 CS276B
Text Information Retrieval, Mining, and Exploitation

Lecture 9
Text Classification IV
Feb 13, 2003
Today’s Topics

- More algorithms
  - Vector space classification
  - Nearest neighbor classification
  - Support vector machines
- Hypertext classification
Vector Space Classification
K Nearest Neighbor Classification
Recall Vector Space Representation

- Each document is a vector, one component for each term (= word).
- Normalize to unit length.
- Properties of vector space
  - terms are axes
  - \( n \) docs live in this space
  - even with stemming, may have 10,000+ dimensions, or even 1,000,000+
Classification Using Vector Spaces

- Each training doc a point (vector) labeled by its class
- Similarity hypothesis: docs of the same class form a contiguous region of space. Or: Similar documents are usually in the same class.
- Define surfaces to delineate classes in space
Classes in a Vector Space

Similarity hypothesis true in general?

- Government
- Science
- Arts
Given a Test Document

- Figure out which region it lies in
- Assign corresponding class
Test Document = Government
Binary Classification

- Consider 2 class problems
- How do we define (and find) the separating surface?
- How do we test which region a test doc is in?
Separation by Hyperplanes

- Assume *linear separability* for now:
  - in 2 dimensions, can separate by a line
  - in higher dimensions, need hyperplanes
- Can find separating hyperplane by *linear programming* (e.g. perceptron):
  - separator can be expressed as $ax + by = c$
Linear Programming / Perceptron

Find $a, b, c$, such that

$ax + by \geq c$ for red points

$ax + by \leq c$ for green points.
Relationship to Naïve Bayes?

Find $a, b, c$, such that
$ax + by \geq c$ for red points
$ax + by \leq c$ for green points.
Linear Classifiers

- Many common text classifiers are linear classifiers
- Despite this similarity, large performance differences
  - For separable problems, there is an infinite number of separating hyperplanes. Which one do you choose?
  - What to do for non-separable problems?
Which Hyperplane?

In general, lots of possible solutions for $a, b, c$. 
Which Hyperplane?

- Lots of possible solutions for \(a,b,c\).
- Some methods find a separating hyperplane, but not the optimal one (e.g., perceptron)
- Most methods find an optimal separating hyperplane
- Which points should influence optimality?
  - All points
    - Linear regression
    - Naïve Bayes
  - Only “difficult points” close to decision boundary
    - Support vector machines
    - Logistic regression (kind of)
Hyperplane: Example

- Class: “interest” (as in interest rate)
- Example features of a linear classifier (SVM)

\[
\begin{array}{cc}
w_i & t_i \\
0.70 \text{ prime} & -0.71 \text{ dlr} \\
0.67 \text{ rate} & -0.35 \text{ world} \\
0.63 \text{ interest} & -0.33 \text{ sees} \\
0.60 \text{ rates} & -0.25 \text{ year} \\
0.46 \text{ discount} & -0.24 \text{ group} \\
0.43 \text{ bundesbank} & -0.24 \text{ dlr}
\end{array}
\]
More Than Two Classes

- One-of classification: each document belongs to exactly one class
  - How do we compose separating surfaces into regions?
- Any-of or multiclass classification
  - For n classes, decompose into n binary problems
- Vector space classifiers for one-of classification
  - Use a set of binary classifiers
  - Centroid classification
  - K nearest neighbor classification
Composing Surfaces: Issues
Set of Binary Classifiers

- Build a separator between each class and its complementary set (docs from all other classes).
- Given test doc, evaluate it for membership in each class.
- For one-of classification, declare membership in classes for class with:
  - maximum score
  - maximum confidence
  - maximum probability
- Why different from multiclass classification?
Negative Examples

- Formulate as above, except negative examples for a class are added to its complementary set.

○ Positive examples
□ Negative examples
Centroid Classification

- Given training docs for a class, compute their centroid
- Now have a centroid for each class
- Given query doc, assign to class whose centroid is nearest.
- Compare to Rocchio
k Nearest Neighbor Classification

- To classify document d into class c
- Define k-neighborhood N as k nearest neighbors of d
- Count number of documents l in N that belong to c
- Estimate P(c|d) as l/k
Example: $k=6$ (6NN)

$P(\text{science}|\diamond)$?

