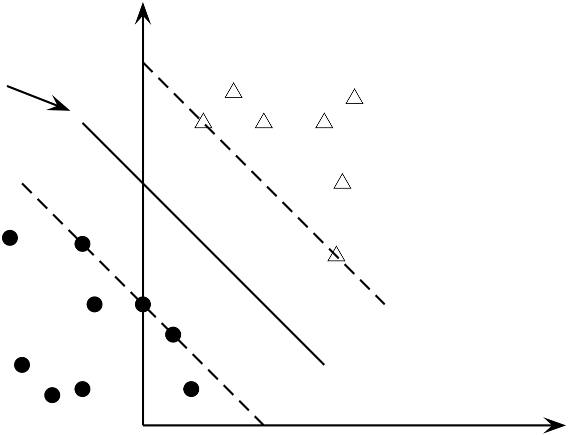






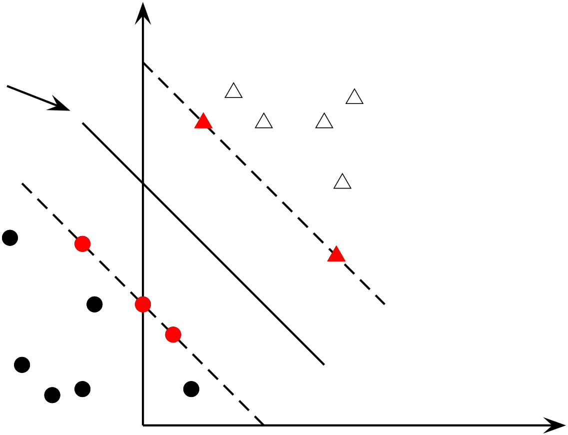

maximum
margin
decision
hyperplane

maximum margin decision hyperplane

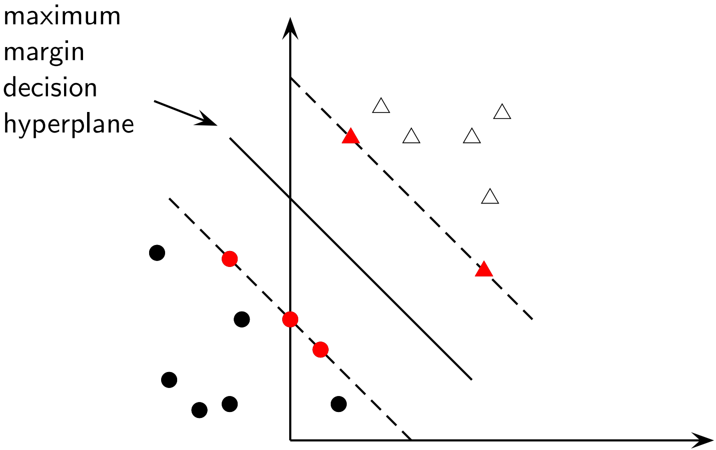


support vectors in red

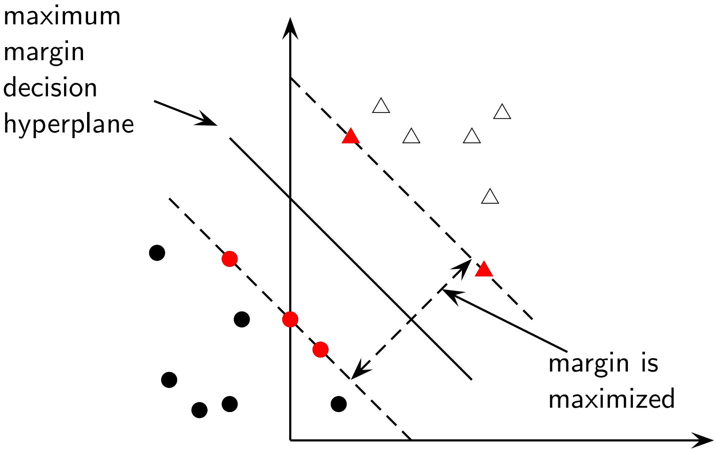
maximum margin decision hyperplane



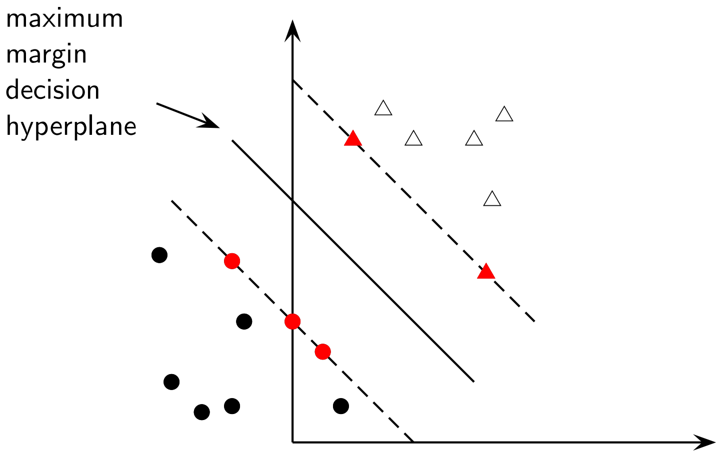
support vectors in red



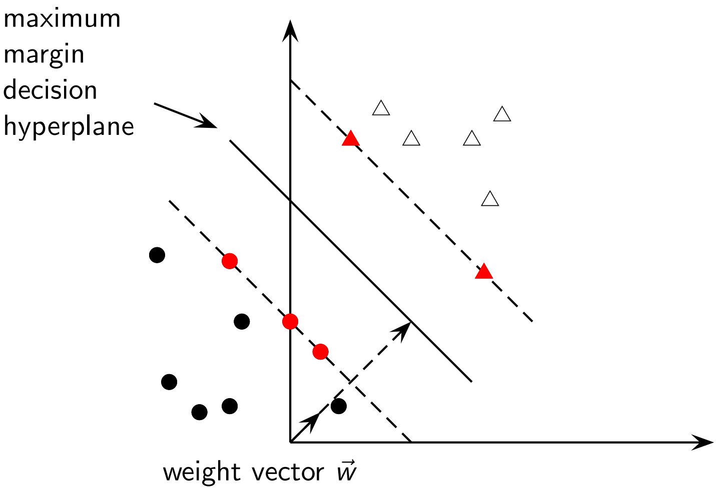
