Introduction to Information Retrieval

http://informationretrieval.org

IIR 15-1: Support Vector Machines

Hinrich Schütze

Institute for Natural Language Processing, Universität Stuttgart

2011-12-13

Overview

1. Recap
2. Vector space classification
3. Linear classifiers
4. Support Vector Machines
5. Discussion

Outline

Feature selection: MI for *poultry*/EXPORT

Goal of feature selection: eleminate noise and useless features for better effectiveness and efficiency

	$e_c = e_{poultry} = 1$	$e_c = e_{poultry} = 0$
$e_t = e_{\text{EXPORT}} = 1$	$N_{11} = 49$	$N_{10} = 27{,}652$
$e_t = e_{\text{EXPORT}} = 0$	$N_{01} = 141$	$N_{00} = 774{,}106$

Plug these values into formula:

$$\begin{aligned}
I(U;C) &= \frac{49}{801{,}948}\log_2\frac{801{,}948\cdot 49}{(49+27{,}652)(49+141)} \\
&+\frac{141}{801{,}948}\log_2\frac{801{,}948\cdot 141}{(141+774{,}106)(49+141)} \\
&+\frac{27{,}652}{801{,}948}\log_2\frac{801{,}948\cdot 27{,}652}{(49+27{,}652)(27{,}652+774{,}106)} \\
&+\frac{774{,}106}{801{,}948}\log_2\frac{801{,}948\cdot 774{,}106}{(141+774{,}106)(27{,}652+774{,}106)} \\
&\approx 0.000105
\end{aligned}$$

Feature selection for Reuters classes coffee and sports

Class: *coffee*

term	MI
COFFEE	0.0111
BAGS	0.0042
GROWERS	0.0025
KG	0.0019
COLOMBIA	0.0018
BRAZIL	0.0016
EXPORT	0.0014
EXPORTERS	0.0013
EXPORTS	0.0013
CROP	0.0012

Class: *sports*

term	MI
SOCCER	0.0681
CUP	0.0515
MATCH	0.0441
MATCHES	0.0408
PLAYED	0.0388
LEAGUE	0.0386
BEAT	0.0301
GAME	0.0299
GAMES	0.0284
TEAM	0.0264

Using language models (LMs) for IR

- LM = language model
- We view the document as a generative model that generates the query.
- What we need to do:
- Define the precise generative model we want to use
- Estimate parameters (different parameters for each document's model)
- Smooth to avoid zeros
- Apply to query and find document most likely to have generated the query
- Present most likely document(s) to user

Jelinek-Mercer smoothing

- $P(t|d) = \lambda P(t|M_d) + (1 - \lambda)P(t|M_c)$
- Mixes the probability from the document with the general collection frequency of the word.
- High value of λ: "conjunctive-like" search – tends to retrieve documents containing all query words.
- Low value of λ: more disjunctive, suitable for long queries
- Correctly setting λ is very important for good performance.

Take-away today

Take-away today

- Vector space text classification: Classification defined as the problem of finding a separating hyperplane in high-dimensional space

Take-away today

- Vector space text classification: Classification defined as the problem of finding a separating hyperplane in high-dimensional space
- Linear vector space classifiers: Simple, but often very effective

Take-away today

- Vector space text classification: Classification defined as the problem of finding a separating hyperplane in high-dimensional space
- Linear vector space classifiers: Simple, but often very effective
- Support vector machines: State-of-the-art text classification methods (linear and nonlinear)

Take-away today

- Vector space text classification: Classification defined as the problem of finding a separating hyperplane in high-dimensional space
- Linear vector space classifiers: Simple, but often very effective
- Support vector machines: State-of-the-art text classification methods (linear and nonlinear)
- Discussion: Which classifier should I use for my problem?

Outline

Recall vector space representation

Recall vector space representation

- Each document is a vector, one component for each term.

Recall vector space representation

- Each document is a vector, one component for each term.
- Terms are axes.

Recall vector space representation

- Each document is a vector, one component for each term.
- Terms are axes.
- High dimensionality: 10,000s of dimensions and more

Recall vector space representation

- Each document is a vector, one component for each term.
- Terms are axes.
- High dimensionality: 10,000s of dimensions and more
- Normalize vectors (documents) to unit length

Recall vector space representation

- Each document is a vector, one component for each term.
- Terms are axes.
- High dimensionality: 10,000s of dimensions and more
- Normalize vectors (documents) to unit length
- How can we do classification in this space?

Vector space classification

Vector space classification

- As before, the training set is a set of documents, each labeled with its class.

Vector space classification

- As before, the training set is a set of documents, each labeled with its class.
- In vector space classification, this set corresponds to a labeled set of points or vectors in the vector space.

Vector space classification

- As before, the training set is a set of documents, each labeled with its class.
- In vector space classification, this set corresponds to a labeled set of points or vectors in the vector space.
- Premise 1: Documents in the same class form a contiguous region.

Vector space classification

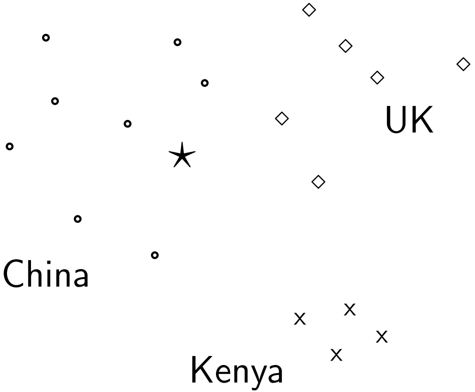
- As before, the training set is a set of documents, each labeled with its class.
- In vector space classification, this set corresponds to a labeled set of points or vectors in the vector space.
- Premise 1: Documents in the same class form a contiguous region.
- Premise 2: Documents from different classes don't overlap.

Vector space classification

- As before, the training set is a set of documents, each labeled with its class.
- In vector space classification, this set corresponds to a labeled set of points or vectors in the vector space.
- Premise 1: Documents in the same class form a contiguous region.
- Premise 2: Documents from different classes don't overlap.
- We define lines, surfaces, hypersurfaces to divide regions.

