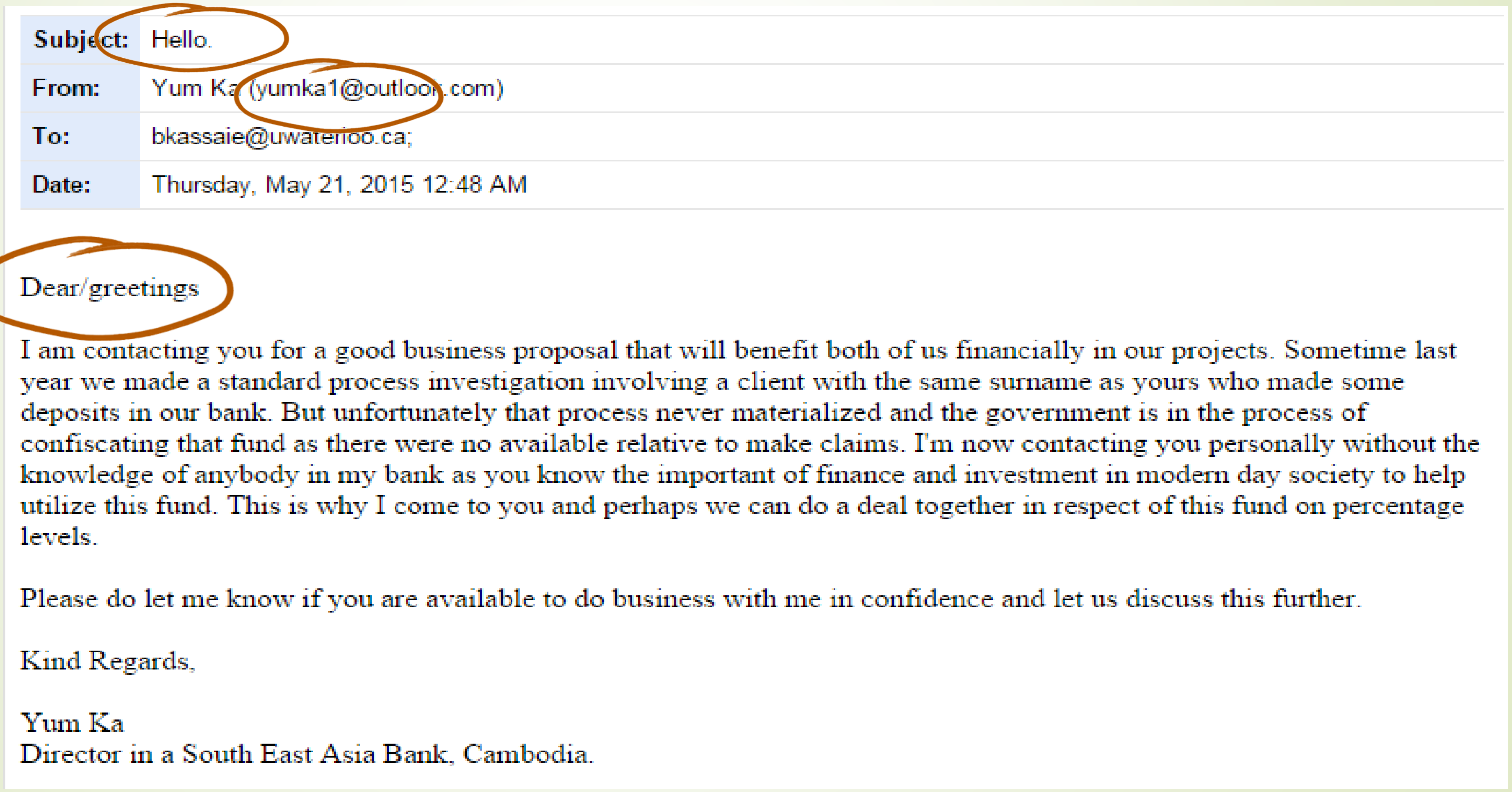
Text Classification

Besat Kassaie

Outline

- Introduction
- Text classification definition
- Naive Bayes
- Vector Space Classification
  - Rocchio Classficiation
  - KNN Classification
- New Applications

Spam Detection



Subject:	Hello.
From:	Yum Ka (yumka1@outlook.com)
To:	bkassaie@uwaterloo.ca;
Date:	Thursday, May 21, 2015 12:48 AM

Dear/greetings

I am contacting you for a good business proposal that will benefit both of us financially in our projects. Sometime last year we made a standard process investigation involving a client with the same surname as yours who made some deposits in our bank. But unfortunately that process never materialized and the government is in the process of confiscating that fund as there were no available relative to make claims. I'm now contacting you personally without the knowledge of anybody in my bank as you know the important of finance and investment in modern day society to help utilize this fund. This is why I come to you and perhaps we can do a deal together in respect of this fund on percentage levels.

