Introduction to Information Retrieval

http://informationretrieval.org

IIR 15: Brief Introduction to Support Vector Machines

Mihai Surdeanu

(Based on slides by Hinrich Schütze at informationretrieval.org)

Spring 2017

Overview

Take-away today

- Support vector machines: State-of-the-art text classification methods (linear and nonlinear)
- Introduction to SVMs
- Formalization (not covered!)
- Discussion: Which classifier should I use for my problem?

Overview

1. SVM intro
2. SVM details (not covered!)
3. Classification in the real world

Outline

Support vector machines

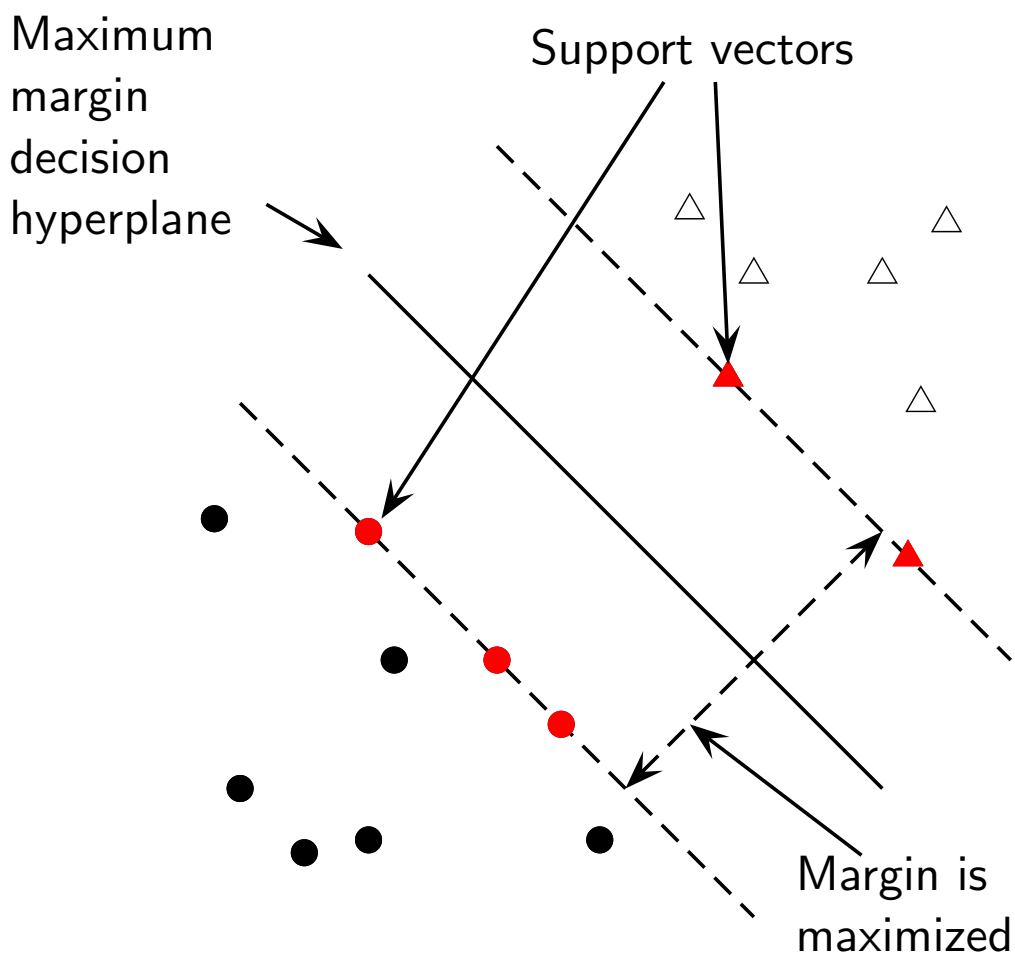
- Machine-learning research in the last two decades has improved classifier effectiveness.
- New generation of state-of-the-art classifiers: support vector machines (SVMs), boosted decision trees, regularized logistic regression, maximum entropy, neural networks, and random forests
- As we saw in IIR: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

What is a support vector machine – first take

- 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.