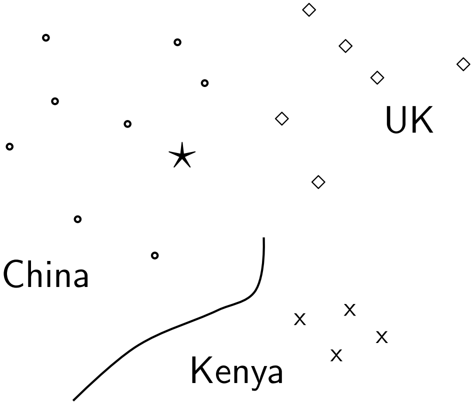
Classes in the vector space

Classes in the vector space



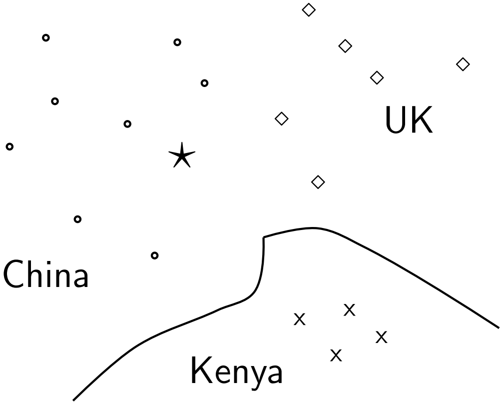
Should the document $\star$ be assigned to *China*, *UK* or *Kenya*?

Classes in the vector space



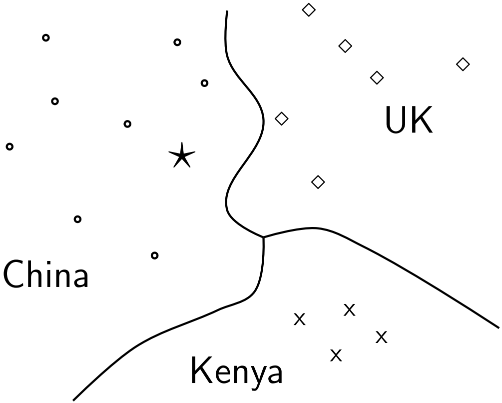
Find separators between the classes

Classes in the vector space



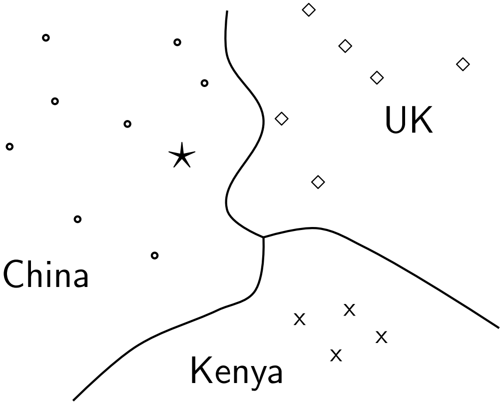
Find separators between the classes

Classes in the vector space



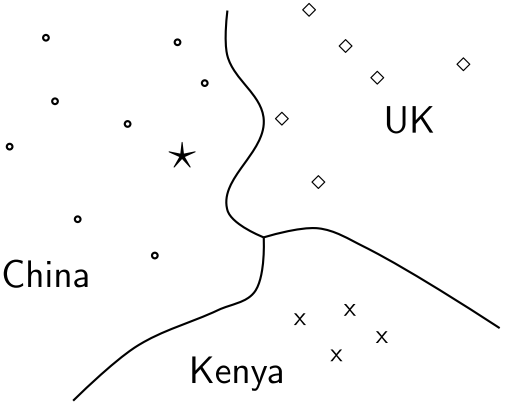
Find separators between the classes

Classes in the vector space



Based on these separators: $\star$ should be assigned to *China*

Classes in the vector space



How do we find separators that do a good job at classifying new documents like $\star$? – Main topic of today

Outline

1. Recap
2. Vector space classification
3. Linear classifiers
4. Support Vector Machines
5. Discussion

Linear classifiers

Linear classifiers

- Definition:

Linear classifiers

- Definition:
  - A linear classifier computes a linear combination or weighted sum $\sum_i w_i x_i$ of the feature values.

Linear classifiers

- Definition:
  - A linear classifier computes a linear combination or weighted sum $\sum_i w_i x_i$ of the feature values.
  - Classification decision: $\sum_i w_i x_i > \theta$?

Linear classifiers

- Definition:
  - A linear classifier computes a linear combination or weighted sum $\sum_i w_i x_i$ of the feature values.
  - Classification decision: $\sum_i w_i x_i > \theta$?
  - ... where θ (the threshold) is a parameter.

Linear classifiers

- Definition:
  - A linear classifier computes a linear combination or weighted sum $\sum_i w_i x_i$ of the feature values.
  - Classification decision: $\sum_i w_i x_i > \theta$?
  - ... where θ (the threshold) is a parameter.
- (First, we only consider binary classifiers.)

Linear classifiers

- Definition:
  - A linear classifier computes a linear combination or weighted sum $\sum_i w_i x_i$ of the feature values.
  - Classification decision: $\sum_i w_i x_i > \theta$?
  - ... where θ (the threshold) is a parameter.
- (First, we only consider binary classifiers.)
- Geometrically, this corresponds to a line (2D), a plane (3D) or a hyperplane (higher dimensionalities).

Linear classifiers

- Definition:
  - A linear classifier computes a linear combination or weighted sum $\sum_i w_i x_i$ of the feature values.
  - Classification decision: $\sum_i w_i x_i > \theta$?
  - . . . where θ (the threshold) is a parameter.
- (First, we only consider binary classifiers.)
- Geometrically, this corresponds to a line (2D), a plane (3D) or a hyperplane (higher dimensionalities).
- We call this the separator or decision boundary.

Linear classifiers

- Definition:
  - A linear classifier computes a linear combination or weighted sum $\sum_i w_i x_i$ of the feature values.
  - Classification decision: $\sum_i w_i x_i > \theta$?
  - ... where θ (the threshold) is a parameter.
- (First, we only consider binary classifiers.)
- Geometrically, this corresponds to a line (2D), a plane (3D) or a hyperplane (higher dimensionalities).
- We call this the separator or decision boundary.
- We find the separator based on training set.

