Information Retrieval and Web Search

Text classification

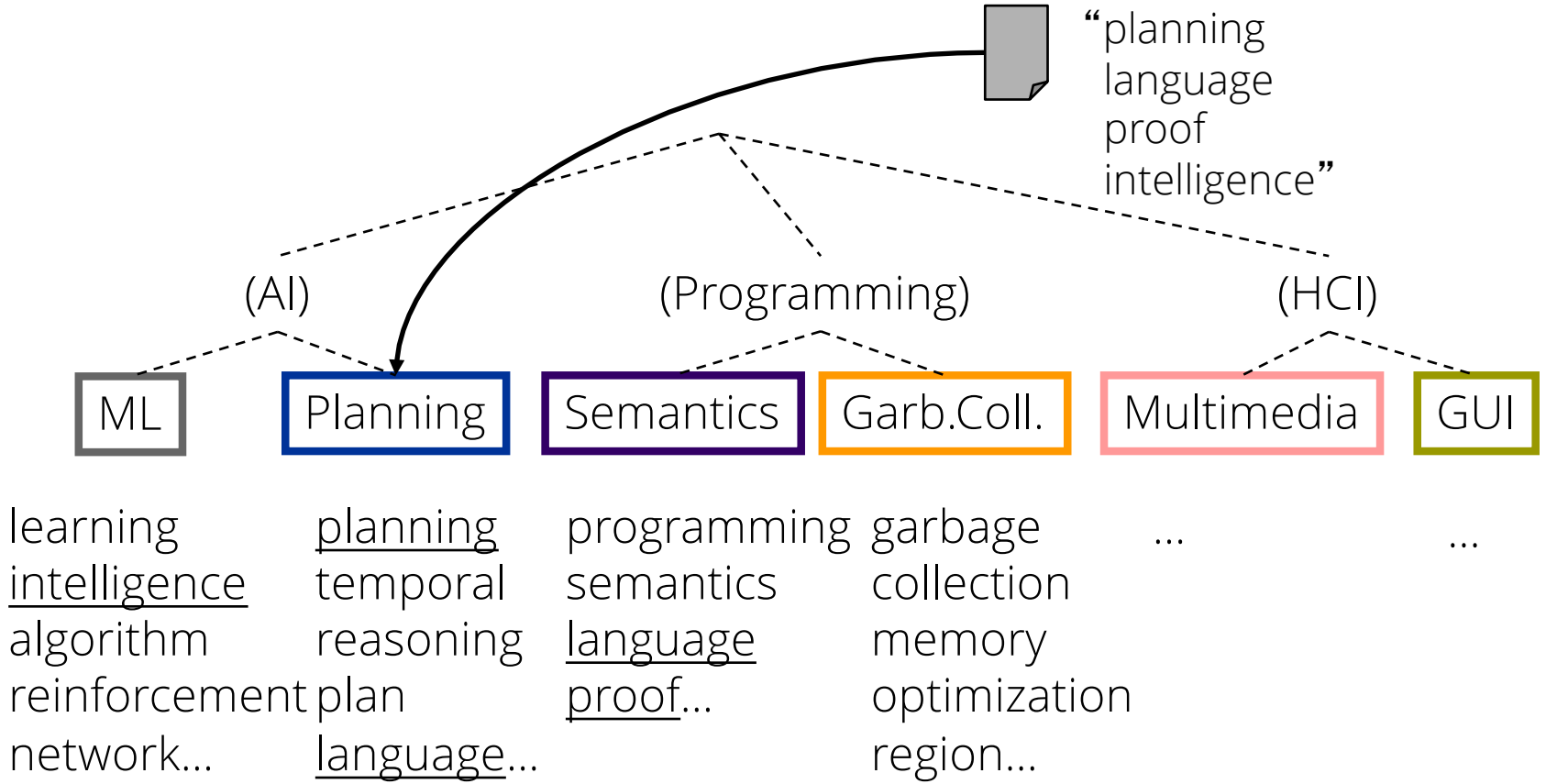
Instructor: Rada Mihalcea

Text Classification

- Also known as text categorization
- Given:
  - A description of an instance, $x \in X$, where X is the *instance language* or *instance space*.
    - Issue: how to represent text documents.
  - 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 ("classifiers").