support vectors in red



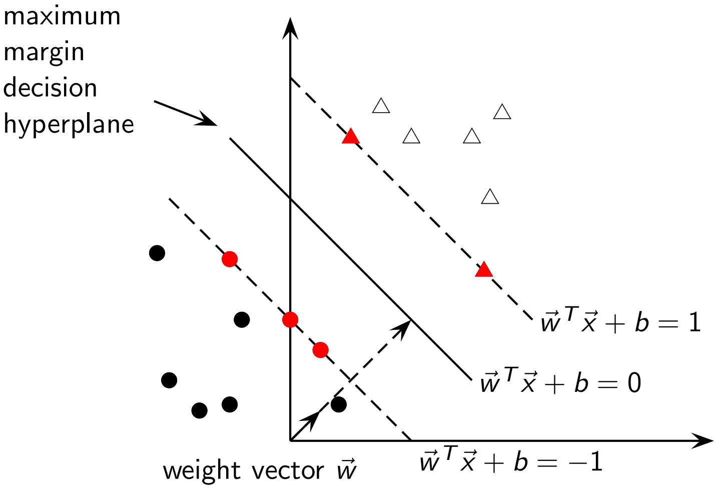
support vectors in red

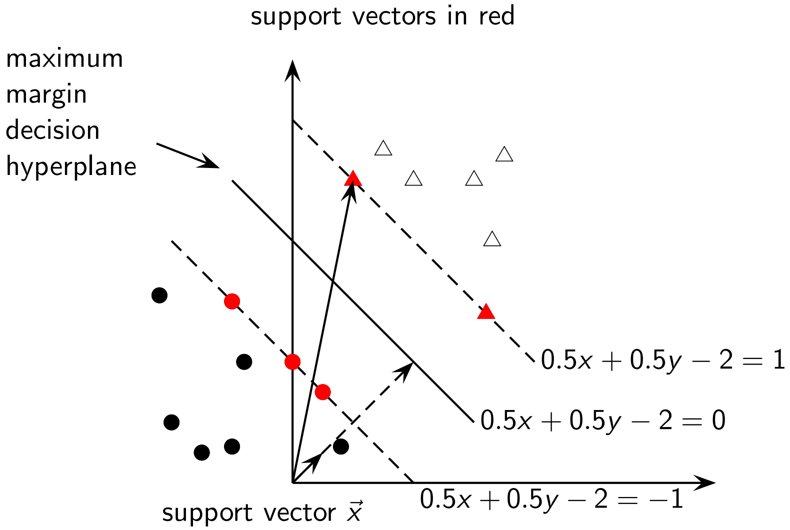


support vectors in red



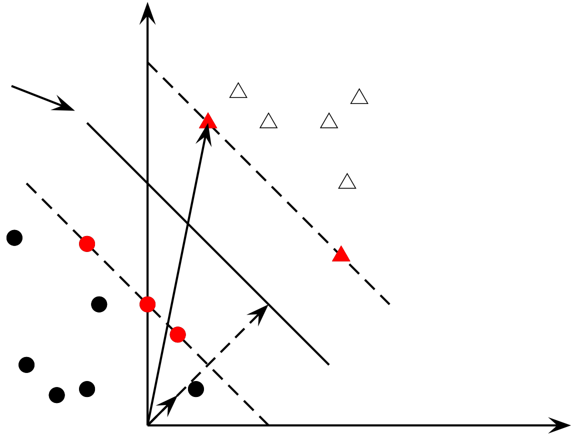
support vectors in red

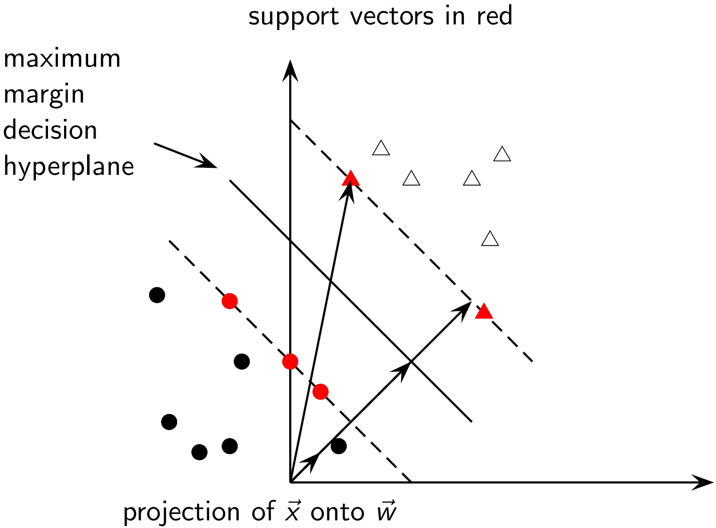



support vectors in red
maximum margin decision hyperplane
$0.5x + 0.5y - 2 = 1$
$0.5x + 0.5y - 2 = 0$
$0.5x + 0.5y - 2 = -1$
support vector $\vec{x}$