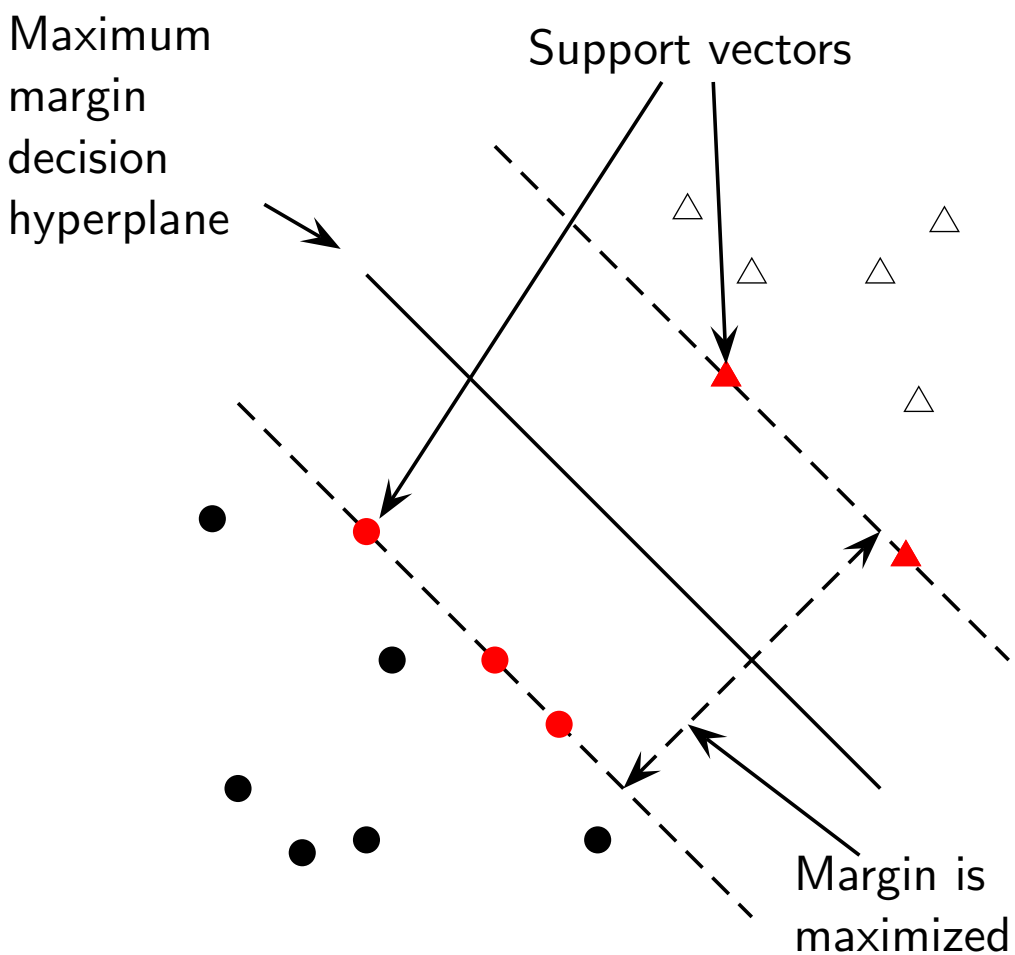
(Linear) Support Vector Machines

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

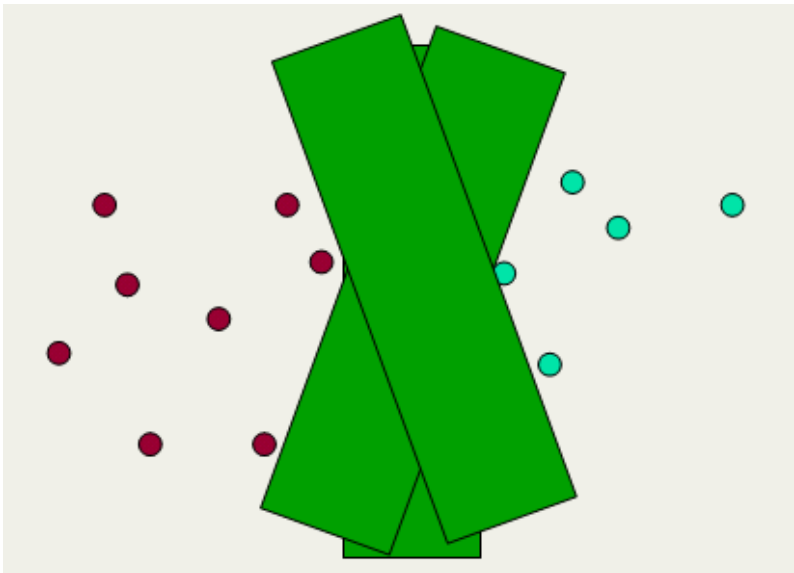


Why maximize the margin?

Points near the decision surface are uncertain classification decisions.
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