- Government
- Science
- Arts
Cover and Hart 1967

- Asymptotically, the error rate of 1-nearest-neighbor classification is less than twice the Bayes rate.
- Assume: query point coincides with a training point.
- Both query point and training point contribute error $\rightarrow$ 2 times Bayes rate
- In particular, asymptotic error rate 0 if Bayes rate is 0.
kNN vs. Regression

- Bias/Variance tradeoff
- Variance $\approx$ Capacity
- kNN has high variance and low bias.
- Regression has low variance and high bias.
- Consider: Is an object a tree? (Burges)
- Too much capacity/variance, low bias
  - Botanist who memorizes
  - Will always say “no” to new object (e.g., #leaves)
- Not enough capacity/variance, high bias
  - Lazy botanist
  - Says “yes” if the object is green
kNN: Discussion

- Classification time linear in training set
- No feature selection necessary
- Scales well with large number of classes
  - Don’t need to train n classifiers for n classes
- Classes can influence each other
  - Small changes to one class can have ripple effect
- Scores can be hard to convert to probabilities
- No training necessary
  - Actually: not true. Why?
Number of Neighbors
Hypertext Classification
Classifying Hypertext

- Given a set of hyperlinked docs
- Class labels for some docs available
- Figure out class labels for remaining docs
Example
Bayesian Hypertext Classification

- Besides the terms in a doc, derive cues from linked docs to assign a class to test doc.
- Cues could be any abstract features from doc and its neighbors.
Feature Representation

- Attempt 1:
  - use terms in doc + those in its neighbors.
  - Generally does worse than terms in doc alone. Why?
Representation: Attempt 2

- Use terms in doc, plus tagged terms from neighbors.
- E.g.,
  - \textit{car} denotes a term occurring in \textit{d}.
  - \textit{car@l} denotes a term occurring in a doc with a link into \textit{d}.
  - \textit{car@O} denotes a term occurring in a doc with a link from \textit{d}.
- Generalizations possible: \textit{car@OIOI}
Attempt 2 Also Fails

- Key terms lose density
- e.g., *car* gets split into *car*, *car@l*, *car@O*
Better Attempt

- Use class labels of (in- and out-) neighbors as features in classifying $d$.
  - e.g., docs about physics point to docs about physics.
- Setting: some neighbors have pre-assigned labels; need to figure out the rest.
Example
Content + Neighbors’ Classes

- Naïve Bayes gives $\Pr[c_j|d]$ based on the words in $d$.
- Now consider $\Pr[c_j|N]$ where $N$ is the set of labels of $d$’s neighbors.
  (Can separate $N$ into in- and out-neighbors.)
- Can combine conditional probs for $c_j$ from text- and link-based evidence.
Training

- As before, use training data to compute $\Pr[N|c_j]$ etc.
- Assume labels of $d$’s neighbors independent (as we did with word occurrences).
- (Also continue to assume word occurrences within $d$ are independent.)
Classification

- Can invert probs using Bayes to derive $\Pr[c_j|N]$.
- Need to know class labels for all of $d$’s neighbors.
Unknown Neighbor Labels

- What if all neighbors’ class labels are not known?
- First, use word content alone to assign a tentative class label to each unlabelled doc.
- Next, iteratively recompute all tentative labels using word content as well as neighbors’ classes (some tentative).
Convergence

- This iterative relabeling will converge provided tentative labels “not too far off”.
- Guarantee requires ideas from Markov random fields, used in computer vision.
- Error rates significantly below text-alone classification.
Typical Empirical Observations

- Training ~ 100’s to 1000+ docs/class
- Accuracy
  ~ 90% in the very best circumstances
  below 50% in the worst
Support Vector Machines
Recall: Which Hyperplane?