Categories



(Note: in real life the hierarchies are often deep. Also, you may get papers on e.g., ML approaches to Garb. Coll.)

Text Classification Applications

- Labels are most often topics such as Yahoo-categories

  e.g., "finance," "sports," "news>world>asia>business"
- Labels may be genres

  e.g., "editorials" "movie-reviews" "news"
- Labels may be opinion

  e.g., "like", "hate", "neutral"
- Labels may be domain-specific binary

  e.g., "interesting-to-me" : "not-interesting-to-me"

  e.g., "spam" : "not-spam"

  e.g., "is a toner cartridge ad" : "isn't"

- As a discipline, computer science spans a range of topics from theoretical studies of algorithms and the limits of computation to the practical issues of implementing computing systems in hardware and software. The Association for Computing Machinery (ACM), and the IEEE Computer Society (IEEE-CS) identify four areas: *theory of computation*, *algorithms and data structures*, *programming methodology and languages*, and *computer elements and architecture*.

- The notes of the 12-tone scale can be written by their letter names A–G, possibly with a trailing sharp or flat symbol, such as A♯ or B♭ . This is the most common way of specifying a note in English speech or written text.In Northern and Central Europe, the letter system used is slightly different for historical reasons. In these countries' languages, the note called simply B in English (i.e., B♮) is called H, and the note B♭ is named B.

- Hi John,

  I hope you are doing well. Have you found a job after your graduation?

  I was wondering if you could tell me where is the thermal camera that you used for the discomfort experiments? Is it still in Dr. Yong's lab? I had borrowed it from Prof. Doe in the CSE department, and I should eventually return it to him at some point.

- Dear Rada Mihalcea, ICISA (International Conference on Information Science & Applications) has been scheduled on May 6th - 9th, 2014 in Seoul, South Korea. The final paper submission date is **February 28th, 2014** please make sure to submit your paper before this date! With IEEE ICISA will be holding its 5th annual conference. ICISA 2014 paper submission system is now open and ready for you to upload your paper.

- Spring at Wellington ! Or was it summer ?? Oh who cares ... summer felt like winter the last time around especially when things ahem ... didnt quite worked out as planned . (Grr) Taken during the Tulip Week that took place 2 years ago . They were huge , gorgeous and colourful . Spring is in the air !!! Ah!

- Vegas + other travels . Kev has been to the southern United States several times . Did some sound engineering thing in California and interned for Trent R . in N ' Orleans in like 2002 Spent alot of time in the southern United States SO , anyways he wants to visit New Orleans , Phoenix again (spent some time there but I have a bad memory) and a bunch of places in the south . We 're commiting to Vegas .

- My best friend is very funny. He's always making jokes and making people laugh. We're such good friends because he can also be very serious when it comes to emotions and relationships. It keeps me from getting too relaxed and making a mistake like taking advantage of our friendship or not making an effort to keep it going. He's a pretty fragile person, and although it can be hard to keep him happy sometimes, it's all the more rewarding.

- My best friend never gives me a hard time about anything. If we don't see each other or talk to each other for a while, it's not like anyone's mad. We could not see each other for years, and if we met up it would be like nothing happened. A lot of people in life can make you feel like your being judged, and most of the time make you feel that what your doing isn't good enough. My best friend is one of the few people I don't feel that way around.