support vectors in red

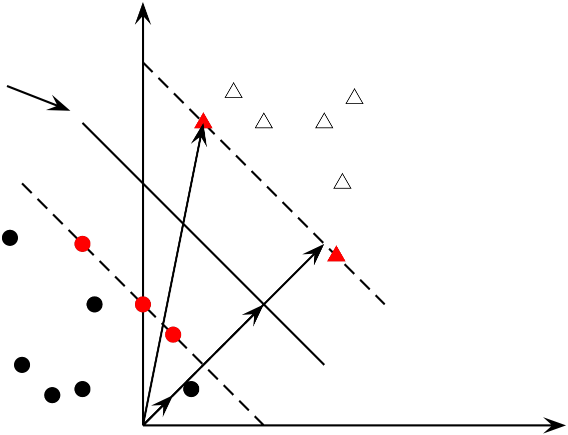
maximum margin decision hyperplane



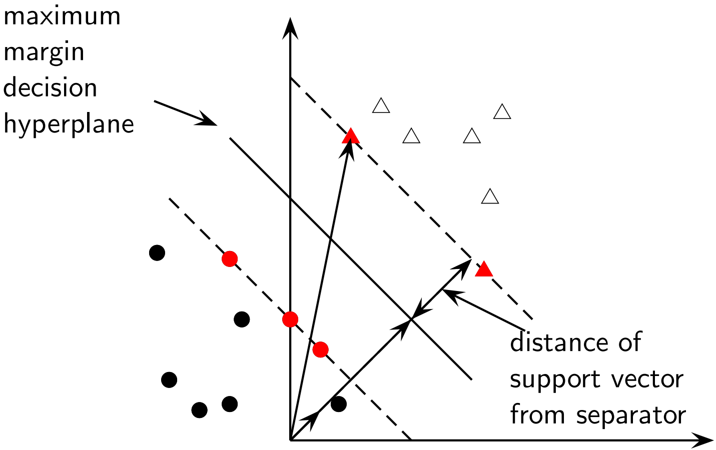

support vectors in red
maximum
margin
decision
hyperplane
projection of $\vec{x}$ onto $\vec{w}$

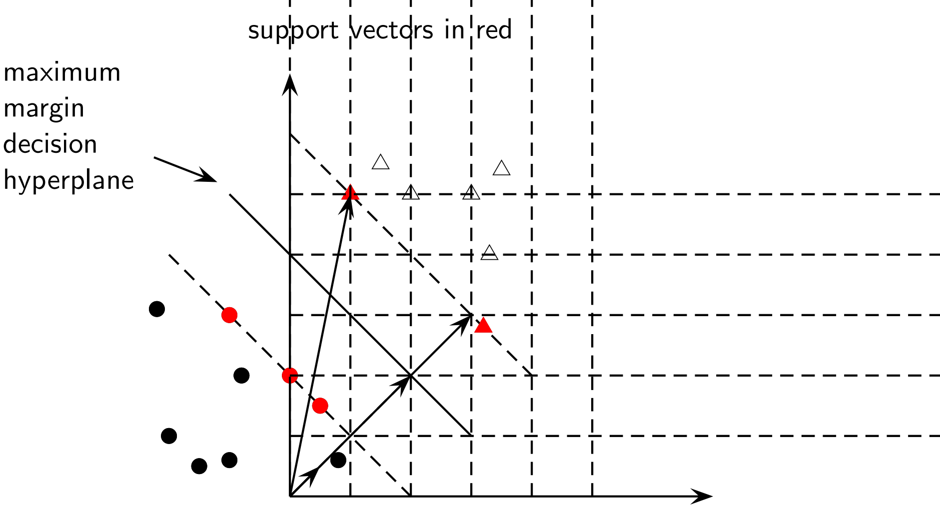
support vectors in red

maximum margin decision hyperplane



support vectors in red




support vectors in red
maximum margin decision hyperplane

$$\vec{w}^{\mathsf{T}}\vec{w}' + b = 0$$

$$b = -\vec{w}^{\mathsf{T}}\vec{w}'$$

$$\frac{b}{|\vec{w}|} = -\frac{\vec{w}^{\mathsf{T}}\vec{w}'}{|\vec{w}|}$$

Distance of support vector from separator =
(length of projection of $\vec{x}$ onto $\vec{w}$) minus (length of $\vec{w}'$)

$$\frac{\vec{w}^{\mathsf{T}}\vec{x}}{|\vec{w}|} - \frac{\vec{w}^{\mathsf{T}}\vec{w}'}{|\vec{w}|}$$