- In general, lots of possible solutions for $a, b, c$.
- Support Vector Machine (SVM) finds an optimal...
Support Vector Machine (SVM)

- SVMs maximize the *margin* around the separating hyperplane.
- The decision function is fully specified by a subset of training samples, the *support vectors*.
- *Quadratic programming* problem
- Text classification method du jour
Maximum Margin: Formalization

- $w$: hyperplane normal
- $x_i$: data point $i$
- $y_i$: class of data point $i$ (+1 or -1)

Constraint optimization formalization:

1. $x_i \cdot w + b \geq +1$ for $y_i = +1$
2. $x_i \cdot w + b \leq -1$ for $y_i = -1$

(2) maximize margin: $2/||w||$
Quadratic Programming

- One can show that hyperplane w with maximum margin is:

\[ w = \sum \alpha_i y_i x_i \]

- \( \alpha_i \): lagrange multipliers
- \( x_i \): data point i
- \( y_i \): class of data point i (+1 or -1)

Where the \( \alpha_i \) are the solution to maximizing:

\[ L_D = \sum \alpha_i - \frac{1}{2} \sum \alpha_i \alpha_j y_i y_j x_i \cdot x_j \]

Most \( \alpha_i \) will be zero.
Building an SVM Classifier

- Now we know how to build a separator for two linearly separable classes
- What about classes whose exemplary docs are not linearly separable?
Not Linearly Separable

Find a line that penalizes points on "the wrong side"
Penalizing Bad Points

Define distance for each point \( v \) respect to separator \( ax + by = c \)

- \((ax + by) - c\) for red points
- \(c - (ax + by)\) for green points.
Solve Quadratic Program

- Solution gives “separator” between two classes: choice of $a, b$.
- Given a new point $(x, y)$, can score its proximity to each class:
  - evaluate $ax + by$.
  - Set confidence threshold.
Predicting Generalization

- We want the classifier with the best generalization (best accuracy on new data).
- What are clues for good generalization?
  - Large training set
  - Low error on training set
  - Low capacity/variance (≈ model with few parameters)
- SVMs give you an explicit bound based on these.
Capacity/Variance: VC Dimension

- Theoretical risk boundary:
  \[ R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\left( \frac{h(\log(2l/h) + 1) - \log(\eta/4)}{l} \right)} \]

- \( R_{emp} \) - empirical risk, \( l \) - #observations, \( h \) - VC dimension, the above holds with prob. \((1-\eta)\)

- VC dimension/Capacity: max number of points that can be shattered

- A set can be shattered if the classifier can learn every possible labeling.
Capacity of Hyperplanes?
Exercise

- Suppose you have $n$ points in $d$ dimensions, labeled red or green. How big need $n$ be (as a function of $d$) in order to create an example with the red and green points not linearly separable?
- E.g., for $d=2$, $n \geq 4$. 

![Diagram of points in two dimensions]
Capacity/Variance: VC Dimension

- Theoretical risk boundary:
  \[ R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\left(\frac{h(\log(2l/h) + 1) - \log(\eta/4)}{l}\right)} \]

- \( R_{emp} \) - empirical risk, \( l \) - #observations, \( h \) - VC dimension, the above holds with prob. \((1-\eta)\)

- VC dimension/Capacity: max number of points that can be shattered

- A set can be shattered if the classifier can learn every possible labeling.
Kernels

- Recall: We’re maximizing:

\[
L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j
\]

- Observation: data only occur in dot products.
- We can map data into a very high dimensional space (even infinite!) as long as kernel computable.
- For mapping function \( \Phi \), compute kernel \( K(i,j) = \Phi(x_i) \cdot \Phi(x_j) \)
- Example:

\[
K(x_i, x_j) = (x_i \cdot x_j)^2 \quad \Phi(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2} \ x_1 x_2 \\ x_2^2 \end{pmatrix}
\]
Kernels

- Why use kernels?
Kernels

- Why use kernels?
  - Make non-separable problem separable.
  - Map data into better representational space

- Common kernels
  - Linear
  - Polynomial
  - Radial basis function

\[ K(x_i, x_j) = e^{-\|x_i - x_j\|^2/2\sigma^2} \]
Performance of SVM

- SVM are seen as best-performing method by many.
- Statistical significance of most results not clear.
- There are many methods that perform about as well as SVM.
- Example: regularized regression (Zhang&Oles)
- Example of a comparison study: Yang&Liu
Yang & Liu: SVM vs Other Methods