Linear classifiers

- Definition:
  - A linear classifier computes a linear combination or weighted sum $\sum_i w_i x_i$ of the feature values.
  - Classification decision: $\sum_i w_i x_i > \theta$?
  - ... where θ (the threshold) is a parameter.
- (First, we only consider binary classifiers.)
- Geometrically, this corresponds to a line (2D), a plane (3D) or a hyperplane (higher dimensionalities).
- We call this the separator or decision boundary.
- We find the separator based on training set.
- Methods for finding separator: Perceptron, Rocchio, Naive Bayes, linear SVM

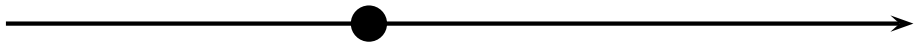
Linear classifiers

- Definition:
  - A linear classifier computes a linear combination or weighted sum $\sum_i w_i x_i$ of the feature values.
  - Classification decision: $\sum_i w_i x_i > \theta$?
  - ... where θ (the threshold) is a parameter.
- (First, we only consider binary classifiers.)
- Geometrically, this corresponds to a line (2D), a plane (3D) or a hyperplane (higher dimensionalities).
- We call this the separator or decision boundary.
- We find the separator based on training set.
- Methods for finding separator: Perceptron, Rocchio, Naive Bayes, linear SVM
- Assumption: The classes are linearly separable.

A linear classifier in 1D

- A linear classifier in 1D is a point x described by the equation $w_1 d_1 = \theta$

A linear classifier in 1D

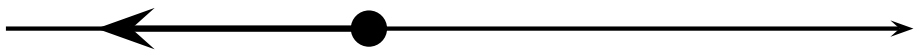


- A linear classifier in 1D is a point x described by the equation $w_1 d_1 = \theta$
- $x = \theta / w_1$

A linear classifier in 1D

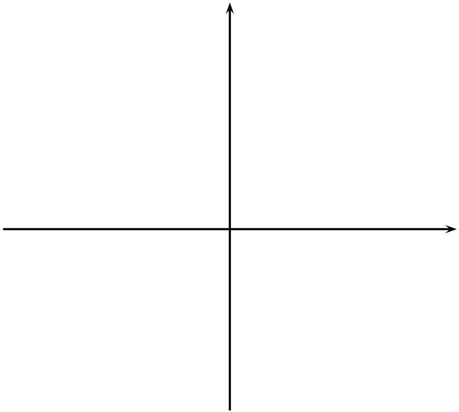
- A linear classifier in 1D is a point x described by the equation $w_1 d_1 = \theta$
- $x = \theta / w_1$
- Points (d_1) with $w_1 d_1 \geq \theta$ are in the class c.

A linear classifier in 1D



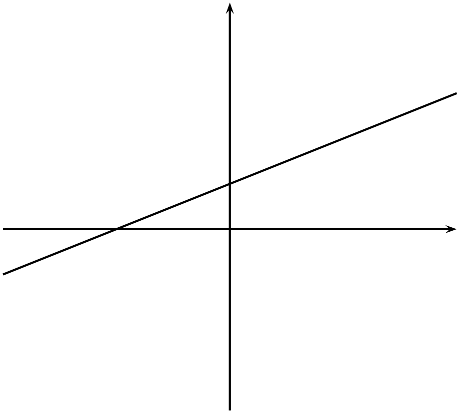
- A linear classifier in 1D is a point x described by the equation $w_1 d_1 = \theta$
- $x = \theta / w_1$
- Points (d_1) with $w_1 d_1 \geq \theta$ are in the class c.
- Points (d_1) with $w_1 d_1 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 2D



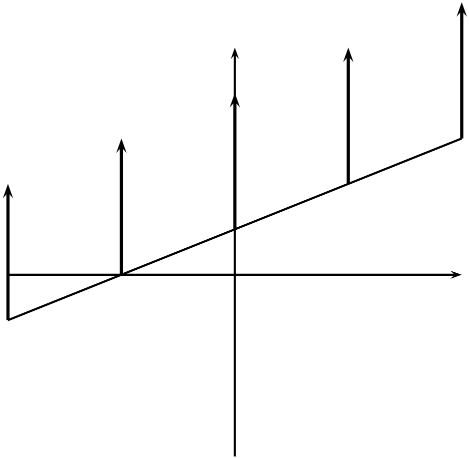
- A linear classifier in 2D is a line described by the equation $w_1 d_1 + w_2 d_2 = \theta$

A linear classifier in 2D



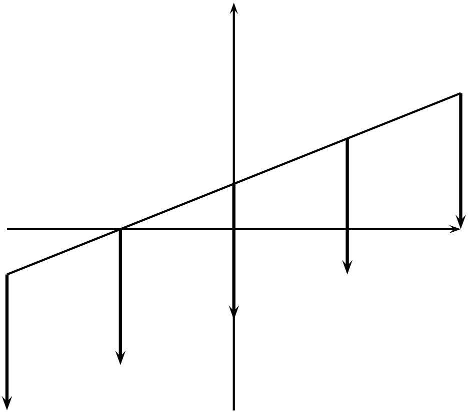
- A linear classifier in 2D is a line described by the equation $w_1 d_1 + w_2 d_2 = \theta$
- Example for a 2D linear classifier

A linear classifier in 2D



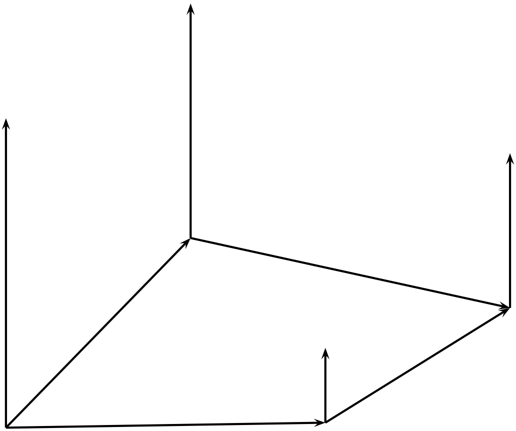
- A linear classifier in 2D is a line described by the equation $w_1 d_1 + w_2 d_2 = \theta$
- Example for a 2D linear classifier
- Points $(d_1 \ d_2)$ with $w_1 d_1 + w_2 d_2 \geq \theta$ are in the class c.

