CSCI 5417
Information Retrieval Systems

Jim Martin

Lecture 12
10/4/2011

Today 10/4

- Classification
  - Review naïve Bayes
  - K-NN methods
- Quiz Review

Categorization/Classification

- Given:
  - A description of an instance, $x \in X$, where X is the *instance language* or *instance space*.
    - Issue: how to represent text documents.
  - And a fixed set of categories:

    $C = \{c_1, c_2, \ldots, c_n\}$
- Determine:
  - The category of x: $c(x) \in C$, where $c(x)$ is a *categorization function* whose domain is X and whose range is C.
    - We want to know how to build categorization functions (i.e. "classifiers").

Bayesian Classifiers

Task: Classify a new instance D based on a tuple of attribute values $D = \langle x_1, x_2, \ldots, x_n \rangle$ into one of the classes $c_j \in C$

$$c_{MAP} = \underset{c_j \in C}{\operatorname{argmax}}\, P(c_j \mid x_1, x_2, \ldots, x_n)$$

$$= \underset{c_j \in C}{\operatorname{argmax}} \frac{P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)}{P(x_1, x_2, \ldots, x_n)}$$

$$= \underset{c_j \in C}{\operatorname{argmax}}\, P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)$$

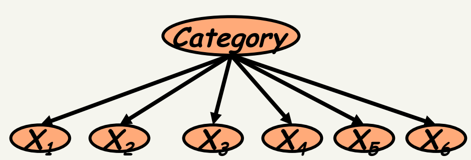
Naïve Bayes Classifiers

- $P(c_j)$
  - Can be estimated from the frequency of classes in the training examples.
- $P(x_1, x_2, \ldots, x_n | c_j)$
  - $O(|X|^n \bullet |C|)$ parameters
  - Could only be estimated if a very, very large number of training examples was available.

Naïve Bayes Conditional Independence Assumption:

- Assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities $P(x_i|c_j)$.

Learning the Model

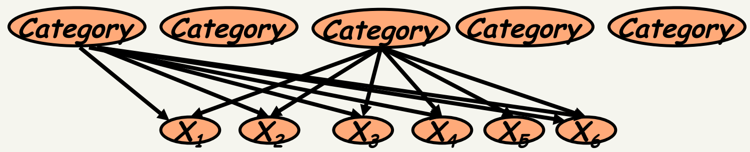


- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{N(C = c_j)}{N}$$

$$\hat{P}(x_i | c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}$$

Learning the Model



- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{N(C = c_j)}{N}$$

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}$$

Smoothing to Avoid Overfitting

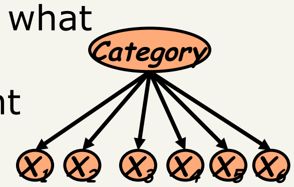
$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j) + 1}{N(C = c_j) + k}$$

Add-One smoothing

# of values of X_i

Generative Models

- This kind of scheme is often referred to as a generative model. To do classification we try to imagine what the process of creating, or generating, the document might have looked like.
- Learning from training data is therefore a process of learning the nature of the categories.
  - What does it mean to be a sports document.



Naïve Bayes example

- Given: 4 documents
  - D1 (sports): China soccer
  - D2 (sports): Japan baseball
  - D3 (politics): China trade
  - D4 (politics): Japan Japan exports
- Classify:
  - D5: soccer
  - D6: Japan
- Use
    - Add-one smoothing
  - Multinomial model
  - Multivariate binomial model

Naïve Bayes example

- V is {China, soccer, Japan, baseball, trade exports}
- |V| = 6
- Sizes
  - Sports = 2 docs, 4 tokens
  - Politics = 2 docs, 5 tokens

Japan	Raw	Sm
Sports	1/4	2/10
Politics	2/5	3/11

soccer	Raw	Sm
Sports	1/4	2/10
Politics	0/5	1/11

Naïve Bayes example

- Classifying
  - Soccer (as a doc)
    - Soccer | sports = .2
    - Soccer | politics = .09

      Sports > Politics or

      .2/.2+.09 = .69

      .09/.2+.09 = .31

New example

- What about a doc like the following?
  - *Japan soccer*
    - Sports
      - P(japan|sports)P(soccer|sports)P(sports)
      - .2 * .2 * .5 = .02
    - Politics
      - P(japan|politics)P(soccer|politics)P(politics)
      - .27 * .09 *. 5 = .01
    - Or
      - .66 to .33

Quiz

1. Sleeping
2. Irrelevant documents due to stemming.
   1. *Stockings and stocks stem to stock*
3. All of the them
4. True
5. True
6. Slows it down. Rel feedback results in long vector lengths in Q_m
7. .6
8. $D_2 > D_3 > D_1$

