Information Storage and Retrieval

CSCE 670
Texas A&M University
Department of Computer Science & Engineering
Instructor: Prof. James Caverlee

Text Classification
21 February 2017

Today

- Text Classification: Definition and Overview
- Vector Space Classification
  - Rocchio
  - kNN

90° Earthquuuuuuuaaakesss!!!

the ONION
America's Finest News Source

Grandmother Classifies 79% Of Everything A Shame

NEWS · Family · Local · ISSUE 47·47 ISSUE 45·29 · Jul 18, 2009

SANDUSKY, OH—According to those close to Gertrude Wharton, the grandmother of nine declares 79 percent of everything she witnesses, experiences, or hears about from friends to be "a shame."

"No matter what happens, her response is always, 'That's a shame,'" said Wharton's son Kevin, 46. "From the recent passing of her friend Lillian to the fact that her coupon for chicken bouillon cubes expired last week, I can't have a conversation with her without being told something is a shame. Is this really how she

Standing queries

- The path from IR to text classification:
  - You have an information need to monitor, say:
    - Snoop Lion performances
  - You want to rerun an appropriate query periodically to find new news items on this topic
  - You will be sent new documents that are found
    - I.e., it's not ranking but classification (relevant vs. not relevant)
- Such queries are called **standing queries**
  - Long used by "information professionals"
  - A modern mass instantiation is **Google Alerts**
- Standing queries are (hand-written) text classifiers

http://www.google.com/alerts

A text classification task: Email spam filtering

```
From: ''' <takworlld@hotmail.com>
Subject: real estate is the only way... gem oalvgkay
Anyone can buy real estate with no money down
Stop paying rent TODAY !
There is no need to spend hundreds or even thousands for
similar courses
I am 22 years old and I have already purchased 6 properties
using the
methods outlined in this truly INCREDIBLE ebook.
Change your life NOW !
=================================================
Click Below to order:
http://www.wholesaledaily.com/sales/nmd.htm
=================================================
```

How would you write a program that would automatically detect and delete this type of message?

Formal definition of Text Classification: Training

Given:

- A document space X
  - Documents are represented in this space – typically some type of high-dimensional space.
- A fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
  - The classes are human-defined for the needs of an application (e.g., relevant vs. nonrelevant).
- A training set D of labeled documents with each labeled document $\langle d, c \rangle \in X \times C$

Using a learning method or learning algorithm, we then wish to learn a classifier Υ that maps documents to classes:

$$\Upsilon : X \rightarrow C$$

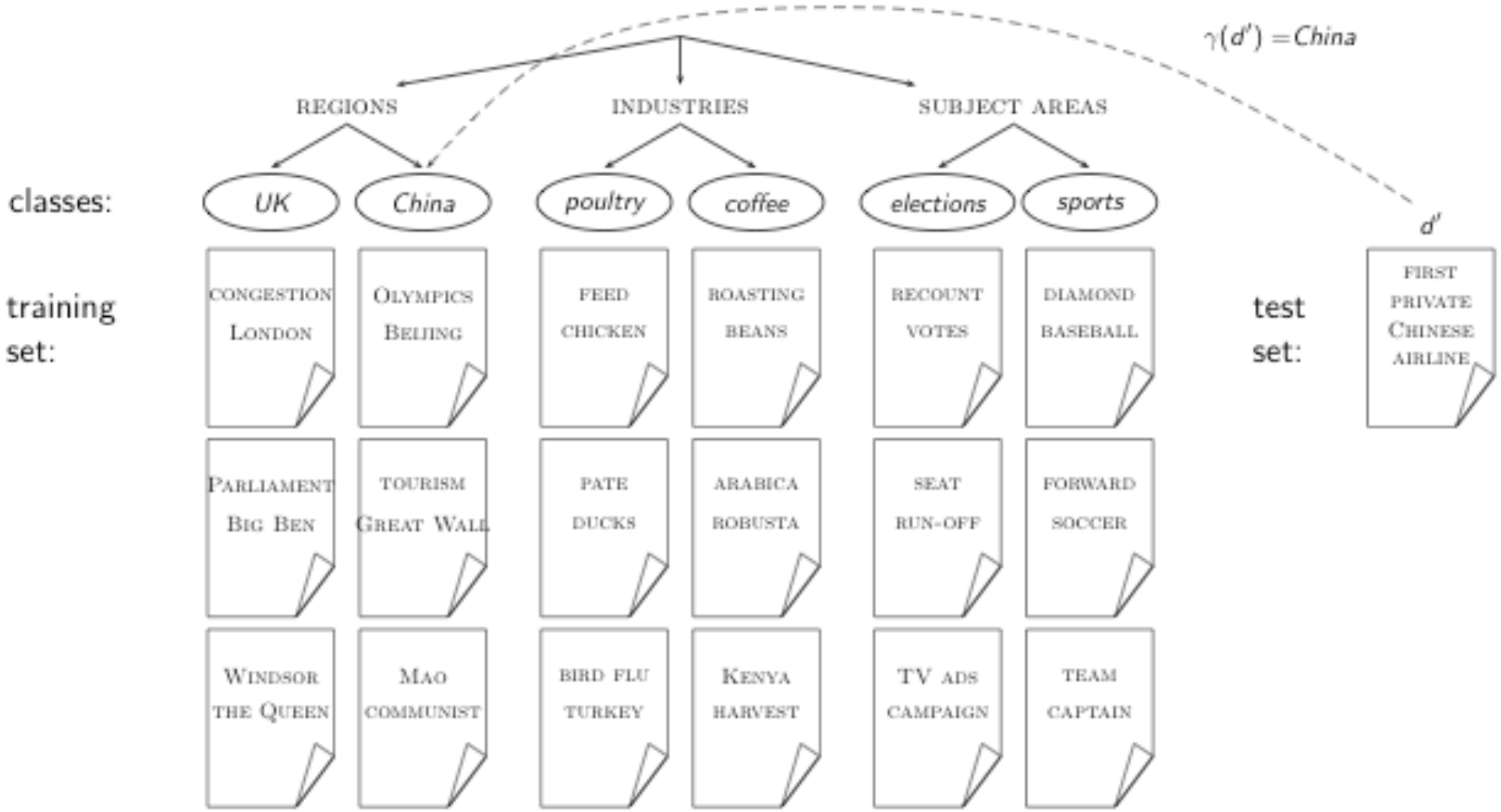
Formal definition of Text Classification: Application/Testing

Given: a description $d \in X$ of a document

Determine: $\Upsilon(d) \in C$,

that is, the class that is most appropriate for *d*

Topic classification



Exercise

- Find examples of uses of text classification in information retrieval

Examples of how search engines use classification

- Crawling —> Classify pages as news or not —> impacts crawling frequency
- Indexing —> Classify pages into tiers — for improving search quality — so “high class”, middle class, landfill
- classify users (personalization) —> TAMU or not, Harvard or not, location, country (but that’s easy!), language (also probably easy), my interests (i want to buy something or not)
- Craigslist —> classify users as browsing (not buying), ready to buy
- More generally, query intent (buy or not buy)
- Queries — return a map? or an infobox? or what?who /what /where?
- Ranking —> Classify document-query pairs as Perfect, Good, etc. (that’s really hard, and something we would like to do)
- Web pages —> topic (sports? or computers? or what?), malicious or not, often updated or not, adult content or not, language of the page, location of the page, static vs dynamic page,

Classification methods: 1. Manual

- Manual classification was used by Yahoo in the beginning of the web. Also: ODP, PubMed
- Very accurate if job is done by experts
- Consistent when the problem size and team is small
- Scaling manual classification is difficult and expensive.
- → We need automatic methods for classification.

Classification methods: 2. Rule-based

- Our Google Alerts example was rule-based classification.
- There are IDE-type development environments for writing very complex rules efficiently. (e.g., Verity)
- Often: Boolean combinations (as in Google Alerts)
- Accuracy is very high if a rule has been carefully refined over time by a subject expert.
- Building and maintaining rule-based classification systems is cumbersome and expensive.