A linear classifier in 2D



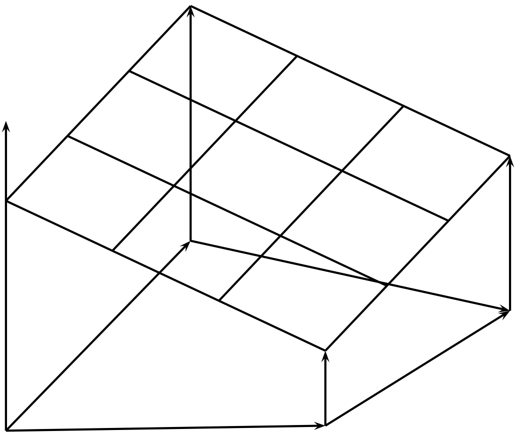
- A linear classifier in 2D is a line described by the equation $w_1 d_1 + w_2 d_2 = \theta$
- Example for a 2D linear classifier
- Points $(d_1 \ d_2)$ with $w_1 d_1 + w_2 d_2 \geq \theta$ are in the class c.
- Points $(d_1 \ d_2)$ with $w_1 d_1 + w_2 d_2 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 3D



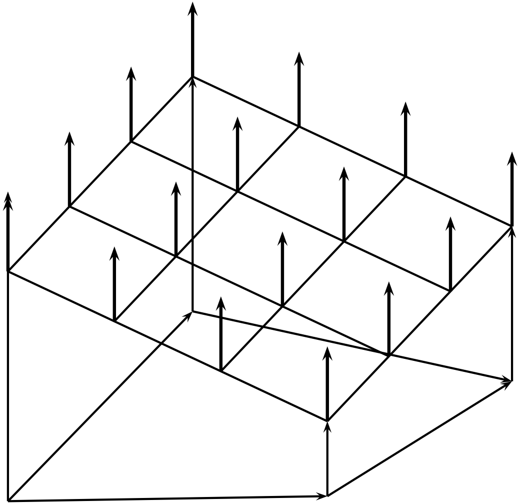
- A linear classifier in 3D is a plane described by the equation $w_1 d_1 + w_2 d_2 + w_3 d_3 = \theta$

A linear classifier in 3D



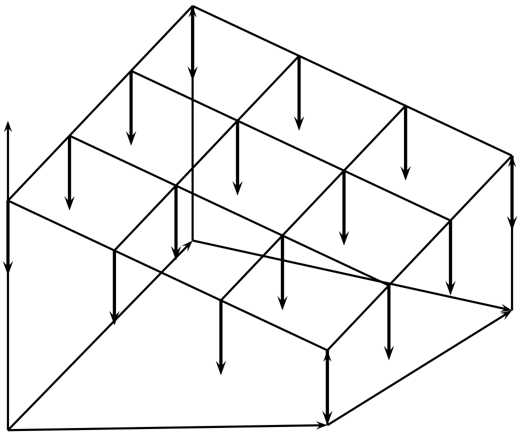
- A linear classifier in 3D is a plane described by the equation $w_1 d_1 + w_2 d_2 + w_3 d_3 = \theta$
- Example for a 3D linear classifier

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.

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 $\overline{c}$.

Naive Bayes as a linear classifier

Naive Bayes as a linear classifier

Multinomial Naive Bayes is a linear classifier (in log space) defined by:

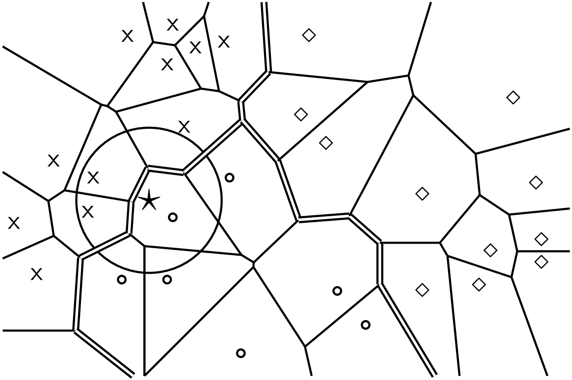
$$\sum_{i=1}^{M} w_i d_i = \theta$$

where $w_i = \log[\hat{P}(t_i|c)/\hat{P}(t_i|\bar{c})]$, $d_i =$ number of occurrences of t_i in d, and $\theta = -\log[\hat{P}(c)/\hat{P}(\bar{c})]$. Here, the index i, $1 \leq i \leq M$, refers to terms of the vocabulary (not to positions in d as k did in our original definition of Naive Bayes)

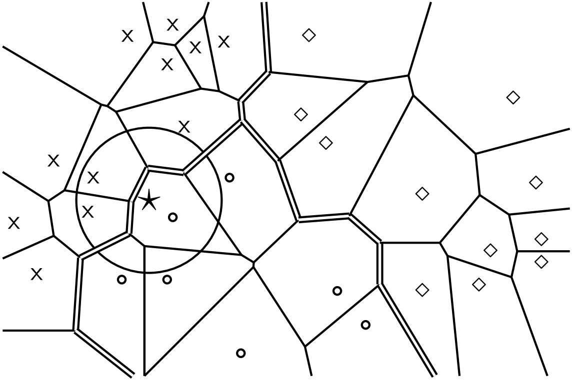
k nearest neighbors is not a linear classifier

k nearest neighbors is not a linear classifier

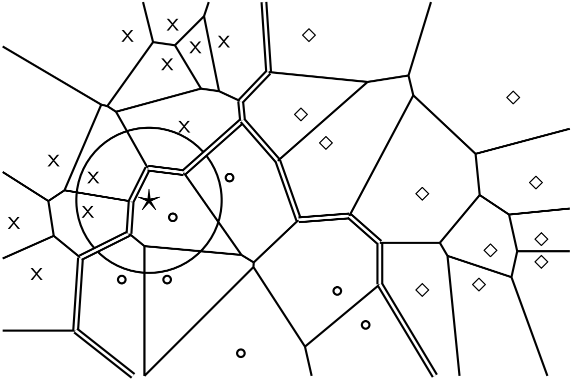
k nearest neighbors is not a linear classifier