$$= \frac{\vec{w}^{\mathsf{T}}\vec{x}}{|\vec{w}|} + \frac{b}{|\vec{w}|}$$

$$= \frac{\vec{w}^{\mathsf{T}}\vec{x} + b}{|\vec{w}|}$$

Distance of support vector from separator =
(length of projection of $\vec{x} = (1\ 5)^\mathsf{T}$ onto $\vec{w}$) minus (length of $\vec{w}'$)

$$\frac{\vec{w}^\mathsf{T}\vec{x}}{|\vec{w}|} - \frac{\vec{w}^\mathsf{T}\vec{w}'}{|\vec{w}|}$$

$$(0.5 \cdot 1 + 0.5 \cdot 5)/(1/\sqrt{2}) - (0.5 \cdot 2 + 0.5 \cdot 2)/(1/\sqrt{2})$$
$$3/(1/\sqrt{2}) - 2/(1/\sqrt{2})$$

$$\frac{\vec{w}^\mathsf{T}\vec{x}}{|\vec{w}|} + \frac{b}{|\vec{w}|}$$

$$3/(1/\sqrt{2}) + (-2)/(1/\sqrt{2})$$

$$\frac{3-2}{1/\sqrt{2}}$$

$$\sqrt{2}$$

Optimization problem solved by SVMs (2)

Optimization problem solved by SVMs (2)

Maximizing $2/|\vec{w}|$ is the same as minimizing $|\vec{w}|/2$.

Optimization problem solved by SVMs (2)

Maximizing $2/|\vec{w}|$ is the same as minimizing $|\vec{w}|/2$.
This gives the final standard formulation of an SVM as a minimization problem:

Optimization problem solved by SVMs (2)

Maximizing $2/|\vec{w}|$ is the same as minimizing $|\vec{w}|/2$.
This gives the final standard formulation of an SVM as a minimization problem:

Optimization problem solved by SVMs

Find $\vec{w}$ and b such that:

Optimization problem solved by SVMs (2)

Maximizing $2/|\vec{w}|$ is the same as minimizing $|\vec{w}|/2$.
This gives the final standard formulation of an SVM as a minimization problem:

Optimization problem solved by SVMs

Find $\vec{w}$ and b such that:

- $\frac{1}{2}\vec{w}^T\vec{w}$ is minimized (because $|\vec{w}| = \sqrt{\vec{w}^T\vec{w}}$), and

Optimization problem solved by SVMs (2)

Maximizing $2/|\vec{w}|$ is the same as minimizing $|\vec{w}|/2$.
This gives the final standard formulation of an SVM as a minimization problem:

Optimization problem solved by SVMs

Find $\vec{w}$ and b such that:

- $\frac{1}{2}\vec{w}^{\mathsf{T}}\vec{w}$ is minimized (because $|\vec{w}| = \sqrt{\vec{w}^{\mathrm{T}}\vec{w}}$), and
- for all $\{(\vec{x}_i, y_i)\}$, $y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b) \geq 1$

Optimization problem solved by SVMs (2)

Maximizing $2/|\vec{w}|$ is the same as minimizing $|\vec{w}|/2$.
This gives the final standard formulation of an SVM as a minimization problem:

Optimization problem solved by SVMs

Find $\vec{w}$ and b such that:

- $\frac{1}{2}\vec{w}^{\mathsf{T}}\vec{w}$ is minimized (because $|\vec{w}| = \sqrt{\vec{w}^{\mathrm{T}}\vec{w}}$), and
- for all $\{(\vec{x}_i, y_i)\}$, $y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b) \geq 1$

We are now optimizing a quadratic function subject to linear constraints. Quadratic optimization problems are standard mathematical optimization problems, and many algorithms exist for solving them (e.g. Quadratic Programming libraries).

Recap

Recap

- We start with a training set.

Recap

- We start with a training set.
- The data set defines the maximum-margin separating hyperplane (if it is separable).

Recap

- We start with a training set.
- The data set defines the maximum-margin separating hyperplane (if it is separable).
- We use quadratic optimization to find this plane.

Recap

- We start with a training set.
- The data set defines the maximum-margin separating hyperplane (if it is separable).
- We use quadratic optimization to find this plane.
- Given a new point $\vec{x}$ to classify, the classification function $f(\vec{x})$ computes the functional margin of the point (= normalized distance).

Recap

- We start with a training set.
- The data set defines the maximum-margin separating hyperplane (if it is separable).
- We use quadratic optimization to find this plane.
- Given a new point $\vec{x}$ to classify, the classification function $f(\vec{x})$ computes the functional margin of the point ($=$ normalized distance).
- The sign of this function determines the class to assign to the point.

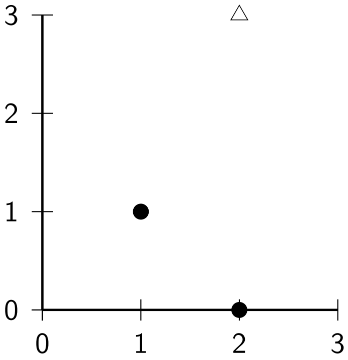
Recap

- We start with a training set.
- The data set defines the maximum-margin separating hyperplane (if it is separable).
- We use quadratic optimization to find this plane.
- Given a new point $\vec{x}$ to classify, the classification function $f(\vec{x})$ computes the functional margin of the point (= normalized distance).
- The sign of this function determines the class to assign to the point.
- If the point is within the margin of the classifier, the classifier can return "don't know" rather than one of the two classes.