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
subtopic
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
* 0.70 film ACCRUE
** 0.50 STEM
    /wordtext = film
** 0.50 motion-picture PHRASE
*** 1.00 WORD
    /wordtext = motion
*** 1.00 WORD
    /wordtext = picture
** 0.50 STEM
    /wordtext = movie
* 0.50 video ACCRUE
** 0.50 STEM
    /wordtext = video
** 0.50 STEM
    /wordtext = vcr
# End of art topic
```

Classification methods: 3. Statistical/Probabilistic

- This was our definition of the classification problem – text classification as a learning problem
- (i) Supervised learning of a the classification function Υ and

  (ii) its application to classifying new documents
- We will look at a couple of methods for doing this: Naive Bayes, Rocchio, kNN, SVMs
- No free lunch: requires hand-classified training data
- But this manual classification can be done by non-experts.

Vector Space Classification

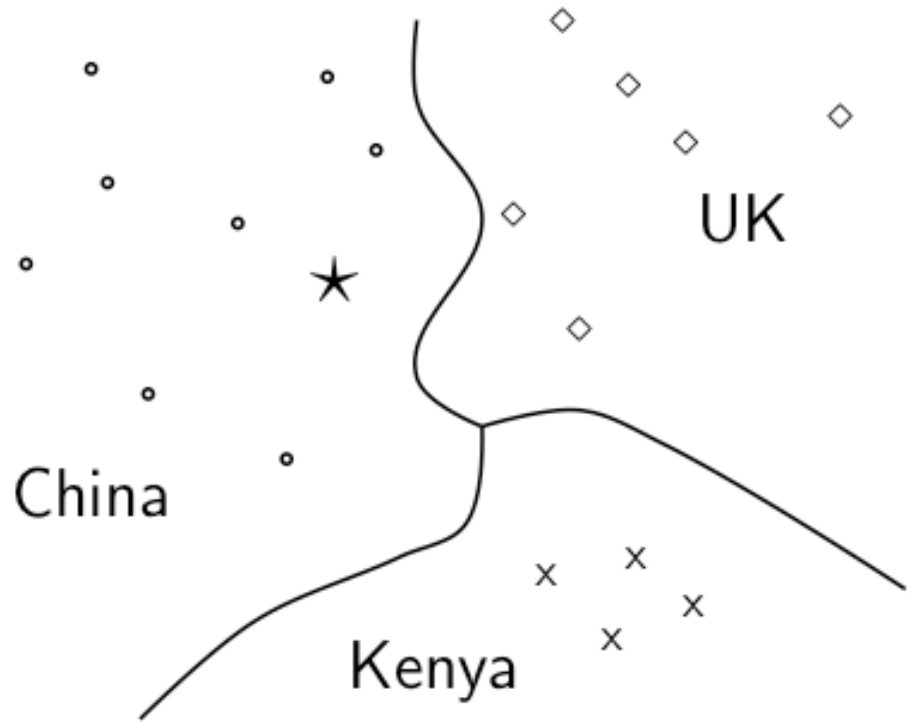
Recall vector space representation

- 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

Rocchio

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.

Definition of centroid

$$\vec{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}(d)$$

where D_c is the set of all documents that belong to class c and $\vec{v}(d)$ is the vector space representation of *d*.

Rocchio algorithm

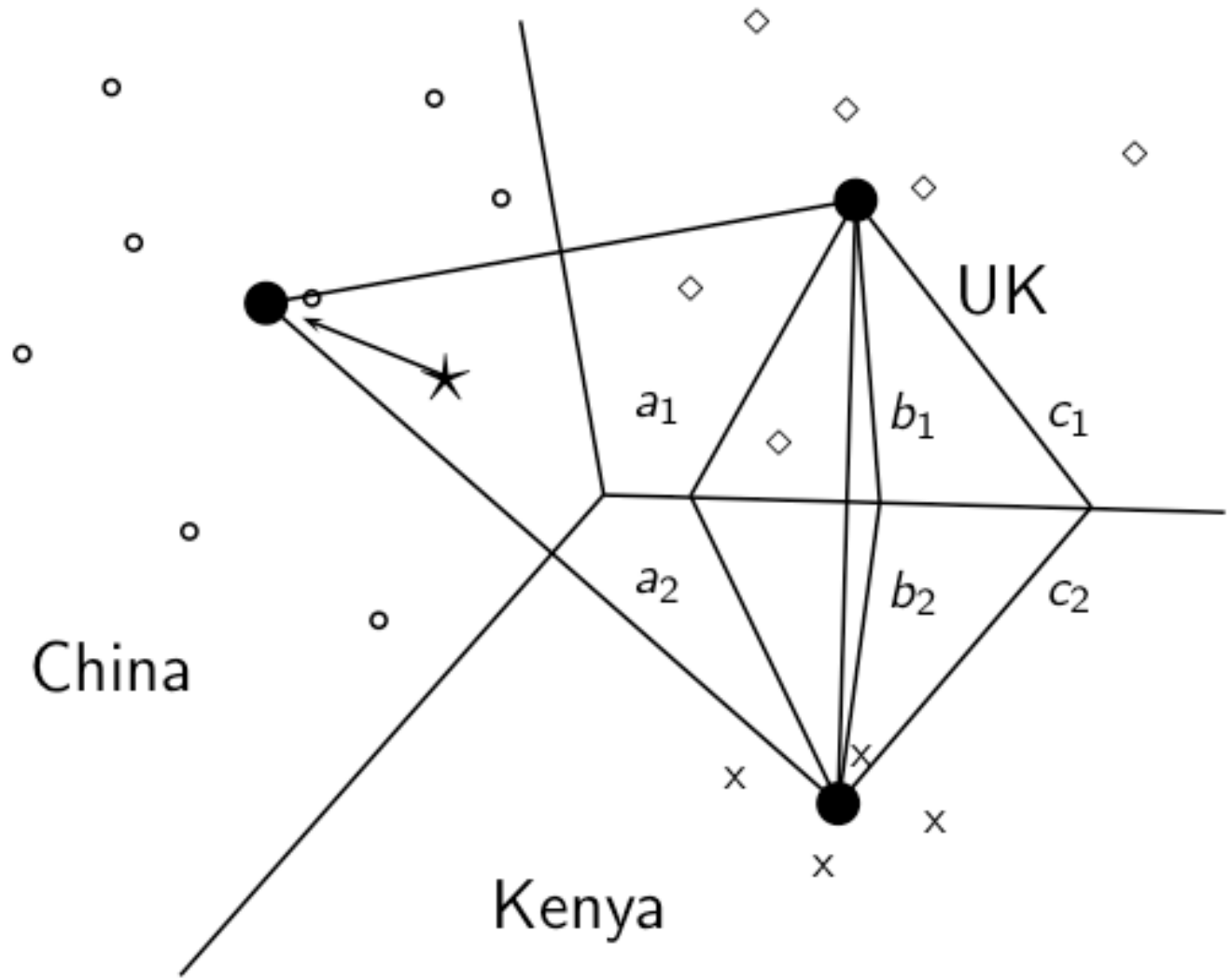
TRAINROCCHIO($\mathbb{C}, \mathbb{D}$)

1 **for each** $c_j \in \mathbb{C}$
2 **do** $D_j \leftarrow \{d : \langle d, c_j \rangle \in \mathbb{D}\}$
3 $\quad \vec{\mu}_j \leftarrow \frac{1}{|D_j|} \sum_{d \in D_j} \vec{v}(d)$
