Introduction to Information Retrieval

`http://informationretrieval.org`

IIR 14: Vector Space Classification

Mihai Surdeanu

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

Spring 2017

Overview

1. Intro vector space classification
2. Rocchio
3. kNN

Take-away today

- Vector space classification: Basic idea of doing text classification for documents that are represented as vectors
- Rocchio classifier: Rocchio relevance feedback idea applied to text classification (briefly covered)
- *k* nearest neighbor classification
- Linear classifiers (not covered)

Outline

1. Intro vector space classification
2. Rocchio
3. kNN

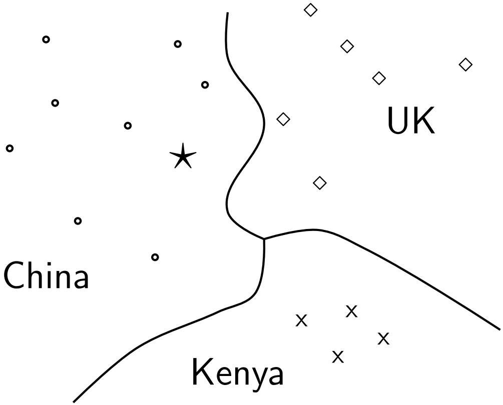
Recall vector space representation

- Each document is a vector, one component for each term.
- Terms are axes.
- High dimensionality: 100,000s of dimensions
- Normalize vectors (documents) to unit length
- How can we do classification in this 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.
- Premise 2: Documents from different classes don't overlap.
- We define lines, surfaces, hypersurfaces to divide regions.

Classes in the vector space



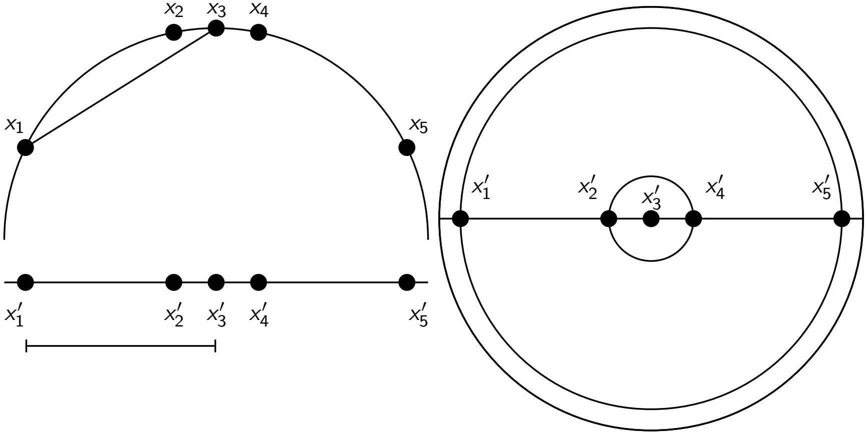
Should the document ⋆ be assigned to *China*, *UK* or *Kenya*?

Find separators between the classes

Based on these separators: ⋆ should be assigned to *China*

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

Aside: 2D/3D graphs can be misleading



Left: A projection of the 2D semicircle to 1D. For the points x_1, x_2, x_3, x_4, x_5 at x coordinates $-0.9, -0.2, 0, 0.2, 0.9$ the distance $|x_2 x_3| \approx 0.201$ only differs by 0.5% from $|x'_2 x'_3| = 0.2$; but $|x_1 x_3|/|x'_1 x'_3| = d_{\text{true}}/d_{\text{projected}} \approx 1.06/0.9 \approx 1.18$ is an example of a large distortion (18%) when projecting a large area. *Right:* The corresponding projection of the 3D hemisphere to 2D.

Outline

1. Intro vector space classification

2. Rocchio

3. kNN

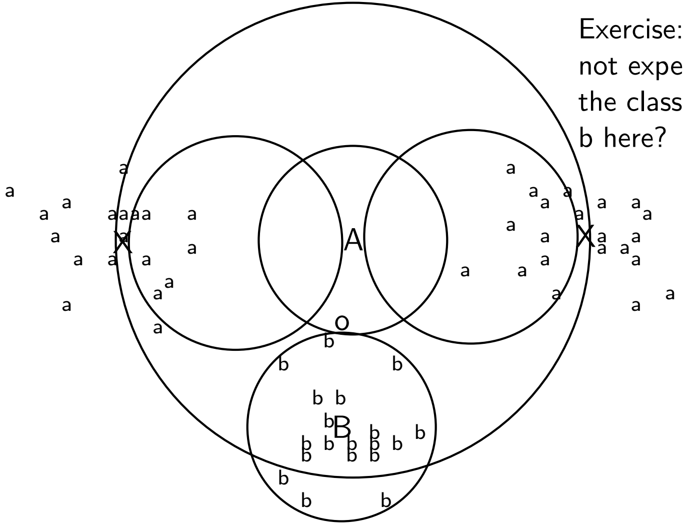
Rocchio classification: Basic idea

- Compute a centroid for each class
  - The centroid is the average of all documents in the class.
- Assign each test document to the class of its closest centroid.

Rocchio vs. Naive Bayes

- In many cases, Rocchio performs worse than Naive Bayes.
- One reason: Rocchio does not handle nonconvex, multimodal classes correctly.