Recap

- We start with a training set.
- The data set defines the maximum-margin separating hyperplane (if it is separable).
- We use quadratic optimization to find this plane.
- Given a new point $\vec{x}$ to classify, the classification function $f(\vec{x})$ computes the functional margin of the point (= normalized distance).
- The sign of this function determines the class to assign to the point.
- If the point is within the margin of the classifier, the classifier can return "don't know" rather than one of the two classes.
- The value of $f(\vec{x})$ may also be transformed into a probability of classification

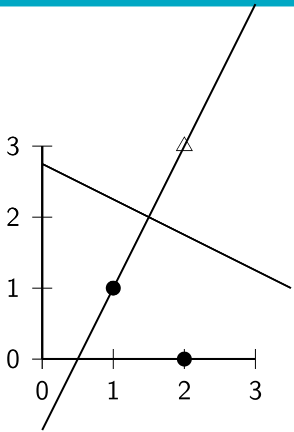
Exercise



Which vectors are the support vectors? Draw the maximum margin separator. What values of w_1, w_2 and b (for $w_1 x + w_2 y + b = 0$) describe this separator? Recall that we must have $w_1 x + w_2 y + b \in \{1, -1\}$ for the support vectors.

Walkthrough example

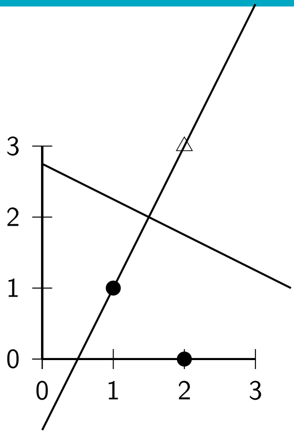
Working geometrically:



Walkthrough example

Working geometrically:

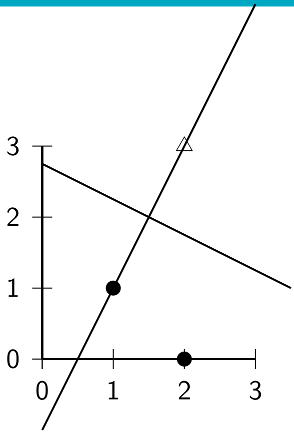
- The maximum margin weight vector will be parallel to the shortest line connecting points of the two classes, that is, the line between $(1,1)$ and $(2,3)$, giving a weight vector of $(1,2)$.



Walkthrough example

Working geometrically:

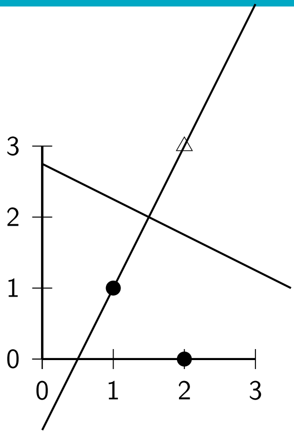
- The maximum margin weight vector will be parallel to the shortest line connecting points of the two classes, that is, the line between $(1, 1)$ and $(2, 3)$, giving a weight vector of $(1, 2)$.
- The optimal decision surface is orthogonal to that line and intersects it at the halfway point. Therefore, it passes through $(1.5, 2)$.



Walkthrough example

Working geometrically:

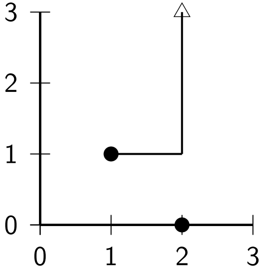
- The maximum margin weight vector will be parallel to the shortest line connecting points of the two classes, that is, the line between $(1, 1)$ and $(2, 3)$, giving a weight vector of $(1, 2)$.
- The optimal decision surface is orthogonal to that line and intersects it at the halfway point. Therefore, it passes through $(1.5, 2)$.
- The SVM decision boundary is:



$$b - b = (1 \cdot x + 2 \cdot y) - (1 \cdot 1.5 + 2 \cdot 2) \Leftrightarrow 0 = \frac{2}{5}x + \frac{4}{5}y - \frac{11}{5}$$

Walkthrough example

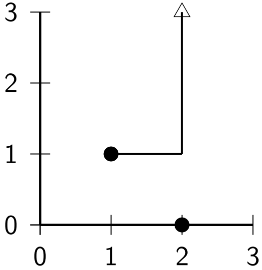
Working algebraically:



Walkthrough example

Working algebraically:

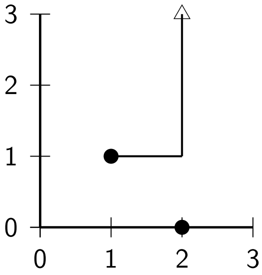
- With the constraint $\text{sign}(y_i(\vec{w}^\mathsf{T}\vec{x}_i + b)) \geq 1$, we seek to minimize $|\vec{w}|$.