4 **return** $\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}$

APPLYROCCHIO($\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}, d$)

1 **return** $\arg\min_j |\vec{\mu}_j - \vec{v}(d)|$

Rocchio illustrated : a1 = a2, b1 = b2, c1 = c2



Rocchio properties

- Rocchio forms a simple representation for each class: the centroid
  - We can interpret the centroid as the prototype of the class.
- Classification is based on similarity to / distance from centroid/prototype.
- Does not guarantee that classifications are consistent with the training data!

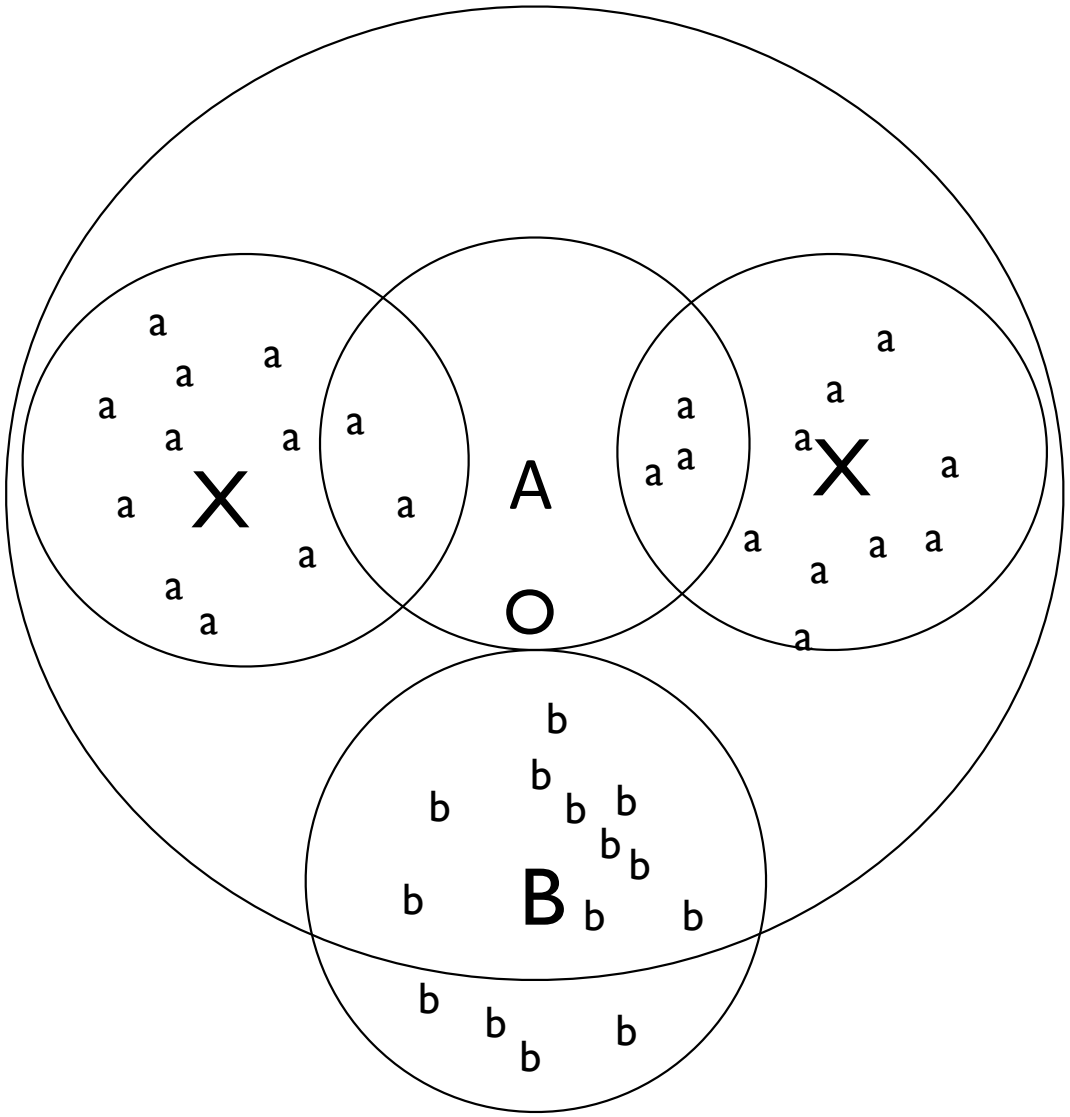
Time complexity of Rocchio

mode	time complexity
training	$\Theta(\lvert\mathbb{D}\rvert L_{\text{ave}} + \lvert\mathbb{C}\rvert\lvert V\rvert) \approx \Theta(\lvert\mathbb{D}\rvert L_{\text{ave}})$