- We can think of the margin as a "fat separator" – a fatter version of our regular decision hyperplane.
- 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

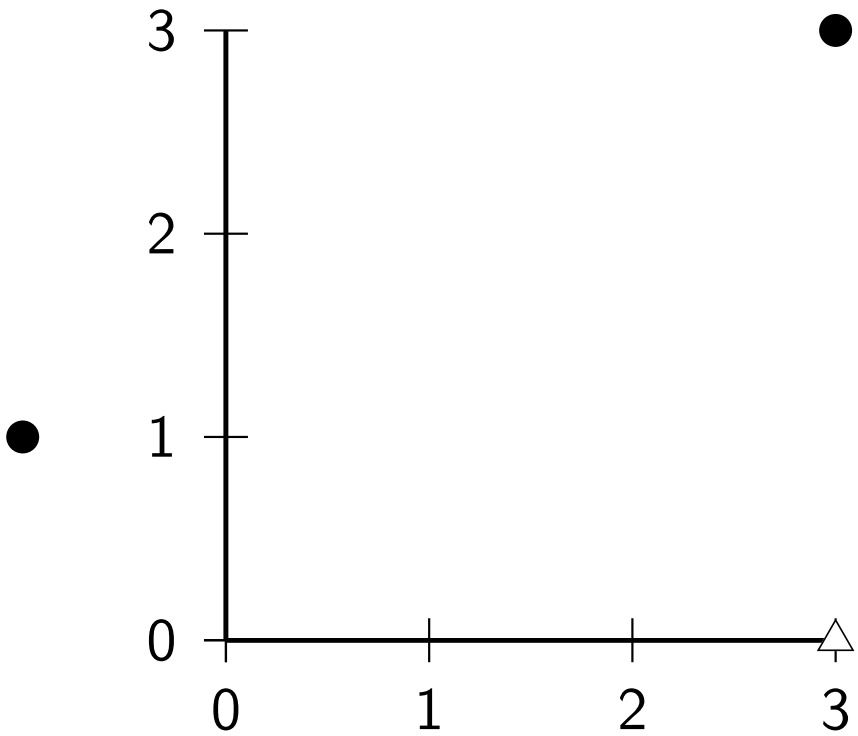
Can be defined by:

- intercept term b (we were calling this θ before)
- normal vector $\vec{w}$ (weight vector)