Walkthrough example

Working algebraically:

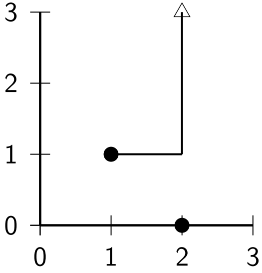
- With the constraint $\text{sign}(y_i(\vec{w}^\mathsf{T}\vec{x}_i + b)) \geq 1$, we seek to minimize $|\vec{w}|$.
- We know that the solution is $\vec{w} = (a, 2a)$ for some a. So: $a + 2a + b = -1$, $2a + 6a + b = 1$



Walkthrough example

Working algebraically:

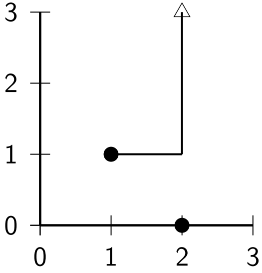
- With the constraint $\text{sign}(y_i(\vec{w}^\mathsf{T}\vec{x}_i + b)) \geq 1$, we seek to minimize $|\vec{w}|$.
- We know that the solution is $\vec{w} = (a, 2a)$ for some a. So: $a + 2a + b = -1$, $2a + 6a + b = 1$
- Hence, $a = 2/5$ and $b = -11/5$. So the optimal hyperplane is given by $\vec{w} = (2/5, 4/5)$ and $b = -11/5$.



Walkthrough example

Working algebraically:

- With the constraint $\text{sign}(y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b)) \geq 1$, we seek to minimize $|\vec{w}|$.
- We know that the solution is $\vec{w} = (a, 2a)$ for some a. So: $a + 2a + b = -1$, $2a + 6a + b = 1$
- Hence, $a = 2/5$ and $b = -11/5$. So the optimal hyperplane is given by $\vec{w} = (2/5, 4/5)$ and $b = -11/5$.
- The margin ρ is $2/|\vec{w}| = 2/\sqrt{4/25 + 16/25} = 2/(2\sqrt{5}/5) = \sqrt{5} = \sqrt{(1-2)^2 + (1-3)^2}$.



Soft margin classification

Soft margin classification

What happens if data is not linearly separable?

Soft margin classification

What happens if data is not linearly separable?

- Standard approach: allow the fat decision margin to make a few mistakes

Soft margin classification

What happens if data is not linearly separable?

- Standard approach: allow the fat decision margin to make a few mistakes
  - some points, outliers, noisy examples are inside or on the wrong side of the margin

Soft margin classification

What happens if data is not linearly separable?

- Standard approach: allow the fat decision margin to make a few mistakes
  - some points, outliers, noisy examples are inside or on the wrong side of the margin
- Pay cost for each misclassified example, depending on how far it is from meeting the margin requirement

Soft margin classification

What happens if data is not linearly separable?

- Standard approach: allow the fat decision margin to make a few mistakes
  - some points, outliers, noisy examples are inside or on the wrong side of the margin
- Pay cost for each misclassified example, depending on how far it is from meeting the margin requirement

Slack variable ξ_i: A non-zero value for ξ_i allows $\vec{x}_i$ to not meet the margin requirement at a cost proportional to the value of ξ_i.

Soft margin classification

What happens if data is not linearly separable?

- Standard approach: allow the fat decision margin to make a few mistakes
  - some points, outliers, noisy examples are inside or on the wrong side of the margin
- Pay cost for each misclassified example, depending on how far it is from meeting the margin requirement

Slack variable ξ_i: A non-zero value for ξ_i allows $\vec{x}_i$ to not meet the margin requirement at a cost proportional to the value of ξ_i.

Optimization problem: trading off how fat it can make the margin vs. how many points have to be moved around to allow this margin.

Soft margin classification

What happens if data is not linearly separable?

- Standard approach: allow the fat decision margin to make a few mistakes
    - some points, outliers, noisy examples are inside or on the wrong side of the margin
- Pay cost for each misclassified example, depending on how far it is from meeting the margin requirement

Slack variable ξ_i: A non-zero value for ξ_i allows $\vec{x}_i$ to not meet the margin requirement at a cost proportional to the value of ξ_i.

Optimization problem: trading off how fat it can make the margin vs. how many points have to be moved around to allow this margin.

The sum of the ξ_i gives an upper bound on the number of training errors. Soft-margin SVMs minimize training error traded off against margin.

Using SVM for one-of classification

Using SVM for one-of classification

- Recall how to use binary linear classifiers (k classes) for one-of: train and run k classifiers and then select the class with the highest confidence

Using SVM for one-of classification

- Recall how to use binary linear classifiers (k classes) for one-of: train and run k classifiers and then select the class with the highest confidence
- Another strategy used with SVMs: build $k(k-1)/2$ one-versus-one classifiers, and choose the class that is selected by the most classifiers. While this involves building a very large number of classifiers, the time for training classifiers may actually decrease, since the training data set for each classifier is much smaller.

Using SVM for one-of classification

- Recall how to use binary linear classifiers (k classes) for one-of: train and run k classifiers and then select the class with the highest confidence
- Another strategy used with SVMs: build $k(k-1)/2$ one-versus-one classifiers, and choose the class that is selected by the most classifiers. While this involves building a very large number of classifiers, the time for training classifiers may actually decrease, since the training data set for each classifier is much smaller.
- Yet another possibility: structured prediction. Generalization of classification where the classes are not just a set of independent, categorical labels, but may be arbitrary structured objects with relationships defined between them

Outline

Text classification

Text classification

- Many commercial applications

Text classification

- Many commercial applications
- There are many applications of text classification for corporate Intranets, government departments, and Internet publishers.