testing	$\Theta(L_{\text{a}} + \lvert\mathbb{C}\rvert M_{\text{a}}) \approx \Theta(\lvert\mathbb{C}\rvert M_{\text{a}})$

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?

- A is centroid of the a's, B is centroid of the b's.
- The point o is closer to A than to B.
- But o is a better fit for the b class.
- A is a multimodal class with two prototypes.
- But in Rocchio we only have one prototype.

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.

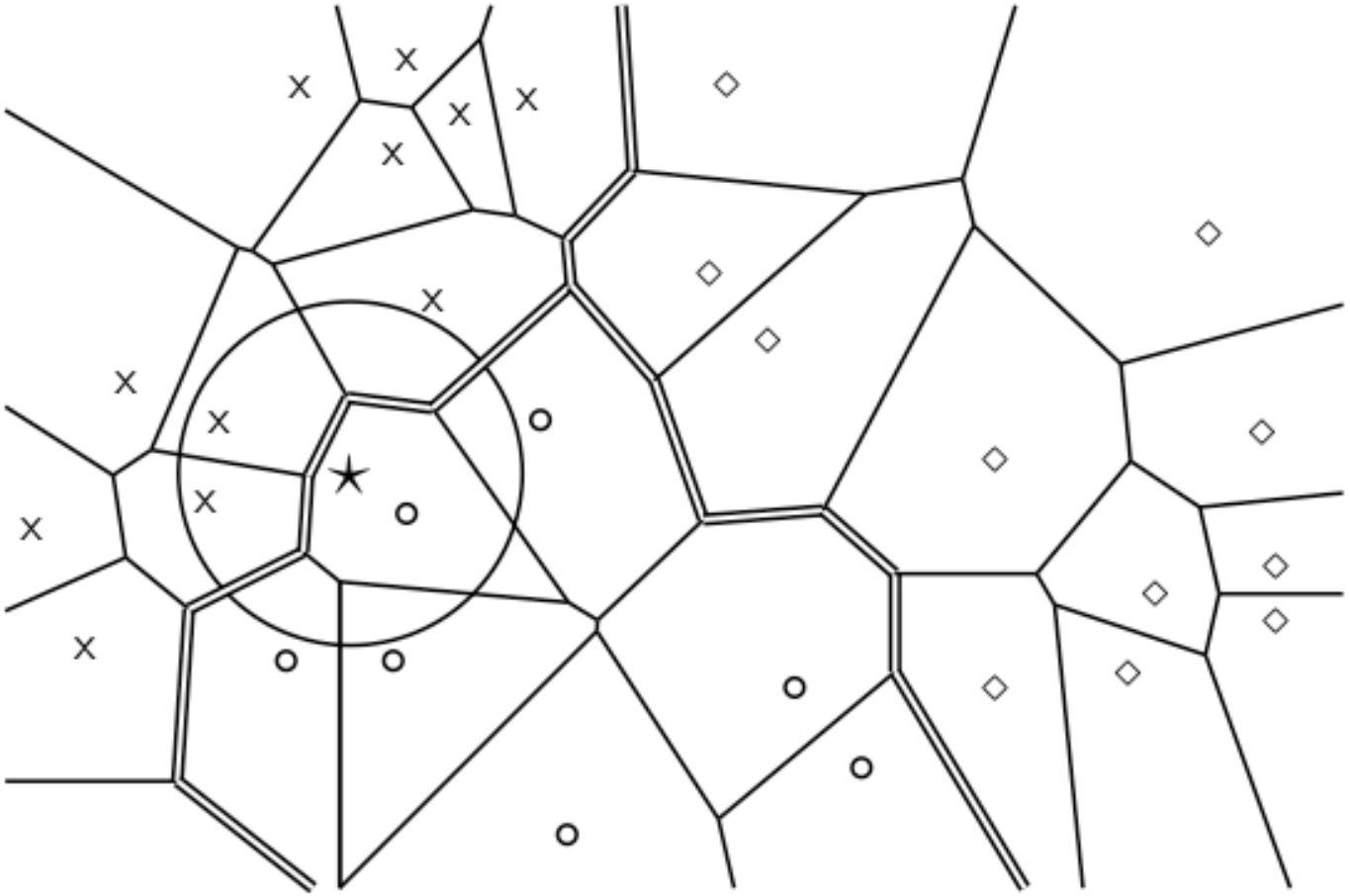
kNN classification

- 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 $P(c|d)$

Probabilistic kNN



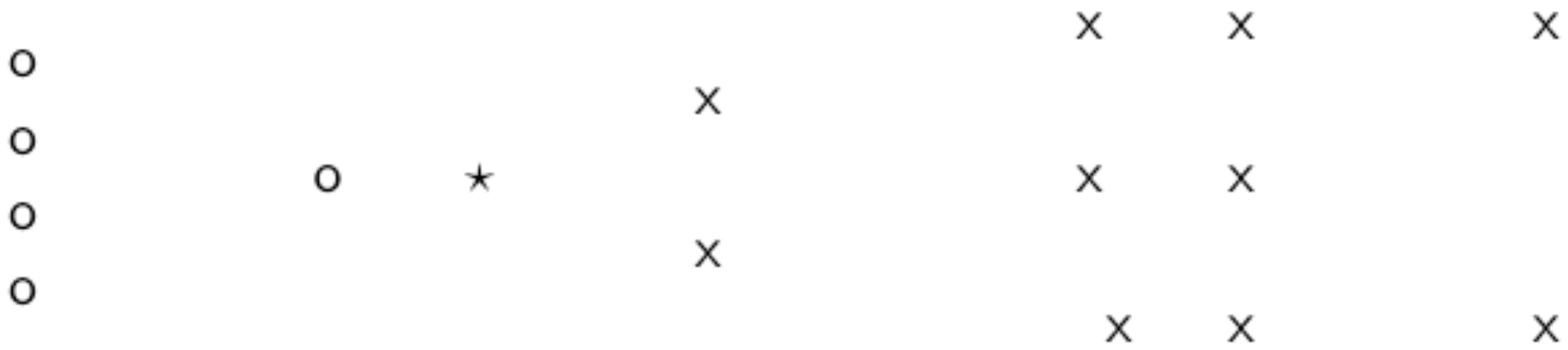
1NN, 3NN classification decision for star?

kNN algorithm

TRAIN-KNN($\mathbb{C}, \mathbb{D}$)
1 $\mathbb{D}' \leftarrow$ PREPROCESS($\mathbb{D}$)
2 $k \leftarrow$ SELECT-K($\mathbb{C}, \mathbb{D}'$)