Rocchio cannot handle nonconvex, multimodal classes



Exercise: Why is Rocchio not expected to do well for the classification task a vs. b here?

Outline

1 Intro vector space classification

2 Rocchio

3 kNN

kNN classification

- kNN classification is another vector space classification method.
- It also is very simple and easy to implement.
- kNN is more accurate (in most cases) than Naive Bayes and Rocchio.
- If you need to get a pretty accurate classifier up and running in a short time . . .
- . . . and you don't care about efficiency that much . . .
- . . . use kNN.

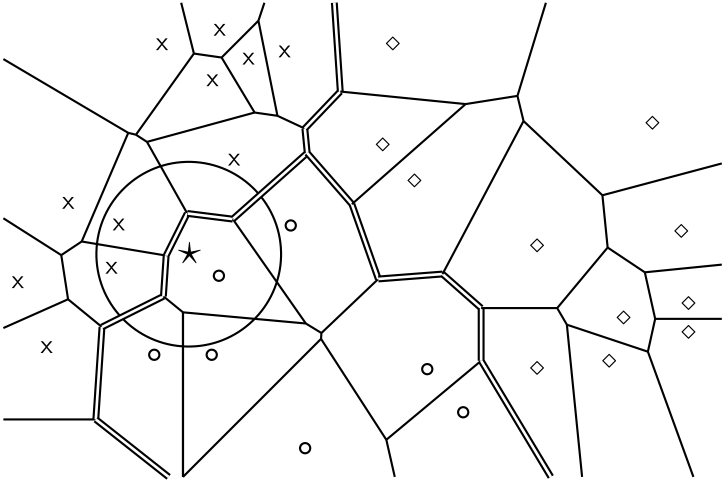
kNN classification

- kNN $=$ k nearest neighbors
- kNN classification rule for $k = 1$ (1NN): Assign each test document to the class of its nearest neighbor in the training set.
- 1NN is not very robust – one document can be mislabeled or atypical.
- kNN classification rule for $k > 1$ (kNN): Assign each test document to the majority class of its k nearest neighbors in the training set.
- Rationale of kNN: contiguity hypothesis
  - We expect a test document d to have the same label as the training documents located in the local region surrounding d.

Probabilistic kNN

- Probabilistic version of kNN: $P(c|d) =$ fraction of k neighbors of d that are in c
- kNN classification rule for probabilistic kNN: Assign d to class c with highest $P(c|d)$

kNN is based on Voronoi tessellation



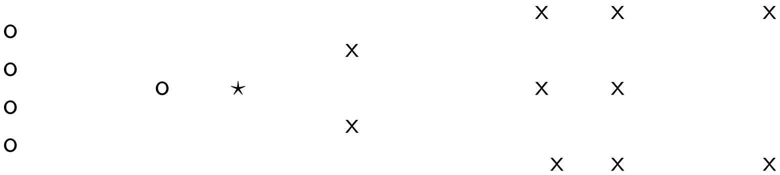
kNN is a non-linear classifier! That is, separators are not “lines”.

kNN algorithm

```
$\textsc{Train-kNN}(\mathbb{C}, \mathbb{D})$
$\mathbb{D}' \leftarrow \textsc{Preprocess}(\mathbb{D})$
$k \leftarrow \textsc{Select-k}(\mathbb{C}, \mathbb{D}')$//tuning
return $\mathbb{D}', k$

$\textsc{Apply-kNN}(\mathbb{D}', k, d)$
$S_k \leftarrow \textsc{ComputeNearestNeighbors}(\mathbb{D}', k, d)$
for each $c_j \in \mathbb{C}(\mathbb{D}')$
do $p_j \leftarrow |S_k \cap c_j|/k$
return $\arg\max_j p_j$
```

Exercise



How is star classified by:

(i) 1-NN (ii) 3-NN (iii) 9-NN (iv) 15-NN (v) Rocchio?

Software

- TiMBL: `http://ilk.uvt.nl/timbl/`
- Weka: `http://www.cs.waikato.ac.nz/ml/weka/` and `http://www.programcreek.com/2013/01/use-k-nearest-neighbors-knn-classifier-in-java/`

Time complexity of kNN

kNN with preprocessing of training set

training	$\Theta(\lvert\mathbb{D}\rvert L_{\mathrm{ave}})$
testing	$\Theta(L_{\mathrm{a}} + \lvert\mathbb{D}\rvert M_{\mathrm{ave}} M_{\mathrm{a}}) = \Theta(\lvert\mathbb{D}\rvert M_{\mathrm{ave}} M_{\mathrm{a}})$

- kNN test time proportional to the size of the training set!
- The larger the training set, the longer it takes to classify a test document.
- kNN is inefficient for very large training sets.

kNN: Discussion

- No training necessary
  - But linear preprocessing of documents is as expensive as training Naive Bayes.
  - We always preprocess the training set, so in reality training time of kNN is linear.
- kNN is very accurate if training set is large.
- But kNN can be very inaccurate if training set is small.

Take-away today

- Vector space classification: Basic idea of doing text classification for documents that are represented as vectors
- Rocchio classifier: Rocchio relevance feedback idea applied to text classification (briefly covered)
- *k* nearest neighbor classification
- Linear classifiers (not covered)