Classification Methods (overview)

- Manual classification
  - Used by Yahoo!, Looksmart, about.com, ODP, Medline
  - Very accurate when job is done by experts
  - Consistent when the problem size and team is small
  - Difficult and expensive to scale
- Automatic document classification with hand-coded rule-based systems
  - Used in specialized searches
  - Assign category if document contains a given boolean combination of words
  - Some commercial systems (e.g., Lexis Nexis) have complex query languages (similar to IR query languages)
  - Accuracy is often very high if a query has been carefully refined over time by a subject expert
  - Building and maintaining these queries is expensive

Classification Methods (overview)

- Supervised learning of document-label assignment function
  - Most new systems rely on machine learning
    - k-Nearest Neighbors (simple, powerful)
    - Naive Bayes (simple, common method)
    - Support-vector machines (newer, more powerful)
    - …
    - The most recent learning method?
  - No free lunch: requires hand-classified training data
  - But can be built (and refined) by non-experts

Text Classification Attributes

- Representations of text are very high dimensional (one feature for each word).
- For most text classification tasks, there are many irrelevant and many relevant features.
- Methods that combine evidence from many or all features (e.g., naive Bayes, kNN, neural-nets) tend to work better than the ones that try to isolate just a few relevant features (e.g., decision trees)

Learning Methods

- Broad classification:
  - Eager
  - Lazy
- Algorithms:
  - Naïve Bayes
  - Decision Trees
  - Nearest Neighbor
  - Rocchio
  - Neural Networks
  - Regression (only with numerical output)
  - Support Vector Machine
  - Random Forests

Naïve Bayes

Naïve Bayes Text Classification

- Learning and classification method based on probability theory
- Bayes theorem plays a critical role in probabilistic learning and classification
- Build a *generative model* that approximates how data is produced
- Uses *prior* probability of each category given no information about an item
- Classification produces a *posterior* probability distribution over the possible categories given a description of an item

Bayes' Rule

$$P(C,X) = P(C \mid X)P(X) = P(X \mid C)P(C)$$

$$\boxed{P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)}}$$

Naive Bayes Classifiers

Task: Classify a new instance based on a tuple of attribute values

$$\langle x_1, x_2, \ldots, x_n \rangle$$

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

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

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

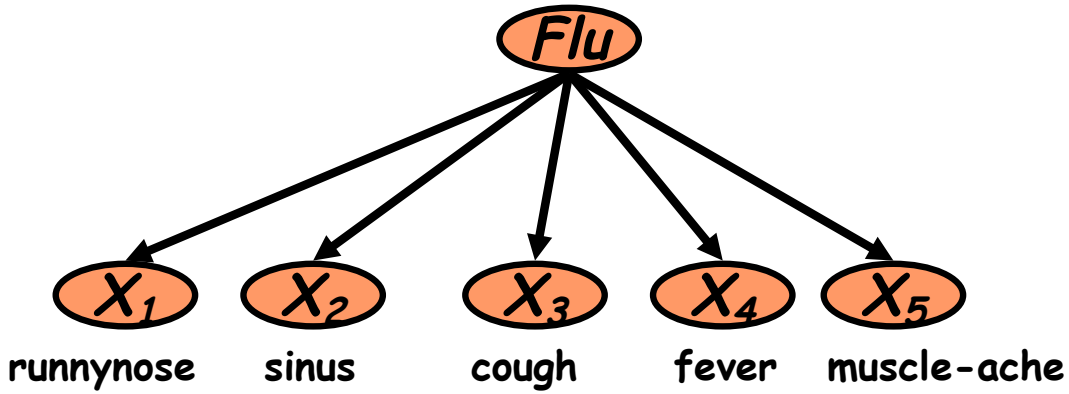
Naïve Bayes Classifier: Assumptions

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

Conditional Independence Assumption:

$\Rightarrow$ Assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities.

The Naïve Bayes Classifier



- Conditional Independence Assumption: features are independent of each other given the class:

$$P(X_1,\ldots,X_5 \mid C) = P(X_1 \mid C) \bullet P(X_2 \mid C) \bullet \cdots \bullet P(X_5 \mid C)$$