Table 1: Performance summary of classifiers

<table>
<thead>
<tr>
<th>method</th>
<th>miR</th>
<th>miP</th>
<th>miF1</th>
<th>maF1</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>.8120</td>
<td>.9137</td>
<td>.8599</td>
<td>.5251</td>
<td>.00365</td>
</tr>
<tr>
<td>KNN</td>
<td>.8339</td>
<td>.8807</td>
<td>.8567</td>
<td>.5242</td>
<td>.00385</td>
</tr>
<tr>
<td>LSF</td>
<td>.8507</td>
<td>.8489</td>
<td>.8498</td>
<td>.5008</td>
<td>.00414</td>
</tr>
<tr>
<td>NNet</td>
<td>.7842</td>
<td>.8785</td>
<td>.8287</td>
<td>.3765</td>
<td>.00447</td>
</tr>
<tr>
<td>NB</td>
<td>.7688</td>
<td>.8245</td>
<td>.7956</td>
<td>.3886</td>
<td>.00544</td>
</tr>
</tbody>
</table>

miR = micro-avg recall;  
miP = micro-avg prec.;  
miF1 = micro-avg F1;  
maF1 = macro-avg F1.
### Table 2: Statistical significance test results

<table>
<thead>
<tr>
<th>sysA</th>
<th>sysB</th>
<th>s-test</th>
<th>S-test</th>
<th>T-test</th>
<th>T''-test</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>kNN</td>
<td>&gt;&gt;</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>SVM</td>
<td>LLSF</td>
<td>&gt;&gt;</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>kNN</td>
<td>LLSF</td>
<td>&gt;&gt;</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>SVM</td>
<td>NNet</td>
<td>&gt;&gt;</td>
<td>&gt;</td>
<td>&gt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>kNN</td>
<td>NNet</td>
<td>&gt;&gt;</td>
<td>&gt;</td>
<td>&gt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>LLSF</td>
<td>NNet</td>
<td>2</td>
<td>&gt;</td>
<td>&gt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>NB</td>
<td>kNN</td>
<td>&lt;&lt;</td>
<td>&gt;</td>
<td>&gt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>NB</td>
<td>LLSF</td>
<td>&lt;&lt;</td>
<td>&gt;</td>
<td>&gt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>NB</td>
<td>SVM</td>
<td>&lt;&lt;</td>
<td>&gt;</td>
<td>&gt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>NB</td>
<td>NNet</td>
<td>&lt;&lt;</td>
<td>&gt;</td>
<td>&gt;</td>
<td>&gt;</td>
</tr>
</tbody>
</table>

“>>” or “<<” means P-value ≤ 0.01;
“>” or “<” means 0.01 < P-value ≤ 0.05;
“~” means P-value > 0.05.
Yang & Liu: Small Classes
### Results for Kernels (Joachims)

