PV211: Introduction to Information Retrieval

https://www.fi.muni.cz/~sojka/PV211

IIR 15-1: Support Vector Machines
Handout version

Petr Sojka, Hinrich Schütze et al.

Faculty of Informatics, Masaryk University, Brno
Center for Information and Language Processing, University of Munich

2024-04-24

(compiled on 2024-06-10 08:54:14)

Overview

1. SVM intro
2. SVM details
3. Classification in the real world

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?

Support vector machines

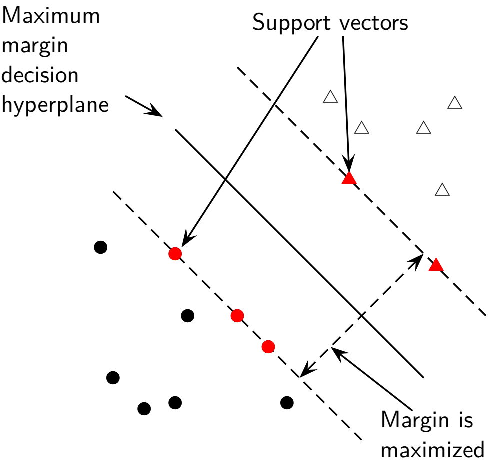
- Machine-learning research in the last two decades has improved classifier effectiveness.
- New generation of state-of-the-art classifiers: support vector machines (SVMs), boosted decision trees, regularized logistic regression, maximum entropy, neural networks, and random forests
- As we saw in IIR: 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 nonlinear separability: We may have to discount some points as outliers or noise.

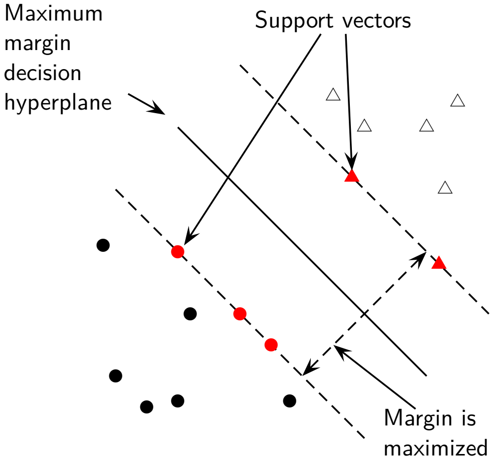
(Linear) Support Vector Machines

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

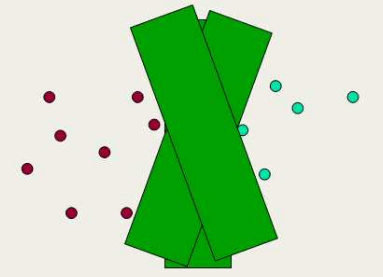


Why maximize the margin?

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

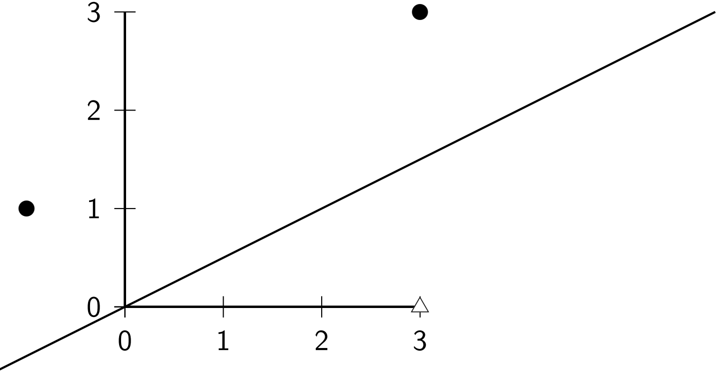
Can be defined by:

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

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)

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

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.
$\rightarrow$ We need to place some constraint on the size of $\vec{w}$.

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)

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

Assume canonical "functional margin" distance

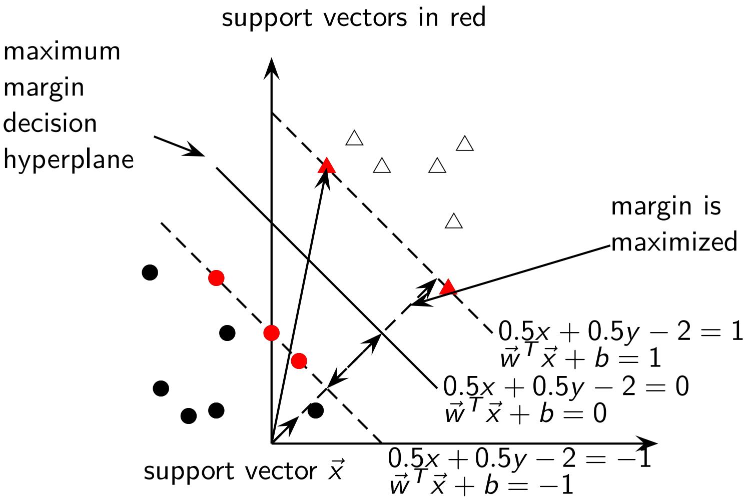
Assume that every data point has at least distance 1 from the hyperplane, then:

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

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

Support Vector Machines



support vectors in red

maximum margin decision hyperplane

margin is maximized

$0.5x + 0.5y - 2 = 1$
$\vec{w}^T\vec{x} + b = 1$

$0.5x + 0.5y - 2 = 0$
$\vec{w}^T\vec{x} + b = 0$

$0.5x + 0.5y - 2 = -1$
$\vec{w}^T\vec{x} + b = -1$

support vector $\vec{x}$

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

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

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

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 $\vec{w}'$)

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

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

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

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

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

$$\sqrt{2}$$

Optimization problem solved by SVMs (2)

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:

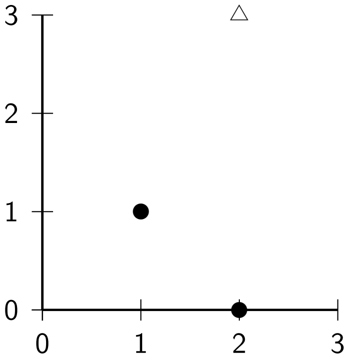
- $\frac{1}{2}\vec{w}^{\mathsf{T}}\vec{w}$ is minimized (because $|\vec{w}| = \sqrt{\vec{w}^{\mathrm{T}}\vec{w}}$), and
- for all $\{(\vec{x}_i, y_i)\}$, $y_i(\vec{w}^{\mathsf{T}}\vec{x}_i + b) \geq 1$

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 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



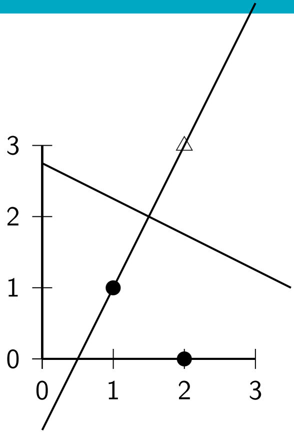
Which vectors are the support vectors? Draw the maximum margin separator. What values of w_1, w_2 and b (for $w_1x + w_2y + b = 0$) describe this separator? Recall that we must have $w_1x + w_2y + b \in \{1, -1\}$ for the support vectors.

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:

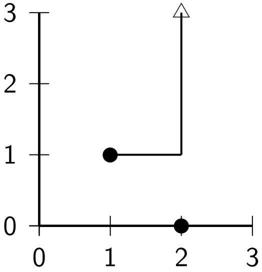
$$0 = x+2y-(1\cdot 1.5+2\cdot 2) \Leftrightarrow 0 = \frac{2}{5}x+\frac{4}{5}y-\frac{11}{5}$$



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

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

- 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

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

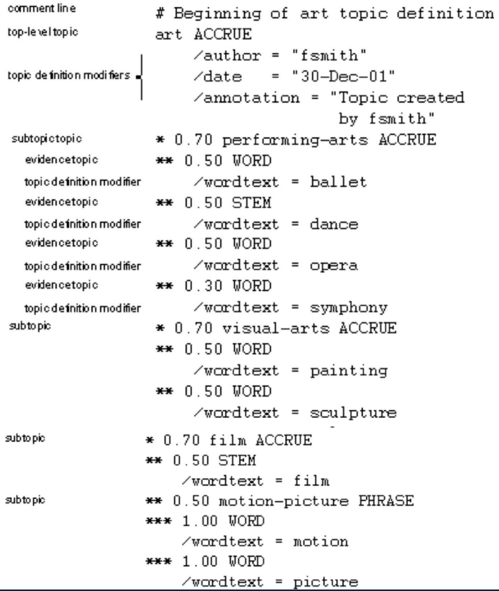
Use hand-written rules!

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               # Beginning of art topic definition
top-level topic            art ACCRUE
                               /author = "fsmith"
topic definition modifiers     /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 PHRASE
                           *** 1.00 WORD
                               /wordtext = motion
                           *** 1.00 WORD
                               /wordtext = picture
```

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.

- 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 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.

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 trade-off 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

- 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 15-1 of IIR
- Resources at `https://www.fi.muni.cz/~sojka/PV211/` and `http://cislmu.org`, materials in MU IS and FI MU library
- Discussion of "how to select the right classifier for my problem" in Russell and Norvig