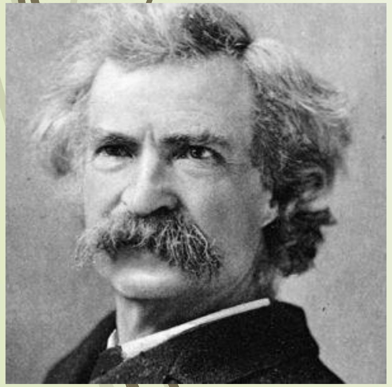
Please do let me know if you are available to do business with me in confidence and let us discuss this further.

Kind Regards,

Yum Ka
Director in a South East Asia Bank, Cambodia.

Authorship identification

- Was Mark Twain the writer of "Quintus Curtius Snodgrass" (QCS) letters?
- Mark Twain's role in the Civil War has been a subject of dispute for years.
- The evidence of Twain's possible military connection in New Orleans was from content of ten letters published in New Orleans' Daily Crescent.
- In these letters, which have been credited to Twain and were signed "Quintus Curtius Snodgrass" (QCS), the writer described his military adventures.
- Bringar (1963) applied statistical tests to QCS letters.
- Bringar used word frequency distribution to determine Mark Twain's writing style
- Mark Twain was not the author of the disputed letters!



Gender Identification

- Determining if an author is male or female
- Men and women inherently use different classes of language styles.
  - Samples:
  - a large number of determiners (a, the, that, these) and quantifiers (one, two, more, some) are male indicators.
  - Using large number of the pronouns (I, you, she, her, their, myself, yourself, herself) are all strong female indicators.

- Applications:
  - Marketing, personalization, legal investigation

Sentiment Analysis

- Is the attitude of this text positive or negative?
- Rank the attitude of this text from 1 to 5
- Sample Applications:
  - Movie: is this review positive or negative?
  - Products: what do people think about the new iPhone?
  - Public sentiment: how is consumer confidence? Is despair increasing?
  - Politics: what do people think about this candidate or issue?
  - Prediction: predict election outcomes or market trends from sentiment

Subject Assignment

MEDLINE Article

Brain and Cognition

Syntactic frame and verb bias in aphasia: Plausibility judgments of undergoer-subject sentences

[illegible]

??

- What is the subject of this article:
  - Antagonists and inhibitors
  - Blood Supply
  - Chemistry
  - Drug Therapy
  - Embryology
  - ….

Text Classification Definition

- Input:
  - a document d
  - a fixed set of classes $C=\{c_1,c_2,\ldots,c_j\}$
- Output: a predicted class $c \in C$

Approaches to Text Classification

- Manual
  - Many classification tasks have traditionally been solved manually.
    - e.g. Books in a library are assigned Library of Congress categories by a librarian
  - manual classification is expensive to scale
- Hand-crafted rules
  - A rule captures a certain combination of keywords that indicates a class.
    - e.g. (multicore OR multi-core) AND (chip OR processor OR microprocessor)
  - good scaling properties
  - creating and maintaining them over time is labor intensive
- Machine learning based methods

Supervised Learning

- Input:
  - a document d
  - a fixed set of classes $C=\{c_1,c_2,\dots,c_j\}$
  - a training set of m labeled documents $(d_1,c_1), (d_2,c_2),\dots,(d_m,c_m)$
- Output: a learned classifier $\gamma : d \rightarrow c$

Supervised Learning

- We can use any kind of classifiers:
  - **Naïve bayes**
  - **kNN**
  - **Rocchio**
  - SVM
  - Logistic Regression
  - ….

Naïve Bayes Classification

Naïve Bayes Intuition

- It is a simple classification technique based on Bayes rule
- Documents are represented by "bag of words" model
- The *bag of words model :*
  - a document *d*, is represented by the set of words and their weights determined by the TF weights
  - TF is the number of occurrences of each term.
  - The document "Mary is quicker than John" is, in this view, identical to the document "John is quicker than Mary".

The bag of words representation

$\gamma($

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun… It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.

$)=c$

The bag of words representation

$$\gamma\left(\begin{array}{|l|l|}\hline \texttt{great} & \texttt{2} \\ \hline \texttt{love} & \texttt{2} \\ \hline \texttt{recommend} & \texttt{1} \\ \hline \texttt{laugh} & \texttt{1} \\ \hline \texttt{happy} & \texttt{1} \\ \hline \ldots & \ldots \\ \hline \end{array}\right) = c$$

Features

- Supervised learning classifiers can use any sort of feature
  - URL, email address, punctuation, capitalization, social network features and so on
- In the bag of words view of documents
  - We use **only** word features

Baye’s Rule Applied to Documents and Classes

- For a document d and a class c

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

Naïve Bayes Classifier

$$c_{map} = \underset{c \in C}{argmax}\, p(c|d)$$

MAP is "maximum a posteriori" = most likely class

$$= \underset{c \in C}{argmax} \frac{P(d|c)P(c)}{P(d)}$$

Bayes Rule

$$= \underset{c \in C}{argmax}\, P(d|c)P(c)$$

Dropping the denominator

$$= \underset{c \in C}{argmax} P(x_1, x_1, \dots, x_n|c)\; P(c)$$

Document d represented as Features $x_1..x_n$

Multinomial NB VS Bernoulli NB

- There are two different approaches for setting up an NB classifier.
- **multinomial model**: generates one term from the vocabulary in each position of the document
- **multivariate Bernoulli model:** generates an indicator for each term of the vocabulary, either 1 indicating presence of the term in the document or 0 indicating absence
- Difference in estimation strategies