Classification: Vector Space Version

- The naïve Bayes (probabilistic approach) is fine, but it ignores all the infrastructure we've built up based on the vector-space model.
  - Infrastructure that supports ad hoc retrieval and is highly optimized in terms of space and time.
  - It would be nice to be able to use it for something

Recall: Vector Space Representation

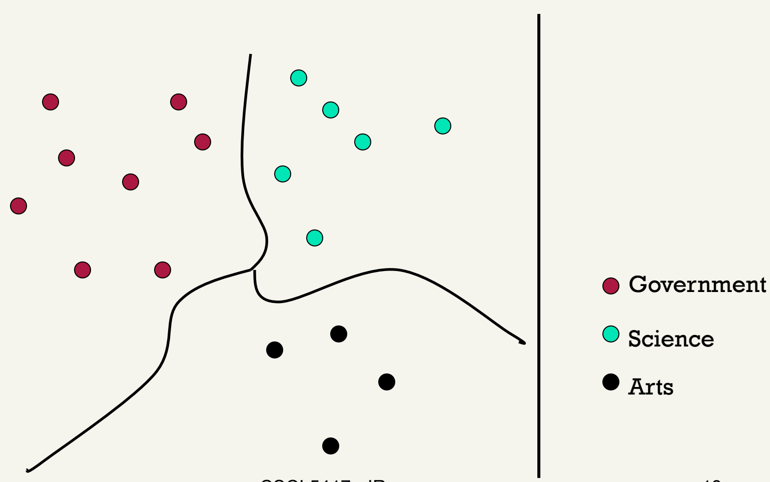
- Each document is a vector, one component for each term in the dictionary
  - Maybe normalize to unit length
- High-dimensional vector space
  - Terms are axes
  - 10,000+ dimensions, or even 100,000+
  - Document vectors define points in this space
- Can we classify in this space?

Classification Using Vector Spaces

- Each training document is a vector labeled by its class (or classes)
- Hypothesis: docs of the same class form a contiguous region of space
- All we need is a way to define surfaces to delineate classes in space

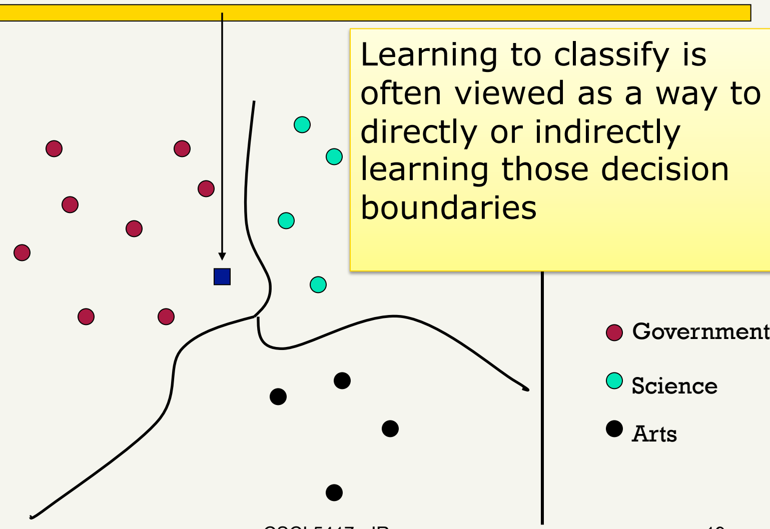
10/17/11 CSCI 5417 - IR 17

Classes in a Vector Space



10/17/11 CSCI 5417 - IR 18

Test Document = Government



Learning to classify is often viewed as a way to directly or indirectly learning those decision boundaries

Nearest-Neighbor Learning

- Learning is just storing the representations of the training examples in *D*.
- Testing instance *x*:
  - Compute similarity between *x* and all examples in *D*.
  - Assign *x* the category of the most similar example in *D*.
- Nearest neighbor learning does not explicitly compute a generalization or category prototypes
- Also called:
  - Case-based learning
  - Memory-based learning
  - Lazy learning

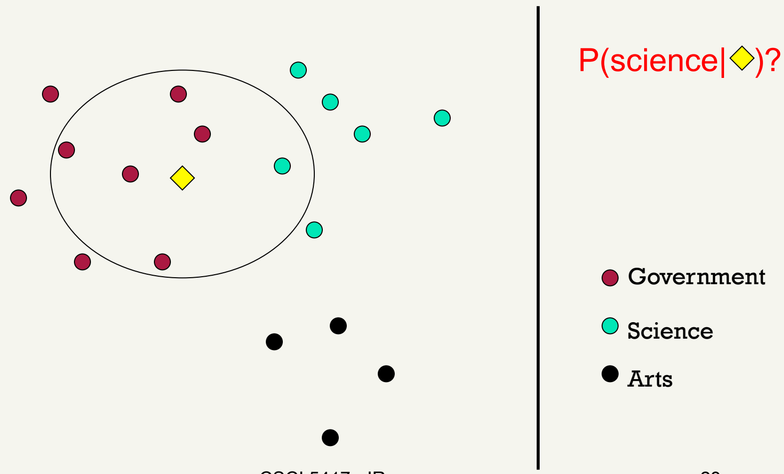
K Nearest-Neighbor

- Using only the closest example to determine the categorization isn't very robust. Errors due to
  - Isolated atypical document
  - Errors in category labels
- More robust alternative is to find the *k* most-similar examples and return the majority category of these *k* examples.
- Value of *k* is typically odd to avoid ties; 3 and 5 are most common.