Text classification

- Many commercial applications
- There are many applications of text classification for corporate Intranets, government departments, and Internet publishers.
- Often greater performance gains from exploiting domain-specific text features than from changing from one machine learning method to another.

Text classification

- Many commercial applications
- There are many applications of text classification for corporate Intranets, government departments, and Internet publishers.
- Often greater performance gains from exploiting domain-specific text features than from changing from one machine learning method to another.
- Understanding the data is one of the keys to successful categorization, yet this is an area in which many categorization tool vendors are weak.

Choosing what kind of classifier to use

Choosing what kind of classifier to use

When building a text classifier, first question: how much training data is there currently available?

Choosing what kind of classifier to use

When building a text classifier, first question: how much training data is there currently available?

Practical challenge: creating or obtaining enough training data

Hundreds or thousands of examples from each class are required to produce a high performance classifier and many real world contexts involve large sets of categories.

Choosing what kind of classifier to use

When building a text classifier, first question: how much training data is there currently available?

Practical challenge: creating or obtaining enough training data

Hundreds or thousands of examples from each class are required to produce a high performance classifier and many real world contexts involve large sets of categories.

- None?
- Very little?
- Quite a lot?
- A huge amount, growing every day?

If you have no labeled training data

If you have no labeled training data

Use hand-written rules!

If you have no labeled training data

Use hand-written rules!

Example

IF (wheat OR grain) AND NOT (whole OR bread) THEN
$c = $ grain

If you have no labeled training data

Use hand-written rules!

Example

IF (wheat OR grain) AND NOT (whole OR bread) THEN
$c =$ grain

In practice, rules get a lot bigger than this, and can be phrased using more sophisticated query languages than just Boolean expressions, including the use of numeric scores.

If you have no labeled training data

Use hand-written rules!

Example

IF (wheat OR grain) AND NOT (whole OR bread) THEN
$c =$ grain

In practice, rules get a lot bigger than this, and can be phrased using more sophisticated query languages than just Boolean expressions, including the use of numeric scores.

With careful crafting, the accuracy of such rules can become very high (high 90% precision, high 80% recall).

If you have no labeled training data

Use hand-written rules!

Example

IF (wheat OR grain) AND NOT (whole OR bread) THEN
$c =$ grain

In practice, rules get a lot bigger than this, and can be phrased using more sophisticated query languages than just Boolean expressions, including the use of numeric scores.
With careful crafting, the accuracy of such rules can become very high (high 90% precision, high 80% recall).
Nevertheless the amount of work to create such well-tuned rules is very large. A reasonable estimate is 2 days per class, and extra time has to go into maintenance of rules, as the content of documents in classes drifts over time.

A Verity topic (a complex classification rule)

comment line	# Beginning of art topic definition
top-level topic	art ACCRUE
topic definition modifiers	/author = "fsmith"
	/date = "30-Dec-01"
	/annotation = "Topic created by fsmith"
subtopictopic	* 0.70 performing-arts ACCRUE
evidencetopic	** 0.50 WORD
topicdefinition modifier	/wordtext = ballet
evidencetopic	** 0.50 STEM
topicdefinition modifier	/wordtext = dance
evidencetopic	** 0.50 WORD
topicdefinition modifier	/wordtext = opera
evidencetopic	** 0.30 WORD
topicdefinition modifier	/wordtext = symphony
subtopic	* 0.70 visual-arts ACCRUE
	** 0.50 WORD
	/wordtext = painting
	** 0.50 WORD
	/wordtext = sculpture
subtopic	* 0.70 film ACCRUE
	** 0.50 STEM
	/wordtext = film
subtopic	** 0.50 motion-picture PHRAS
	*** 1.00 WORD
	/wordtext = motion
	*** 1.00 WORD
	/wordtext = picture
	** 0.50 STEM
	/wordtext = movie
subtopic	* 0.50 video ACCRUE
	** 0.50 STEM
	/wordtext = video
	** 0.50 STEM
	/wordtext = vcr
	# End of art topic

Westlaw: Example queries

Information need: Information on the legal theories involved in preventing the disclosure of trade secrets by employees formerly employed by a competing company

Query: "trade secret" /s disclos! /s prevent /s employe!

Information need: Requirements for disabled people to be able to access a workplace

Query: disab! /p access! /s work-site work-place (employment /3 place)

Information need: Cases about a host's responsibility for drunk guests

Query: host! /p (responsib! liab!) /p (intoxicat! drunk!) /p guest

If you have fairly little data and you are going to train a supervised classifier

If you have fairly little data and you are going to train a supervised classifier

Work out how to get more labeled data as quickly as you can.

If you have fairly little data and you are going to train a supervised classifier

Work out how to get more labeled data as quickly as you can.

- Best way: insert yourself into a process where humans will be willing to label data for you as part of their natural tasks.

If you have fairly little data and you are going to train a supervised classifier

Work out how to get more labeled data as quickly as you can.

- Best way: insert yourself into a process where humans will be willing to label data for you as part of their natural tasks.

Example

Often humans will sort or route email for their own purposes, and these actions give information about classes.

If you have fairly little data and you are going to train a supervised classifier

Work out how to get more labeled data as quickly as you can.

- Best way: insert yourself into a process where humans will be willing to label data for you as part of their natural tasks.