Document Features

$$c_{map} = \underset{c \in C}{argmax}\, P(d|c)P(c)$$

- Multinomial:
  - $c_{map} = \underset{c \in C}{argmax}\, P(w_1, w_2, \dots, w_n|c)P(c)$
  - $w_1, w_2, \dots, w_n$ is the sequence of terms as it occurs in *d*
- Bernoulli:
  - $c_{map} = \underset{c \in C}{argmax}\, P(e_1, e_2, \dots, e_v|c)P(c)$
  - $e_1, e_2, \dots, e_v$ is a binary vector of dimensionality *V* that indicates for each term whether it occurs in *d* or not

ASSUMPTIONS: Conditional independence

- $P(x_1, x_2, \dots, x_n|c) = P(x_1|c) \cdot P(x_2|c) \cdot \dots \cdot P(x_n|c)$
- Multinomial:
  - $c_{map} = \underset{c \in C}{argmax}\, P(w_1, w_2, \dots, w_n|c)P(c)$
  - $c_{NB} = \underset{c \in C}{argmax} \prod_{1 \le k \le n} P(X_k = w_k|c)P(c)$
- X_k is the random variable for position *k* in the document and takes as values terms from the vocabulary.
- $P(X_k = w_k|c)$ is the probability that in a document of class c the term w will occur in position *k*

ASSUMPTIONS: Conditional independence

- Bernoulli:
  - $c_{map} = \underset{c \in C}{argmax}\ P(e_1, e_2, \ldots, e_v|c)P(c)$
  - $c_{NB} = \underset{c \in C}{argmax}\ \prod_{1 \le i \le v} P(U_i = w_i|c)P(c)$
- U_i is the random variable for vocabulary term *i* and takes as values 0 (absence) and 1 (presence).
- $P(U_i = w_i|c)$ is the probability that in a document of class c the term w_i will occur

ASSUMPTIONS: Bag of words assumption

- if we assumed different term distributions for each position *k*, we would have to estimate a different set of parameters for each *k*
- *positional independence*: The conditional probabilities for a term are the same, independent of position in the document.
- E.g. $P(X_{k_1} = t \mid c) = P(X_{k2} = t \mid c)$
- we have a single distribution of terms that is valid for all positions *k*

$$c_{map} = \underset{c \in C}{argmax}\, P(w_1, w_2, \dots, w_n | c) P(c)$$

$$c_{NB} = \underset{c \in C}{argmax} \prod_{w \epsilon W} P(w|c) P(c)$$

Learning Parameters for Multinomial Naïve Bayes

- Based on Maximum Likelihood Estimation

$$\hat{P}(c_j) = \frac{Count(C = c_j)}{N_{doc}}$$

Fraction of documents which are labeled as class c_j

$$\hat{P}(w_i|c_j) = \frac{Count(w_i, c_j)}{\sum_{w \epsilon V} Count(w, c_j)}$$

Fraction of times word w_i appears among all words of class c_j

Create mega document for class j By concatenating all docs In this class. Use frequency of w in mega document.

Laplace (add-1) smoothing for Naïve Bayes

- Solving the problem of zero probabilities

$$\hat{P}(w_i|c_j) = \frac{Count(w_i, c_j) + 1}{\sum_{w \epsilon V}(Count(w, c_j) + 1)}$$

$$= \frac{Count(w_i, c_j) + 1}{\sum_{w \epsilon V} Count(w, c_j) + |V|}$$

Laplace (add-1) smoothing: unknown words

- Add one extra word to the vocabulary, the "unknown word" , w_u

$$\hat{P}(w_u|c_j) = \frac{Count(w_u, c_j) + 1}{(\sum_{w \epsilon V} count(w, c_j)) + |V + 1|}$$

$$= \frac{1}{(\sum_{w \epsilon V} count(w, c_j)) + |V + 1|}$$

Probability for all unknown words

Example

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Priors:

$P(c) = \frac{3}{4}$

$P(j) = \frac{1}{4}$

Conditional Probabilities:

P(Chinese|*c*) = (5+1) / (8+6) = 6/14 = 3/7

P(Tokyo|*c*) = (0+1) / (8+6) = 1/14

P(Japan|*c*) = (0+1) / (8+6) = 1/14

P(Chinese|*j*) = (1+1) / (3+6) = 2/9

P(Tokyo|*j*) = (1+1) / (3+6) = 2/9

P(Japan|*j*) = (1+1) / (3+6) = 2/9

Choosing a class:

$P(c|d5) \propto 3/4 * (3/7)^3 * 1/14 * 1/14 \approx 0.0003$

$P(j|d5) \propto 1/4 * (2/9)^3 * 2/9 * 2/9 \approx 0.0001$

Vector Space Classification

Vector Space Representation

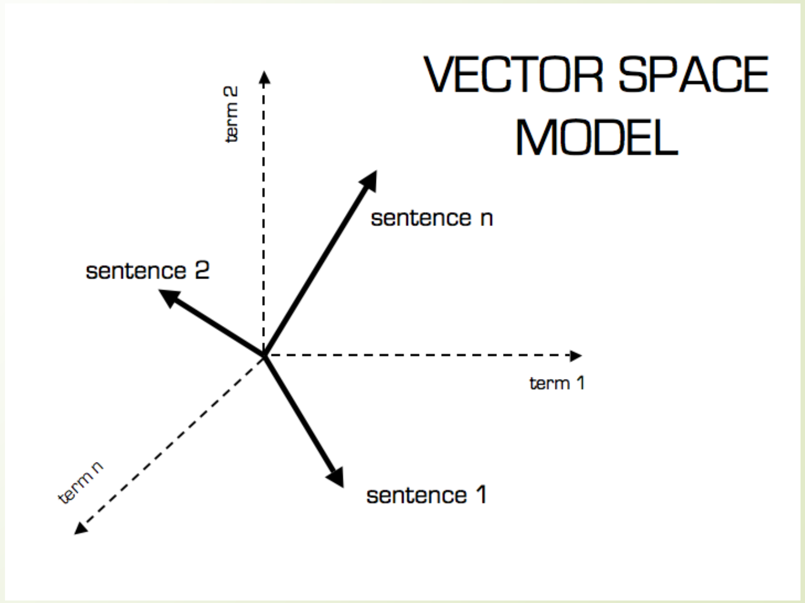
- The document representation in Naive Bayes is a sequence of terms or a binary vector
- A different representation for text classification: **the vector space model**
- Each document represented as a vector with one real-valued component,

usually a tf-idf weight, for each term.

- Each vector is composed of all terms

Of all documents.

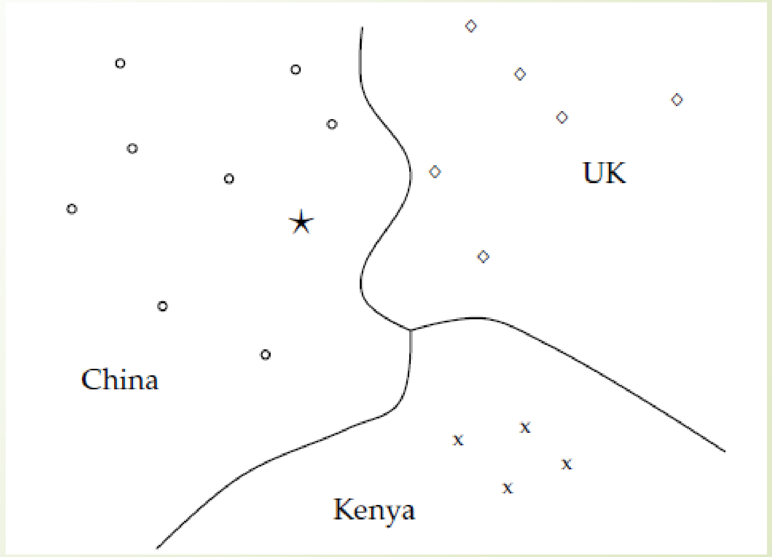
- Terms are axis
- High dimensionality



Vector Space Classification

- Vector space classification is based on the contiguity hypothesis:
  - Objects in the same class form a contiguous region, and regions of different classes do not overlap
- Classification is to compute the boundaries in the space that separate the classes; the decision boundaries
- Two methods will be discussed:

  Rocchio and kNN



Vector Space Classification

Rocchio

Rocchio Classification

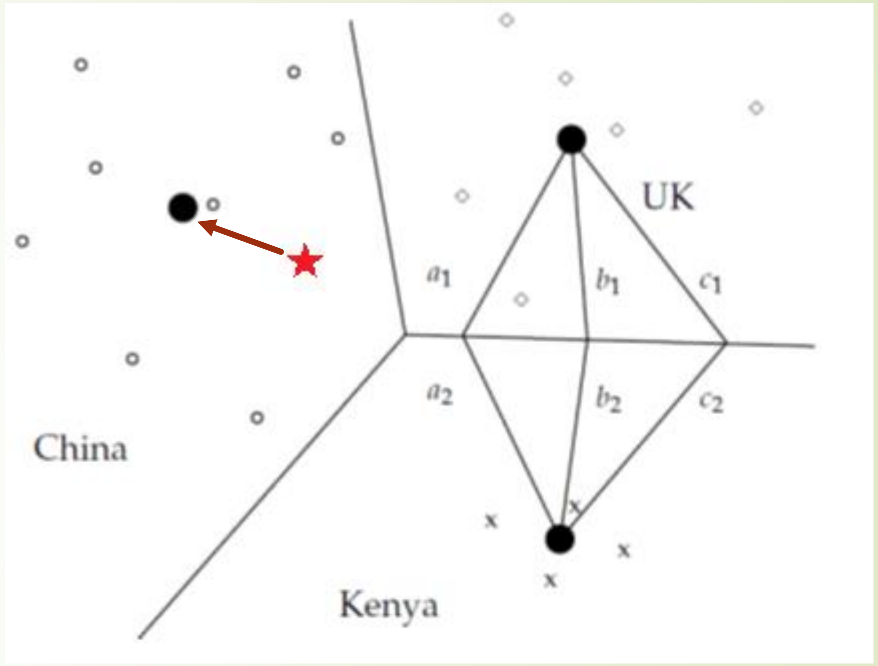
- Basic idea: Rocchio forms a simple representative for each class by using the centroids.
- To compute the centroid of each class:

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

D_c is the set of all documents that belong to class c

v(d) is the vector space representation of d.