k nearest neighbors is not a linear classifier

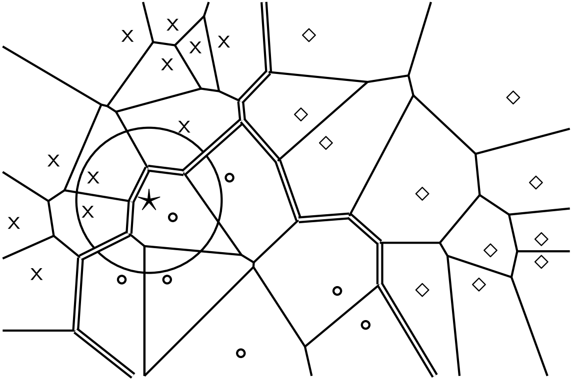


k nearest neighbors is not a linear classifier



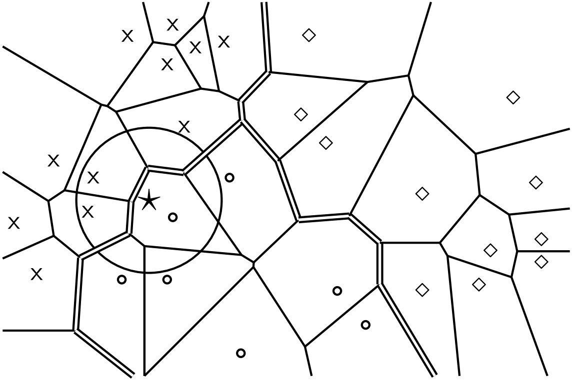
- Classification decision based on majority of *k* nearest neighbors.

k nearest neighbors is not a linear classifier



- Classification decision based on majority of *k* nearest neighbors.
- The decision boundaries between classes are piecewise linear . . .

k nearest neighbors is not a linear classifier



- Classification decision based on majority of k nearest neighbors.
- The decision boundaries between classes are piecewise linear ...
- ... but they are in general not linear classifiers that can be described as $\sum_{i=1}^{M} w_i d_i = \theta$.

Example of a linear classifier

t_i	w_i	d_{1i}	d_{2i}	t_i	w_i	d_{1i}	d_{2i}
prime	0.70	0	1	dlrs	-0.71	1	1
rate	0.67	1	0	world	-0.35	1	0
interest	0.63	0	0	sees	-0.33	0	0
rates	0.60	0	0	year	-0.25	0	0
discount	0.46	1	0	group	-0.24	0	0
bundesbank	0.43	0	0	dlr	-0.24	0	0

- This is for the class *interest* in Reuters-21578.
- For simplicity: assume a simple 0/1 vector representation
- d_1: “rate discount dlrs world”
- d_2: “prime dlrs”
- $\theta = 0$
- Exercise: Which class is d_1 assigned to? Which class is d_2 assigned to?

Example of a linear classifier

t_i	w_i	d_{1i}	d_{2i}	t_i	w_i	d_{1i}	d_{2i}
prime	0.70	0	1	dlrs	-0.71	1	1
rate	0.67	1	0	world	-0.35	1	0
interest	0.63	0	0	sees	-0.33	0	0
rates	0.60	0	0	year	-0.25	0	0
discount	0.46	1	0	group	-0.24	0	0
bundesbank	0.43	0	0	dlr	-0.24	0	0

- This is for the class *interest* in Reuters-21578.
- For simplicity: assume a simple 0/1 vector representation
- d_1: "rate discount dlrs world"
- d_2: "prime dlrs"
- $\theta = 0$
- Exercise: Which class is d_1 assigned to? Which class is d_2 assigned to?

Example of a linear classifier

t_i	w_i	d_{1i}	d_{2i}	t_i	w_i	d_{1i}	d_{2i}
prime	0.70	0	1	dlrs	-0.71	1	1
rate	0.67	1	0	world	-0.35	1	0
interest	0.63	0	0	sees	-0.33	0	0
rates	0.60	0	0	year	-0.25	0	0
discount	0.46	1	0	group	-0.24	0	0
bundesbank	0.43	0	0	dlr	-0.24	0	0

- This is for the class *interest* in Reuters-21578.
- For simplicity: assume a simple 0/1 vector representation
- d_1: "rate discount dlrs world"
- d_2: "prime dlrs"
- $\theta = 0$
- Exercise: Which class is d_1 assigned to? Which class is d_2 assigned to?

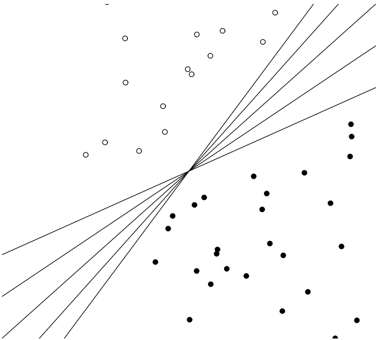
Example of a linear classifier

t_i	w_i	d_{1i}	d_{2i}	t_i	w_i	d_{1i}	d_{2i}
prime	0.70	0	1	dlrs	-0.71	1	1
rate	0.67	1	0	world	-0.35	1	0
interest	0.63	0	0	sees	-0.33	0	0
rates	0.60	0	0	year	-0.25	0	0
discount	0.46	1	0	group	-0.24	0	0
bundesbank	0.43	0	0	dlr	-0.24	0	0

- This is for the class *interest* in Reuters-21578.
- For simplicity: assume a simple 0/1 vector representation
- d_1: "rate discount dlrs world"
- d_2: "prime dlrs"
- $\theta = 0$
- Exercise: Which class is d_1 assigned to? Which class is d_2 assigned to?
- We assign document $\vec{d}_1$ "rate discount dlrs world" to *interest* since $\vec{w}^T\vec{d}_1 = 0.67 \cdot 1 + 0.46 \cdot 1 + (-0.71) \cdot 1 + (-0.35) \cdot 1 = 0.07 > 0 = \theta$.
- We assign $\vec{d}_2$ "prime dlrs" to the complement class (not in *interest*) since $\vec{w}^T\vec{d}_2 = -0.01 \leq \theta$.