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

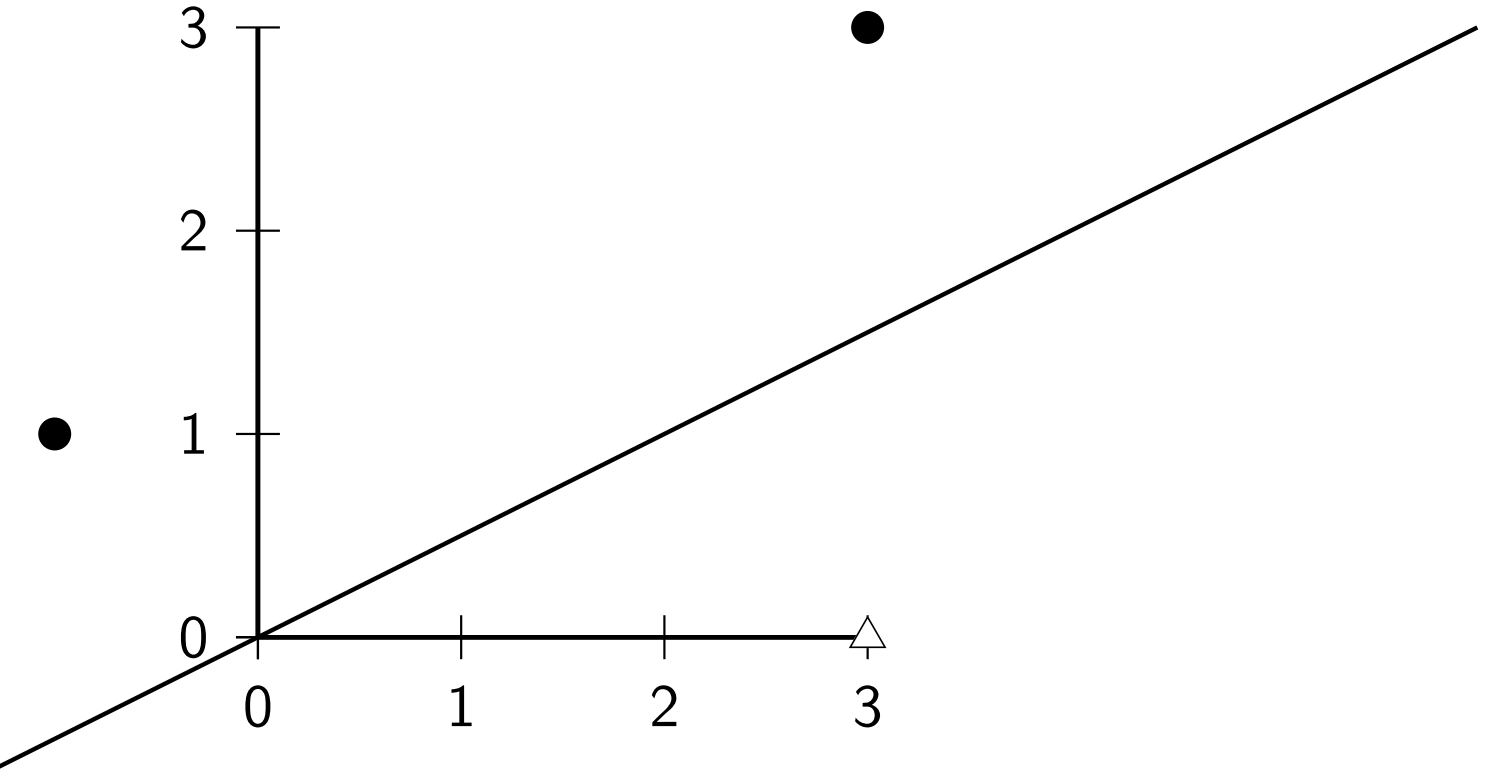
$$\vec{w}^{\mathsf{T}}\vec{x} + b = 0$$

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

Outline

Using SVM for multi-class classification

- We can use binary linear classifiers (k classes) for multi-class classification: 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.

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?

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!

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
top-level topic
topic definition modifiers
subtopic topic
evidence topic
topic definition modifier
evidence topic
topic definition modifier
evidence topic
topic definition modifier
subtopic

```
# Beginning of art topic definition
art ACCRUE
    /author = "fsmith"
    /date   = "30-Dec-01"
    /annotation = "Topic created
                   by fsmith"
* 0.70 performing-arts ACCRUE
** 0.50 WORD
    /wordtext = ballet
** 0.50 STEM
    /wordtext = dance
** 0.50 WORD
    /wordtext = opera
** 0.30 WORD
    /wordtext = symphony
* 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
** 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
```

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. Consider a hybrid approach!

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.

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.