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 i in N that belong to c
- Estimate P(c|*d*) as i/k
- Choose as class argmax_c P(c|*d*)
  - = majority class

Example: k=6 (6NN)



Similarity Metrics

- Nearest neighbor method depends on a similarity (or distance) metric
- For documents, cosine similarity of tf.idf weighted vectors is typically very effective

Nearest Neighbor with Inverted Index

- Naively finding nearest neighbors requires a linear search through $|D|$ documents in collection
- But if cosine is the similarity metric then determining *k* nearest neighbors is the same as determining the *k* best retrievals using the test document as a query to a database of training documents.
- So just use standard vector space inverted index methods to find the *k* nearest neighbors.
- Testing Time: $O(B|V_t|)$ where B is the average number of training documents in which a test-document word appears.
  - Typically $B << |D|$

Preview HW 3

Classification of our medical abstracts...

In particular, assignment of MeSH terms to documents

Medical Subject Headings

MeSH Terms

```
.I 7
.U
87049094
.S
Am J Emerg Med 8703; 4(6):516-9
.M
Adult; Carbon Monoxide Poisoning/CO/*TH; Female; Human; Labor;
Pregnancy; Pregnancy Complications/*TH; Pregnancy Trimester, Third;
Respiration, Artificial; Respiratory Distress Syndrome, Adult/ET/*TH.
.T
Acute carbon monoxide poisoning during pregnancy.
.P
JOURNAL ARTICLE.
.W
The course of a pregnant patient at term who was acutely exposed to
carbon monoxide is described. A review of the fetal-maternal
carboxyhemoglobin relationships and the differences in fetal
oxyhemoglobin physiology are used to explain the recommendation that
pregnant women with carbon monoxide poisoning should receive 100%
oxygen therapy for up to five times longer than is otherwise
necessary. The role of hyperbaric oxygen therapy is considered.
```


Questions?

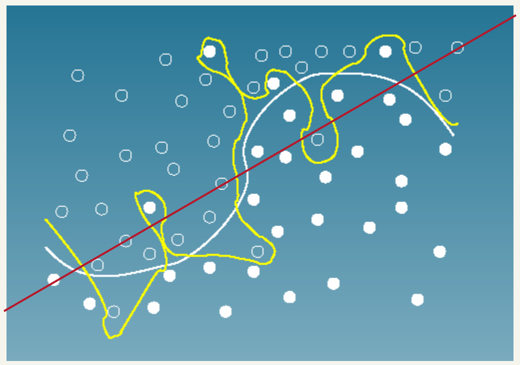
Questions

- Will the settings/approaches/tweeks used in the last HW work for this one?
- What evaluation metric will we be using for this HW?
- Given that, how should we go about doing development?
- How exactly are we supposed to use the MeSH terms? What are all those slashes and *'s?

kNN: Discussion

- No feature selection necessary
- Scales well with large number of classes
  - Don't need to train *n* classifiers for *n* classes
- Scores can be hard to convert to probabilities
- No training necessary
  - Sort of… still need to figure out tf-idf, stemming, stop-lists, etc. All that requires tuning which really is training.

Bias vs. Variance: Choosing the correct model capacity



kNN vs. Naive Bayes

- Bias/Variance tradeoff
  - Variance ≈ Capacity
- kNN has high variance and low bias.
  - Infinite memory
- NB has low variance and high bias.
- Consider: Is an object a tree?
  - 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
  - You want the middle ground

Readings and Next time

- Classification and naïve Bayes
  - Chapter 13
- Vector space classification
  - Chapter 14
- Machine learning
  - Chapter 15

Projects

- Can I use Lucene?
  - Yes
- Do I have to use Lucene
  - No
- Can I do something to extend Lucene
  - Yes but make sure it isn't already there
- Can I try a standard task (bake-off, shared task, etc.)
  - Yes
- Can I do something where it isn't obvious how to evaluate?
  - Yes

Projects

- Can I do something w/ Twitter?
  - Yes
- FaceBook?
  - Yes, but that might be harder
- Can I combine a project with another course project
  - Yes. But it better be good.