Which hyperplane?

Which hyperplane?



Learning algorithms for vector space classification

Learning algorithms for vector space classification

- In terms of actual computation, there are two types of learning algorithms.

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.

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.

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

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

Learning algorithms for vector space classification

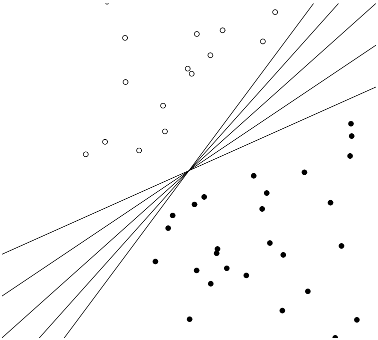
- In terms of actual computation, there are two types of learning algorithms.
- (i) Simple learning algorithms that estimate the parameters of the classifier directly from the training data, often in one linear pass.
  - Naive Bayes, Rocchio, kNN are all examples of this.
- (ii) Iterative algorithms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

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://ifnlp.org/ir/pdfnew/p.pdf)
- The best performing learning algorithms usually require iterative learning.

Which hyperplane?

Which hyperplane?



Which hyperplane?

Which hyperplane?

- For linearly separable training sets: there are infinitely many separating hyperplanes.

Which hyperplane?

- For linearly separable training sets: there are infinitely many separating hyperplanes.
- They all separate the training set perfectly . . .

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.

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.

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?

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?
- Simple perceptron: generally bad; Naive Bayes, Rocchio: ok; linear SVM, "advanced" perceptrons: good

Linear classifiers: Discussion

Linear classifiers: Discussion

- Many common text classifiers are linear classifiers: Naive Bayes, Rocchio, logistic regression, linear support vector machines etc.

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

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

Linear classifiers: Discussion

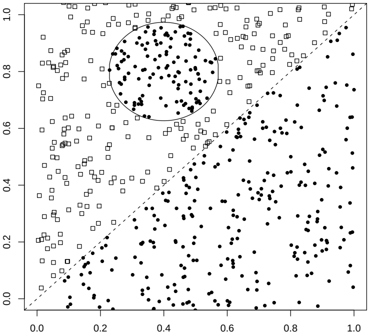
- Many common text classifiers are linear classifiers: Naive Bayes, Rocchio, logistic regression, linear support vector machines etc.
- Each method has a different way of selecting the separating hyperplane
  - Huge differences in performance on test documents
- Can we get better performance with more powerful nonlinear classifiers?

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.

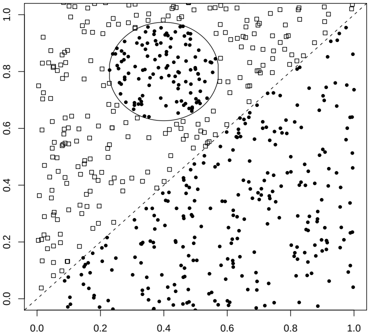
A nonlinear problem

A nonlinear problem



- Linear classifier like Naive Bayes does badly on this task.

A nonlinear problem



- Linear classifier like Naive Bayes does badly on this task.
- kNN will do well (assuming enough training data)

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, 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, neural networks, and random forests
- Applications to IR problems, particularly text classification

Support vector machines

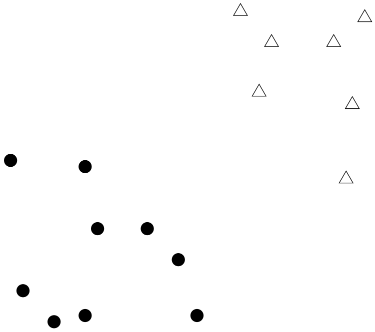
- Machine-learning research in the last two decades has improved classifier effectiveness.
- New generation of state-of-the-art classifiers: support vector machines (SVMs), boosted decision trees, regularized logistic regression, neural networks, and random forests
- Applications to IR problems, particularly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

SVMs: A kind of large-margin classifier

Vector space based machine-learning method aiming to 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)

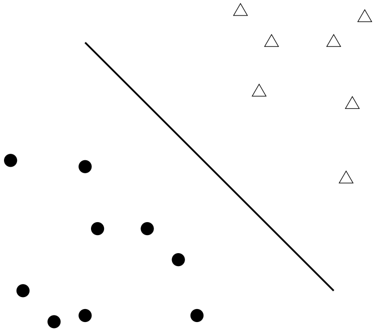
Support Vector Machines

- 2-class training data



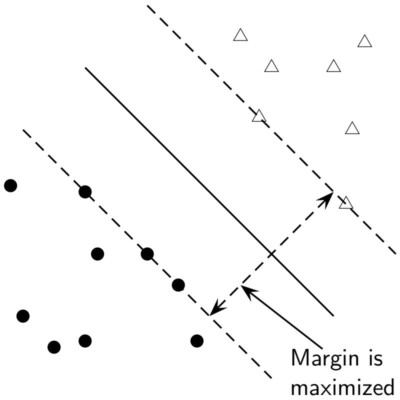
Support Vector Machines

- 2-class training data
- decision boundary
  → **linear separator**



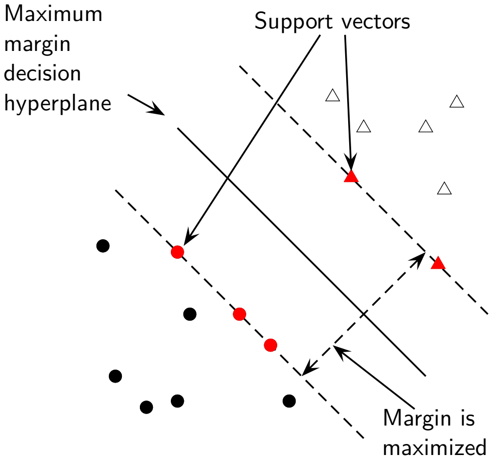
Support Vector Machines

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



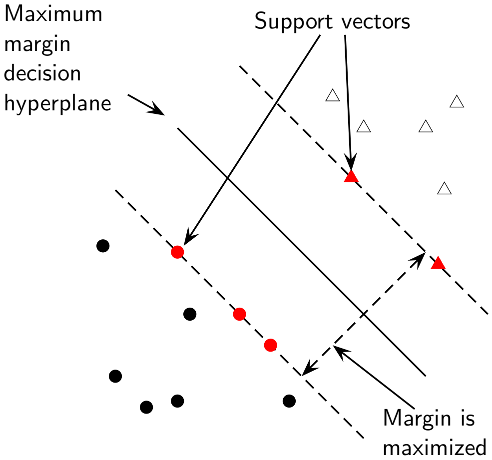
Support Vector Machines

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



Why maximize the margin?