<table>
<thead>
<tr>
<th></th>
<th>Bayes</th>
<th>Rocchio</th>
<th>C4.5</th>
<th>k-NN</th>
<th>SVM (poly) degree $d =$ 1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>SVM (rbf) width $\gamma =$ 0.6</th>
<th>0.8</th>
<th>1.0</th>
<th>1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>earn</td>
<td>95.9</td>
<td>96.1</td>
<td>96.1</td>
<td>97.3</td>
<td>98.2</td>
<td>98.4</td>
<td>98.5</td>
<td>98.4</td>
<td>98.3</td>
<td>98.5</td>
<td>98.5</td>
<td>98.4</td>
<td>98.3</td>
</tr>
<tr>
<td>acq</td>
<td>91.5</td>
<td>92.1</td>
<td>85.3</td>
<td>92.0</td>
<td>92.6</td>
<td>94.6</td>
<td>95.2</td>
<td>95.2</td>
<td>95.3</td>
<td>95.0</td>
<td>95.3</td>
<td>95.3</td>
<td>95.4</td>
</tr>
<tr>
<td>money-fx</td>
<td>62.9</td>
<td>67.6</td>
<td>69.4</td>
<td>78.2</td>
<td>66.9</td>
<td>72.5</td>
<td>75.4</td>
<td>74.9</td>
<td>76.2</td>
<td>74.0</td>
<td>75.4</td>
<td>76.3</td>
<td>75.9</td>
</tr>
<tr>
<td>grain</td>
<td>72.5</td>
<td>79.5</td>
<td>89.1</td>
<td>82.2</td>
<td>91.3</td>
<td>93.1</td>
<td>92.4</td>
<td>91.3</td>
<td>89.9</td>
<td>93.1</td>
<td>91.9</td>
<td>91.9</td>
<td>90.6</td>
</tr>
<tr>
<td>crude</td>
<td>81.0</td>
<td>81.5</td>
<td>75.5</td>
<td>85.7</td>
<td>86.0</td>
<td>87.3</td>
<td>88.6</td>
<td>88.9</td>
<td>87.8</td>
<td>88.9</td>
<td>89.0</td>
<td>88.9</td>
<td>88.2</td>
</tr>
<tr>
<td>trade</td>
<td>50.0</td>
<td>77.4</td>
<td>59.2</td>
<td>77.4</td>
<td>69.2</td>
<td>75.5</td>
<td>76.6</td>
<td>77.3</td>
<td>77.1</td>
<td>76.9</td>
<td>78.0</td>
<td>77.8</td>
<td>76.8</td>
</tr>
<tr>
<td>interest</td>
<td>58.0</td>
<td>72.5</td>
<td>49.1</td>
<td>74.0</td>
<td>69.8</td>
<td>63.3</td>
<td>67.9</td>
<td>73.1</td>
<td>76.2</td>
<td>74.4</td>
<td>75.0</td>
<td>76.2</td>
<td>76.1</td>
</tr>
<tr>
<td>ship</td>
<td>78.7</td>
<td>83.1</td>
<td>80.9</td>
<td>79.2</td>
<td>82.0</td>
<td>85.4</td>
<td>86.0</td>
<td>86.5</td>
<td>86.0</td>
<td>85.4</td>
<td>86.5</td>
<td>87.6</td>
<td>87.1</td>
</tr>
<tr>
<td>wheat</td>
<td>60.6</td>
<td>79.4</td>
<td>85.5</td>
<td>76.6</td>
<td>83.1</td>
<td>84.5</td>
<td>85.2</td>
<td>85.9</td>
<td>83.8</td>
<td>85.2</td>
<td>85.9</td>
<td>85.9</td>
<td>85.9</td>
</tr>
<tr>
<td>corn</td>
<td>47.3</td>
<td>62.2</td>
<td>87.7</td>
<td>77.9</td>
<td>86.0</td>
<td>86.5</td>
<td>85.3</td>
<td>85.7</td>
<td>83.9</td>
<td>85.1</td>
<td>85.7</td>
<td>85.7</td>
<td>84.5</td>
</tr>
<tr>
<td>microavg.</td>
<td>72.0</td>
<td>79.9</td>
<td>79.4</td>
<td>82.3</td>
<td>84.2</td>
<td>85.1</td>
<td>85.9</td>
<td>86.2</td>
<td>85.9</td>
<td>86.4</td>
<td>86.5</td>
<td>86.3</td>
<td>86.2</td>
</tr>
</tbody>
</table>

Combined: 86.0

Combined: 86.4
SVM: Summary

- SVM have optimal or close to optimal performance.
- Kernels are an elegant and efficient way to map data into a better representation.
- SVM can be expensive to train (quadratic programming).
- If efficient training is important, and slightly suboptimal performance ok, don’t use SVM?
- For text, linear kernel is common.
- So most SVMs are linear classifiers (like many others), but find a (close to) optimal separating hyperplane.
SVM: Summary (cont.)

- Model parameters based on small subset (SVs)
- Based on structural risk minimization

\[ R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\left( \frac{h(\log(2l/h) + 1) - \log(\eta/4)}{l} \right) } \]

\[ L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \]
Resources

- Trevor Hastie, Robert Tibshirani and Jerome Friedman, "Elements of Statistical Learning: Data Mining, Inference and Prediction" Springer-Verlag, New York.
- Data Mining and Knowledge Discovery
- S. T. Dumais, Using SVMs for text categorization, IEEE Intelligent Systems, 13(4), Jul/Aug 1998
- re-examination of text categorization methods (1999) Yiming Yang, Xin Liu 22nd Annual International SIGIR