3 **return** $\mathbb{D}', k$

APPLY-KNN($\mathbb{D}', k, d$)
1 $S_k \leftarrow$ COMPUTENEARESTNEIGHBORS($\mathbb{D}', k, d$)
2 **for each** $c_j \in \mathbb{C}(\mathbb{D}')$
3 **do** $p_j \leftarrow |S_k \cap c_j|/k$
4 **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?

Time complexity of kNN

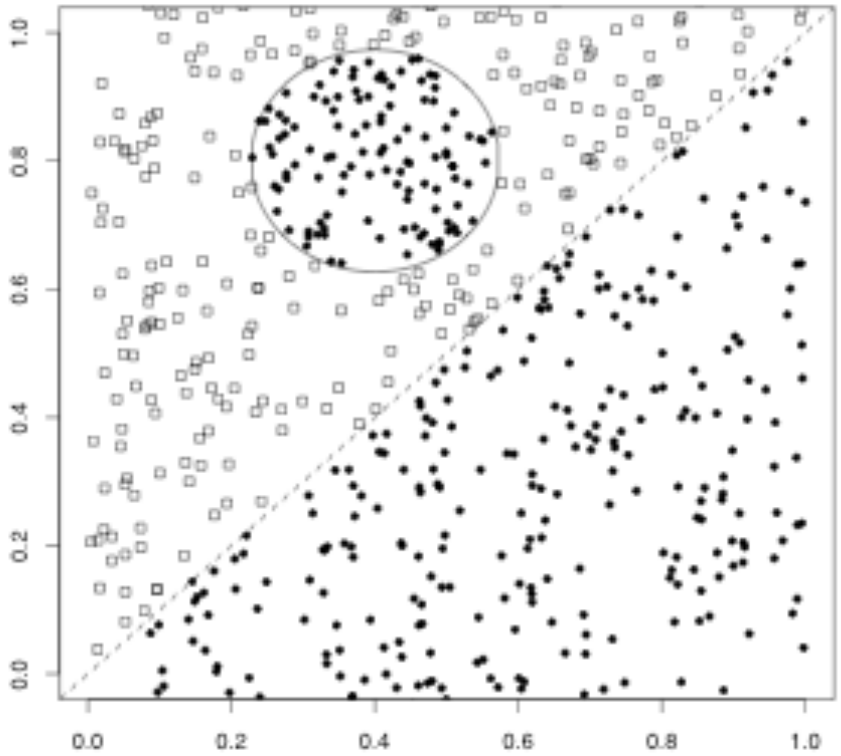
kNN with preprocessing of training set

training $\Theta(|\mathbb{D}|L_{ave})$

testing $\Theta(L_a + |\mathbb{D}|M_{ave}M_a) = \Theta(|\mathbb{D}|M_{ave}M_a)$

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

A nonlinear problem



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

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.
- Optimality result: asymptotically zero error if Bayes rate is zero.
- But kNN can be very inaccurate if training set is small.