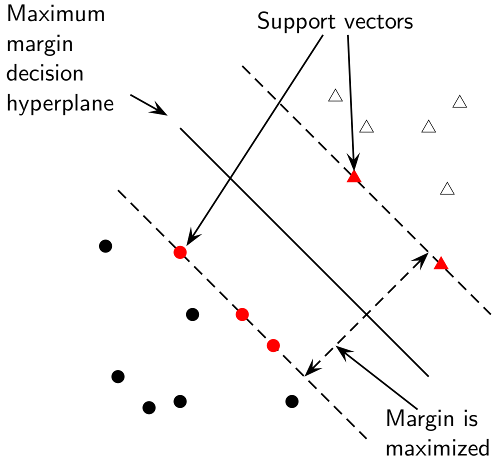
Points near decision surface → uncertain classification decisions (50% either way).



Why maximize the margin?

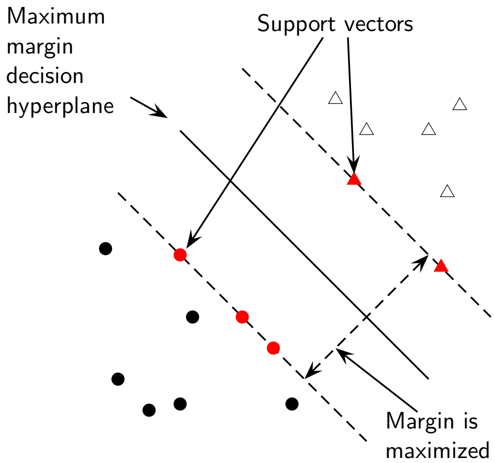
Points near decision surface → uncertain classification decisions (50% either way).

A classifier with a large margin makes no low certainty classification decisions.



Why maximize the margin?

Points near decision surface → uncertain classification decisions (50% either way).
A classifier with a large margin makes no low certainty classification decisions.
Gives classification safety margin w.r.t slight errors in measurement or doc. variation



Why maximize the margin?

Why maximize the margin?

Why maximize the margin?

- SVM classifier: large margin around decision boundary

Why maximize the margin?

- SVM classifier: large margin around decision boundary
- compare to decision hyperplane: place fat separator between classes

Why maximize the margin?

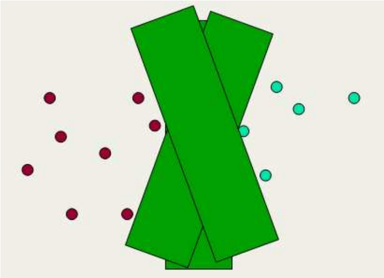
- SVM classifier: large margin around decision boundary
- compare to decision hyperplane: place fat separator between classes
  - unique solution

Why maximize the margin?

- SVM classifier: large margin around decision boundary
- compare to decision hyperplane: place fat separator between classes
  - unique solution
- decreased memory capacity

Why maximize the margin?

- SVM classifier: large margin around decision boundary
- compare to decision hyperplane: place fat separator between classes
  - unique solution
- decreased memory capacity
- 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
- normal vector $\vec{w}$ (**weight vector**) which is perpendicular to the hyperplane

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

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

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 data classes are $y_i = +1$ and $y_i = -1$, and the intercept term is explicitly represented as 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 data classes are $y_i = +1$ and $y_i = -1$, and the intercept term is explicitly represented as b.

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

We are confident in the classification of a point if it is far away from the decision boundary.

Functional margin of a point

We are confident in the classification of a point if it is far away from the decision boundary.

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

We are confident in the classification of a point if it is far away from the decision boundary.

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

We are confident in the classification of a point if it is far away from the decision boundary.

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

But we can increase functional margin by scaling $\vec{w}$ and b.

Functional margin of a point

We are confident in the classification of a point if it is far away from the decision boundary.

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

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

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.

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

The geometric margin is clearly invariant to scaling of parameters: if we replace $\vec{w}$ by $5\vec{w}$ and b by $5b$, then the geometric margin is the same, because it is normalized by the length of $\vec{w}$.

Optimization problem solved by SVMs

Assume canonical distance

Optimization problem solved by SVMs

Assume canonical distance

Assume that all data is 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 distance

Assume that all data is 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 geometric margin is $\rho = 2/|\vec{w}|$.

Optimization problem solved by SVMs

Assume canonical distance

Assume that all data is 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 geometric margin is $\rho = 2/|\vec{w}|$.

We want to maximize this geometric margin.

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

- $\rho = 2/|\vec{w}|$ is maximized
- 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 (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:

Example

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:

Example

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

- 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 projection of the point onto the hyperplane normal.
- 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

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
  - 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 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.

Binary classification → One-of multiclass classification

- Many classification algorithms are binary.
- What do we do for one-of multiclass classification: we have $k > 2$ classes and the k classes are mutually exclusive?
- Common technique: build $|\mathbb{C}|$ one-versus-rest classifiers (commonly referred to as "one-versus-all" or OVA classification), and choose the class which classifies the test data with highest probability (probabilistic classifier) or greatest margin (SVM)
- Another strategy: build a set of one-versus-one classifiers, and choose the class that is selected by the most classifiers. While this involves building $|\mathbb{C}|(|\mathbb{C}| - 1)/2$ classifiers, the time for training classifiers may actually decrease, since the training data set for each classifier is much smaller.