Rocchio Classification



- The boundary between two classes in Rocchio classification is the set of points with equal distance from the two centroids

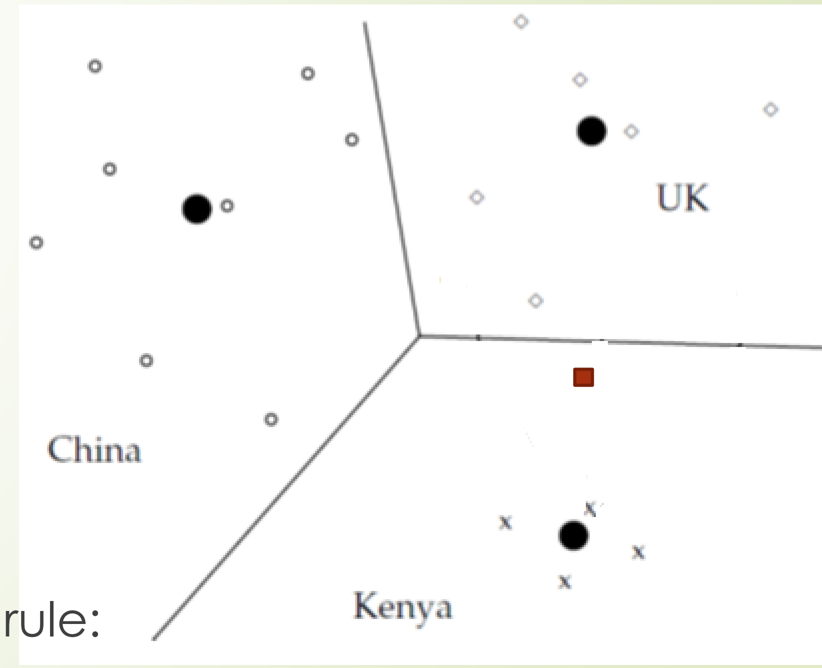
- To classify a new object determine which centroid is closest to and assign it to the corresponding class c_i .

Problems with Rocchio Classification

- Implicitly assumes that classes are spheres with similar radius
- Ignores details of distribution of points within a class and only considers centroid distances.

- Red cube is assigned to *Kenya* class
- But is a better fit for *UK* class

since UK is more scattered than *Kenya*



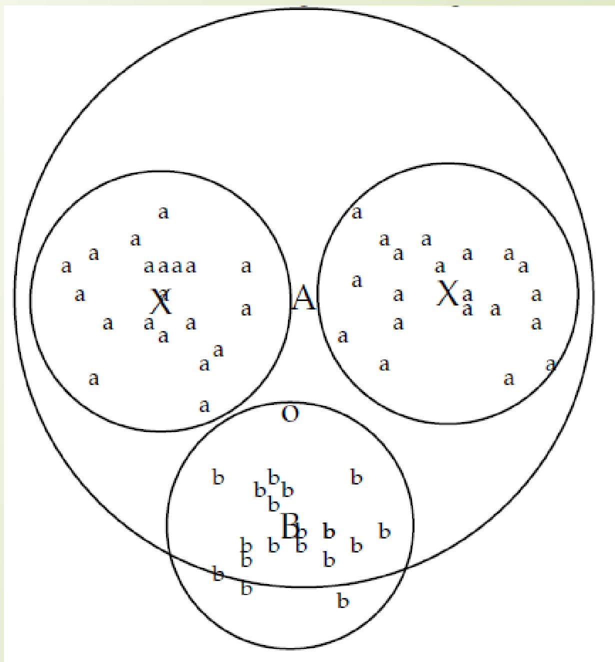
- Not requiring similar radius → modified decision rule:

Assign *d* to class c iff $|\vec{\mu}(c) - \vec{v}(d)| \leq |\vec{\mu}(\overline{c}) - \vec{v}(d)| - b$

Problems with Rocchio Classification

- Does not work well for classes than cannot be accurately represented by a single “center”

- Class “a” consists of two different clusters.
- Rocchio classification will misclassify “o” as “a” because it is closer to the centroid A of the “a” class than to the centroid B of the “b” class.