Example

Often humans will sort or route email for their own purposes, and these actions give information about classes.

Active Learning

A system is built which decides which documents a human should label.

If you have fairly little data and you are going to train a supervised classifier

Work out how to get more labeled data as quickly as you can.

- Best way: insert yourself into a process where humans will be willing to label data for you as part of their natural tasks.

Example

Often humans will sort or route email for their own purposes, and these actions give information about classes.

Active Learning

A system is built which decides which documents a human should label.
Usually these are the ones on which a classifier is uncertain of the correct classification.

If you have labeled data

If you have labeled data

Good amount of labeled data, but not huge

Use everything that we have presented about text classification.

If you have labeled data

Good amount of labeled data, but not huge

Use everything that we have presented about text classification.
Consider hybrid approach (overlay Boolean classifier)

If you have labeled data

Good amount of labeled data, but not huge

Use everything that we have presented about text classification.
Consider hybrid approach (overlay Boolean classifier)

Huge amount of labeled data

Choice of classifier probably has little effect on your results.

If you have labeled data

Good amount of labeled data, but not huge

Use everything that we have presented about text classification.
Consider hybrid approach (overlay Boolean classifier)

Huge amount of labeled data

Choice of classifier probably has little effect on your results.
Choose classifier based on the scalability of training or runtime efficiency.

If you have labeled data

Good amount of labeled data, but not huge

Use everything that we have presented about text classification.
Consider hybrid approach (overlay Boolean classifier)

Huge amount of labeled data

Choice of classifier probably has little effect on your results.
Choose classifier based on the scalability of training or runtime efficiency.
Rule of thumb: each doubling of the training data size produces a linear increase in classifier performance, but with very large amounts of data, the improvement becomes sub-linear.

Large and difficult category taxonomies

Large and difficult category taxonomies

If you have a small number of well-separated categories, then many classification algorithms are likely to work well. But often: very large number of very similar categories.

Large and difficult category taxonomies

If you have a small number of well-separated categories, then many classification algorithms are likely to work well. But often: very large number of very similar categories.

Example

Web directories (e.g. the Yahoo! Directory consists of over 200,000 categories or the Open Directory Project), library classification schemes (Dewey Decimal or Library of Congress), the classification schemes used in legal or medical applications.

Large and difficult category taxonomies

If you have a small number of well-separated categories, then many classification algorithms are likely to work well. But often: very large number of very similar categories.

Example

Web directories (e.g. the Yahoo! Directory consists of over 200,000 categories or the Open Directory Project), library classification schemes (Dewey Decimal or Library of Congress), the classification schemes used in legal or medical applications.

Accurate classification over large sets of closely related classes is inherently difficult. – No general high-accuracy solution.

Recap

Recap

- Is there a learning method that is optimal for all text classification problems?

Recap

- Is there a learning method that is optimal for all text classification problems?
- No, because there is a tradeoff between bias and variance.

Recap

- Is there a learning method that is optimal for all text classification problems?
- No, because there is a tradeoff between bias and variance.
- Factors to take into account:

Recap

- Is there a learning method that is optimal for all text classification problems?
- No, because there is a tradeoff between bias and variance.
- Factors to take into account:
  - How much training data is available?

Recap

- Is there a learning method that is optimal for all text classification problems?
- No, because there is a tradeoff between bias and variance.
- Factors to take into account:
  - How much training data is available?
  - How simple/complex is the problem? (linear vs. nonlinear decision boundary)

Recap

- Is there a learning method that is optimal for all text classification problems?
- No, because there is a tradeoff between bias and variance.
- Factors to take into account:
  - How much training data is available?
  - How simple/complex is the problem? (linear vs. nonlinear decision boundary)
  - How noisy is the problem?

Recap

- Is there a learning method that is optimal for all text classification problems?
- No, because there is a tradeoff between bias and variance.
- Factors to take into account:
  - How much training data is available?
  - How simple/complex is the problem? (linear vs. nonlinear decision boundary)
  - How noisy is the problem?
  - How stable is the problem over time?

Recap

- Is there a learning method that is optimal for all text classification problems?
- No, because there is a tradeoff between bias and variance.
- Factors to take into account:
    - How much training data is available?
    - How simple/complex is the problem? (linear vs. nonlinear decision boundary)
    - How noisy is the problem?
    - How stable is the problem over time?
        - For an unstable problem, it's better to use a simple and robust classifier.

Exercise

You are tasked with building a system that monitors the sentiment expressed by tweeters about a company.
Functionality: the user enters a set of #hashtags, @usernames and keyword queries that are related to the company of interest. The system then computes the proportion of positive and negative sentiment in the messages containing these #hashtags, @usernames and queries.
A key part of this system is a classifier that takes a tweet and classifies it as having positive or negative polarity.
How would you build this classifier? You can use a rule-based or a statistical or a hybrid approach.

Take-away today

- Support vector machines: State-of-the-art text classification methods (linear and nonlinear)
- Introduction to SVMs
- Formalization
- Soft margin case for nonseparable problems
- Discussion: Which classifier should I use for my problem?

Resources

- Chapter 14 of IIR (basic vector space classification)
- Chapter 15 of IIR (SVMs)
- Discussion of "how to select the right classifier for my problem" in Russell and Norvig
- Resources at `http://cislmu.org`