Multiclass support vector machines

Better alternative: structural SVMs

Multiclass support vector machines

Better alternative: structural SVMs

- 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

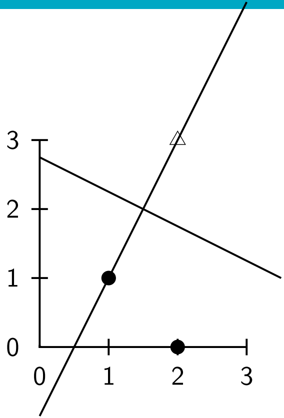
Multiclass support vector machines

Better alternative: structural SVMs

- 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

Walkthrough example: building an SVM over the data set shown in the figure

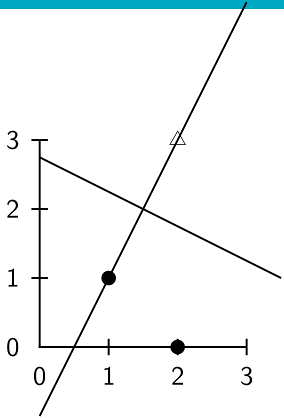
Working geometrically:



Walkthrough example: building an SVM over the data set shown in the figure

Working geometrically:

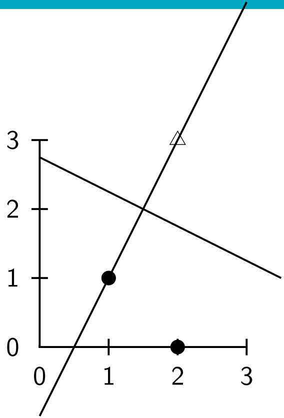
- The maximum margin weight vector will be parallel to the shortest line connecting points of the two classes, that is, the line between $(1,1)$ and $(2,3)$, giving a weight vector of $(1,2)$.



Walkthrough example: building an SVM over the data set shown in the figure

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 $(1.5, 2)$.

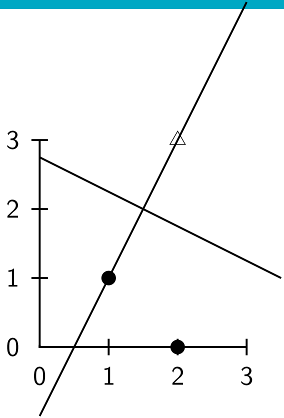


Walkthrough example: building an SVM over the data set shown in the figure

Working geometrically:

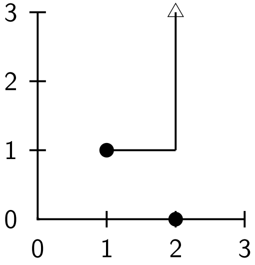
- The maximum margin weight vector will be parallel to the shortest line connecting points of the two classes, that is, the line between $(1, 1)$ and $(2, 3)$, giving a weight vector of $(1, 2)$.
- The optimal decision surface is orthogonal to that line and intersects it at the halfway point. Therefore,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

$$0 = \frac{1}{2}x + y - \frac{11}{4} \Leftrightarrow 0 = \frac{2}{5}x + \frac{4}{5}y - \frac{11}{5}$$



Walkthrough example: building an SVM over the data set shown in the figure

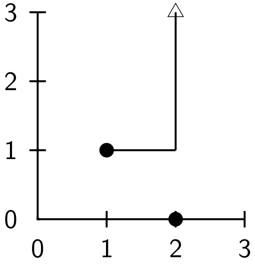
Working algebraically:



Walkthrough example: building an SVM over the data set shown in the figure

Working algebraically:

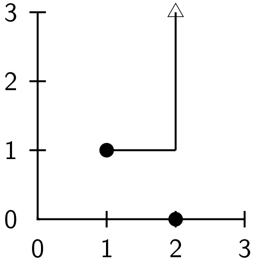
- With the constraint $\text{sign}(y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b)) \geq 1$, we seek to minimize $|\vec{w}|$.



Walkthrough example: building an SVM over the data set shown in the figure

Working algebraically:

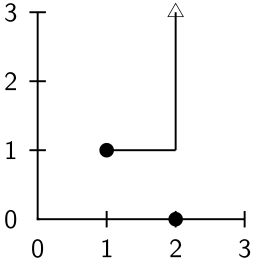
- With the constraint $\text{sign}(y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b)) \geq 1$, we seek to minimize $|\vec{w}|$.
- We know that the solution is $\vec{w} = (a, 2a)$ for some a. So: $a + 2a + b = -1$, $2a + 6a + b = 1$



Walkthrough example: building an SVM over the data set shown in the figure

Working algebraically:

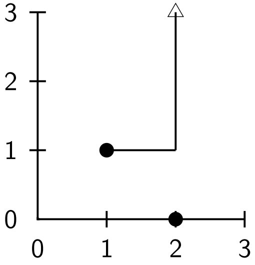
- With the constraint $\text{sign}(y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b)) \geq 1$, we seek to minimize $|\vec{w}|$.
- We know that the solution is $\vec{w} = (a, 2a)$ for some a. So: $a + 2a + b = -1$, $2a + 6a + b = 1$
- Hence, $a = 2/5$ and $b = -11/5$. So the optimal hyperplane is given by $\vec{w}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



Walkthrough example: building an SVM over the data set shown in the figure

Working algebraically:

- With the constraint $\mathrm{sign}(y_i(\vec{w}^\mathsf{T}\vec{x}_i + b)) \geq 1$, 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 $2/|\vec{w}| = 2/\sqrt{4/25 + 16/25} = 2/(2\sqrt{5}/5) = \sqrt{5} = \sqrt{(1-2)^2 + (1-3)^2}$.



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

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

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
$c = \text{grain}$

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)

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

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

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

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

- 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.

Take-away today

- Vector space text classification: Classification defined as the problem of finding a separating hyperplane in high-dimensional space
- Linear vector space classifiers: Simple, but often very effective
- Support vector machines: State-of-the-art text classification methods (linear and nonlinear)
- Discussion: Which classifier should I use for my problem?

Resources

- Chapter 14 of IIR (basic vector space classification)
- Chapter 15 of IIR (SVMs)
- Resources at `http://ifnlp.org/ir`