Vector Space Classification
K Nearest Neighbor

K Nearest Neighbor Classification

- For kNN we assign each document to the majority class of its *k* closest neighbors where *k* is a parameter.
- The rationale of kNN classification is that, based on the contiguity hypothesis, we expect a test document *d* to have the same label as the training documents located in the local region surrounding *d*.
- Learning: just store the labeled training examples D

K Nearest Neighbor Classification

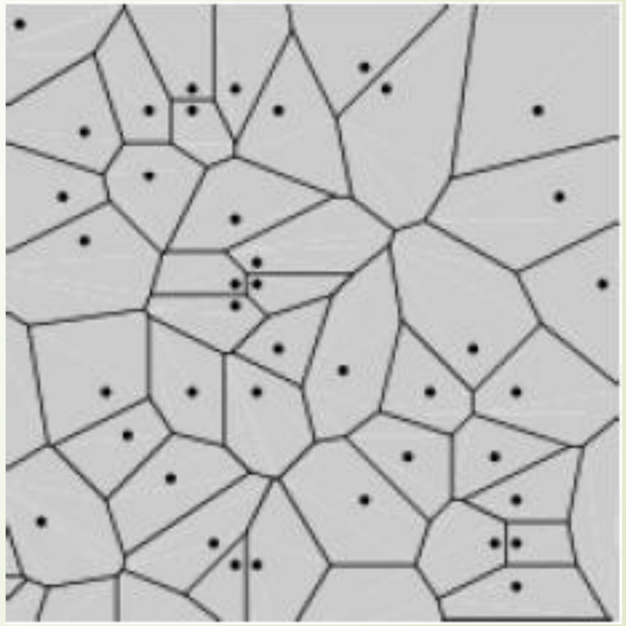
- Determining class of a new document :

- For *k* = 1: Assign each object to the class of its closest neighbor.
- For *k* > 1: Assign each object to the majority class among its *k* closest neighbors.
- The parameter *k* must be specified in advance. *k* = 3 and *k* = 5 are common choices, but much larger values between 50 and 100 are also used

kNN decision boundary

- The decision boundary defined by the Voronoi tessellation
- Assuming k = 1: For a given set of objects in the space, each object define a cell consisting of all points that are closer to that object than to other objects.
- Results in a set of convex polygons;

  so-called Voronoi cells.



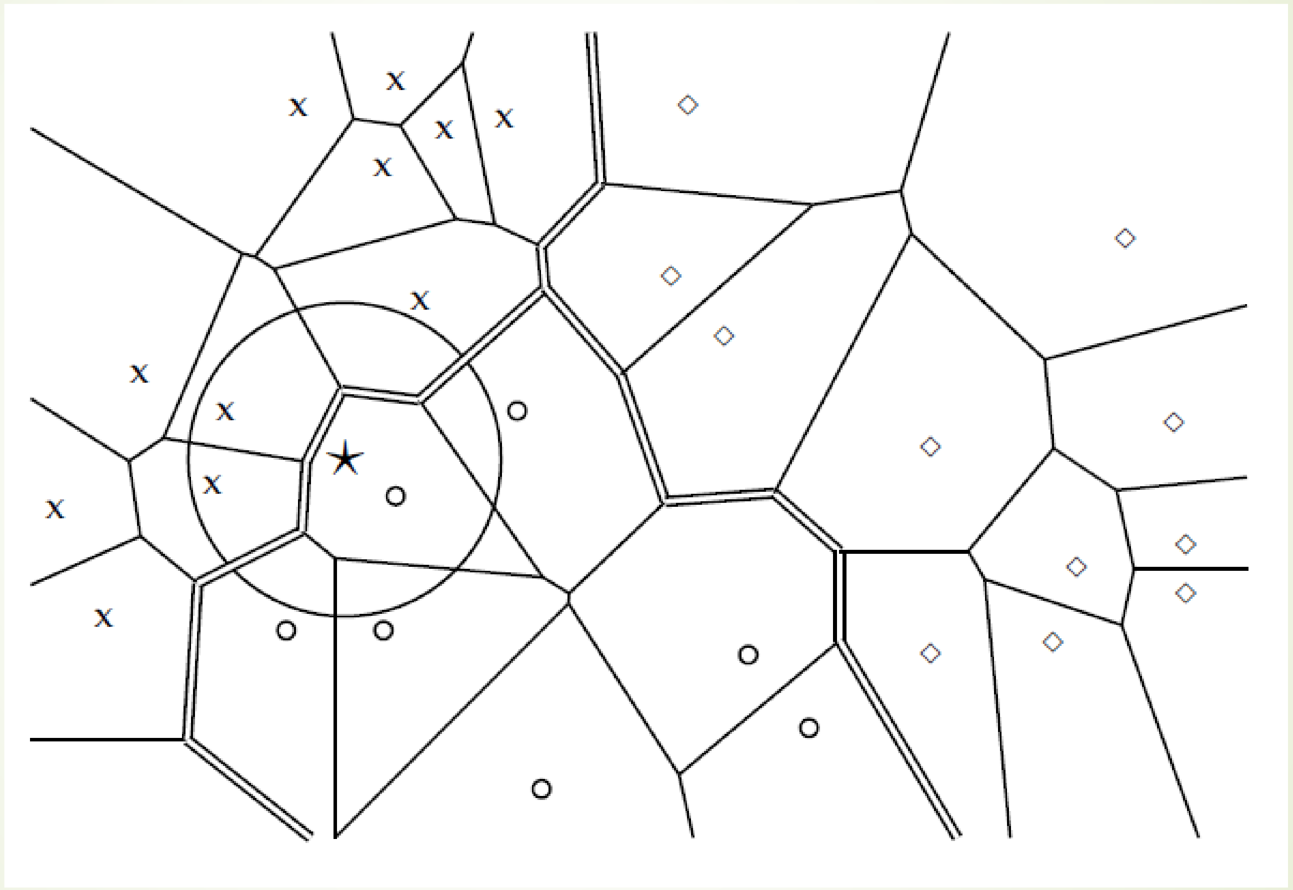
- Decomposing a space into such cells

gives us the so-called Voronoi tessellation.

- In the case of $k > 1$, the Voronoi cells

are given by the regions in the space for which

the set of k nearest neighbors is the same.



- Decision boundary for 1NN (double line): defined along the regions of Voronoi cells for the objects in each class. Shows the non-linearity of *k*NN

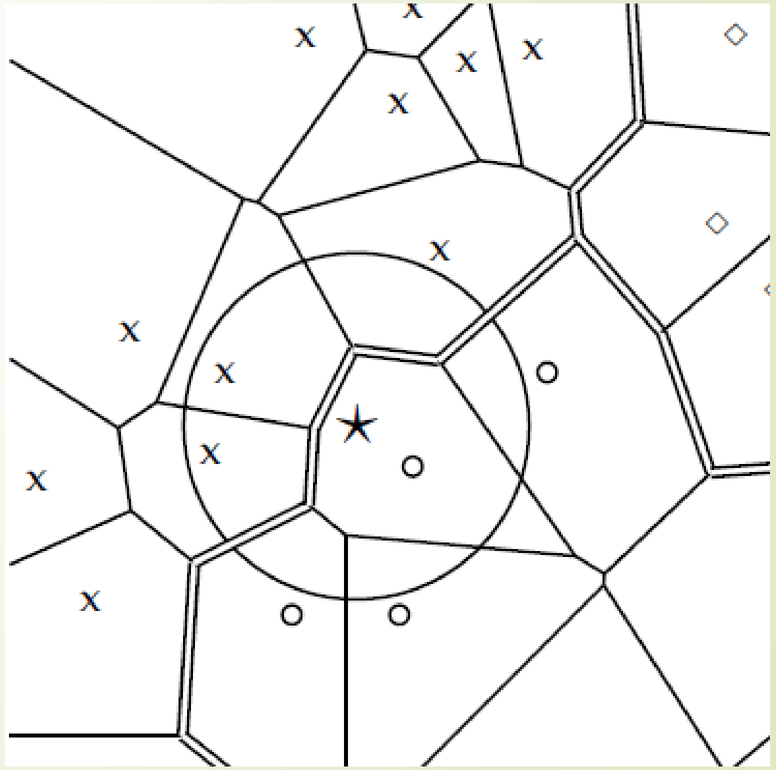
kNN: Probabilistic version

- There is a probabilistic version of kNN classification algorithm.
- We can estimate the probability of membership in class c as the proportion of the *k* nearest neighbors in c.

- $\hat{P}\ (circle\ class|star)\ =\ 1/3$
- $\hat{P}(X\ class|star)\ =\ 2/3$
- $\hat{P}(diamond\ class|star)\ =\ 0$

- The 3NN estimate: $\hat{P}(circle\ class|star)\ =\ 1/3$
- The 1NN estimate: $\hat{P}(circle\ class|star)\ =\ 1$
- 3NN preferring the X class

and 1NN preferring the circle class .



kNN: distance weighted version

- We can weight the "votes" of the *k* nearest neighbors by their cosine similarity:

$$Score(c,d) = \sum_{\acute{d} \epsilon S_k(d)} I_c(\acute{d}) \cos(\vec{v}(d), \vec{v}(\acute{d}))$$

- $S_k(d)$ is the set of d's k nearest neighbors
- $I_c(d')$ = 1 iff d' is in class c and 0 otherwise.
- Assign the document to the class with the highest score.
- Weighting by similarities is often more accurate than simple voting.

kNN Properties

- No training necessary
- Scales well with large number of classes
  - Don’t need to train n classifiers for n classes
- May be expensive at test time
- In most cases it’s more accurate than NB or Rocchio

Some Recent Applications of Text Classification

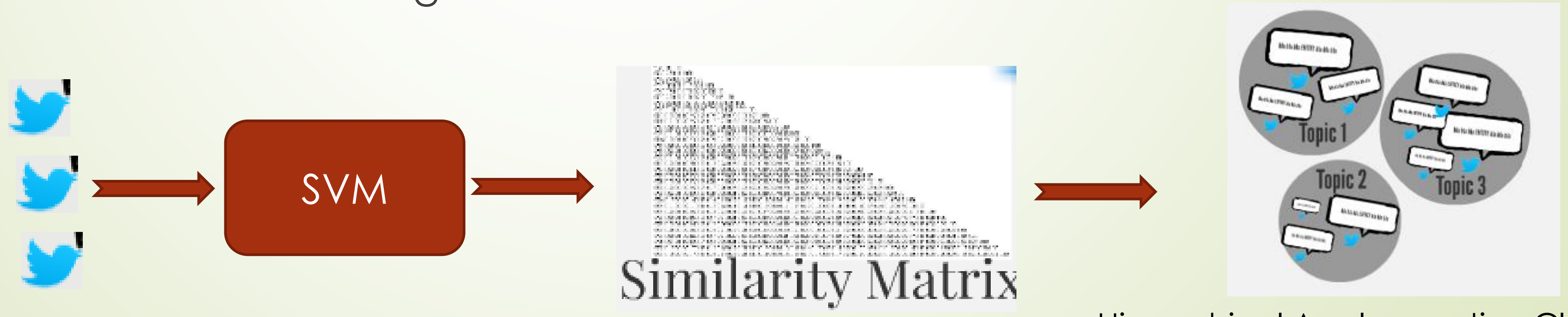
- Social networks are rich source of text data e.g. twitter
- Many new applications are emerging based on these sources
- We will discuss three recent works based on twitter data
- *“Learning Similarity Functions for Topic Detection in Online Reputation Monitoring. Damino Spina, et. Al, SIGIR, 2014”*
- *“Target-dependent Churn Classification in Microblogs, Hadi Amiri, et.al, AAAI 2015”*
- *“Fine-Grained Location Extraction from Tweets with Temporal Awareness, Chenliang Li, SIGIR, 2014”*

Learning Similarity Functions for Topic Detection in Online Reputation Monitoring

- What are people are saying about a given entity (company, brand, organization, personality,…)
- Is there any issues that may damage the reputation of the entity?
- In order to answer such questions, reputation experts have to daily monitor social networks such as twitter
- The paper aim is to solve this problem automatically as topic detection task
- Reputation alerts must be detected early before they explode. So there are a few number of related tweets
- Probabilistic generative approaches are less appropriate here because of data sparsity

Learning Similarity Functions…

- Paper proposes a hybrid approach: supervised + unsupervised
- Tweets need to be clustered based on their topics
- Clusters change over time based on data stream
- Tweets with similar topics belong to the same cluster
- For clustering we need a similarity function.
- Classification (SVM) is used to learn the similarity function.
- Two tweets belong to either similar or not similar classes



Hierarchical Agglomerative Clustering

Learning Similarity Functions…

- Dataset is comprised of 142,527 manually annotated tweets.
- Features used for classification:
- Term Features:
  - taking into account similarity between terms in the tweets
  - Tweets sharing high percentage of vocabulary are likely about same topic
- Semantic features:
  - Useful for tweets that do not have words in common.
  - Used Wikipedia as a knowledge based to find semantically related words. e.g. mexico, mexicanas
- Metadata features:
  - Such as author, URLs, hashtags,…
- Time-aware features:
  - Time stamp of tweets

Fine-Grained Location Extraction from Tweets with Temporal Awareness

- Through tweets, users often casually or implicitly reveal their current locations and short term plans where to visit next, at fine-grained granularity
- Such information enables tremendous opportunities for personalization and location-based services/marketing
- This paper is about extracting fine-grained locations mentioned in tweets with temporal awareness.

Fine-Grained Location Extraction…

- Dataset: a sample of 4000 manually annotated tweets
- Classifier: Conditional Random Fields
- Some features:
  - Lexical Features: words itself, its lower case, bag-of-words of the context window,
  - Grammatical Features: POS tags (for verb tense), word groups (shudd, shuld, shoud)
  - …
- If a user mentions a point-of-interest (POI) (e.g., restaurant, shopping mall,…) in her tweet:
  - The name of the POI is extracted
  - The tweet is assigned into one of these classes: The POI is visited, is currently at, or will soon visit (temporal awareness).

Target-dependent Churn Classification in Microblogs

- The problem of classifying micro-posts as churny or non-churny with respect to a given brand (telecom companies) using Twitter data.
  - Whether a tweet is an indicator that user is going to cancel a service
- Sample tweets:
  - "One of my main goals for 2013 is to leave *BrandName*"
  - "I cant take it anymore, the unlimited data isn't even worth it"
  - "My days with *BrandName* are numbered"

Target-dependent Churn Classification in Microblogs

- Dataset : 4800 tweets about three telecommunication brands
- Classification method: linear regression, SVM, logistic regression
- Features:
  - Demographic Churn Indicators:
    - Activity ratio (if a user is not active with respect to any competitors, then he is less likely to send churny contents about a target brand),
    - # followers and friends, …
  - Content Churn Indicators:
    - Sentiment features, Tense of tweet,…

  …

THANK YOU

References

https://web.stanford.edu/class/cs124/lec/sentiment.pptx

https://web.stanford.edu/class/cs124/lec/naivebayes.pdf

An introduction to Information Retrieval, Christopher D. Manning

,Prabhakar Raghavan,Hinrich Schütze,2009

Learning Similarity Functions for Topic Detection in Online Reputation Monitoring. Damino Spina, et. Al, SIGIR, 2014
Target-dependent Churn Classification in Microblogs, Hadi Amiri, et.al, AAAI 2015
Fine-Grained Location Extraction from Tweets with Temporal Awareness, Chenliang Li, SIGIR, 2014