Learning the Model

- Common practice: maximum likelihood
  - simply use the frequencies in the data

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

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

$$c = \arg\max_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

Smoothing

- To avoid zero-counts

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

# of values of X_i (vocabulary)

Naïve Bayes in Practice

- From training corpus, extract *Vocabulary*
- Calculate required $P(c_j)$ and $P(x_k \mid c_j)$ terms
  - For each c_j in C do
    - $docs_j \leftarrow$ subset of documents for which the target class is c_j
    - $$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$
    - $Text_j \leftarrow$ single document containing all $docs_j$
    - for each word x_k in *Vocabulary*
      - $n_k \leftarrow$ number of occurrences of x_k in $Text_j$
      - $n \leftarrow$ number of words in $Text_j$

$$P(x_k \mid c_j) \leftarrow \frac{n_k + 1}{n + |Vocabulary|}$$

$$c = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod P(x_k \mid c_j)$$

  - Return

Example

- Doc1 BIO cell structure growth study
- Doc2 CS computer network study
- Doc 3 CS structure information retrieval computer
- Doc4 BIO biology cell network distribution
- Doc 5 BIO growth structure evolution
- Doc 6 CS structure social network
- Doc 100 ? structure computer network

- Classify Doc100 using Naïve Bayes
- V=12

Time Complexity

- Training Time: $O(|D|L_d + |C||V|))$
  - where L_d is the average length of a document in D.
  - Generally just $O(|D|L_d)$ since usually $|C||V| < |D|L_d$
- Test Time: $O(|C| L_t)$
  - where L_t is the average length of a test document.
  - Very efficient overall, linearly proportional to the time needed to just read in all the data.

Multinomial vs. Bernoulli

- Multinomial

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

- Bernoulli

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

Multinomial: counts total number of occurrences of x_i in c_j
Bernoulli: counts total number of documents in c_j that include x_i

Underflow Prevention

- Multiplying lots of probabilities, which are between 0 and 1 by definition, can result in floating-point underflow.
- Since $\log(xy) = \log(x) + \log(y)$, it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities.

Decision Trees

Decision Trees

- Build a tree, with a feature (word) in each node
- A branch in the tree represents the classification
- Question:
  - Which features to choose first?
  - Feature ordering in the tree can affect the classification of new data
- Answer:
  - Choose the features with the highest information gain

Basic elements of information theory

- How to determine which attribute is the best classifier?
  - Measure the information gain of each attribute
- Entropy characterizes the (im)purity of an arbitrary collection of examples.
  - Given a collection S of positive and negative examples
  - Entropy(S) = - p log p – q log q
  - Entropy is at its maximum when p = q = ½
  - Entropy is at its minimum when p = 1 and q = 0
- Example:
  - S contains 14 examples: 9 positive and 5 negative
  - Entropy(S) = - (9/14) log (9/14) – (5/14) log (5/14) = 0.94
    - log 0 = 0

Basic elements of information theory

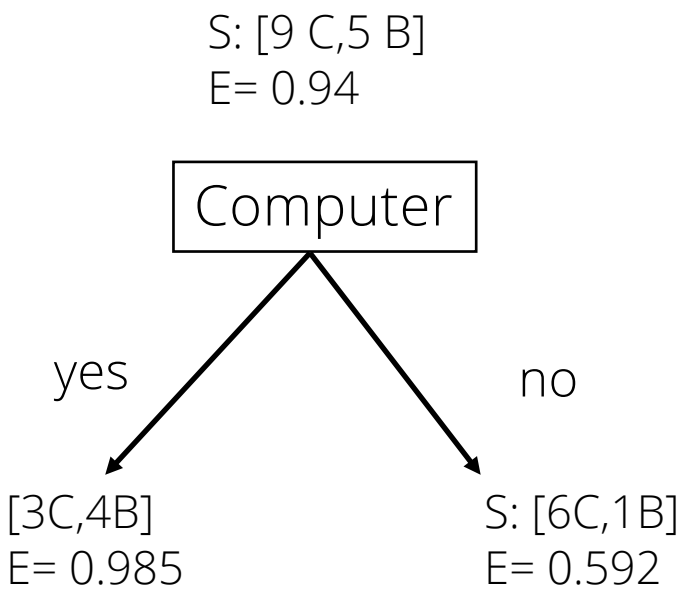
- Information gain
  - Measures the expected reduction in entropy

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

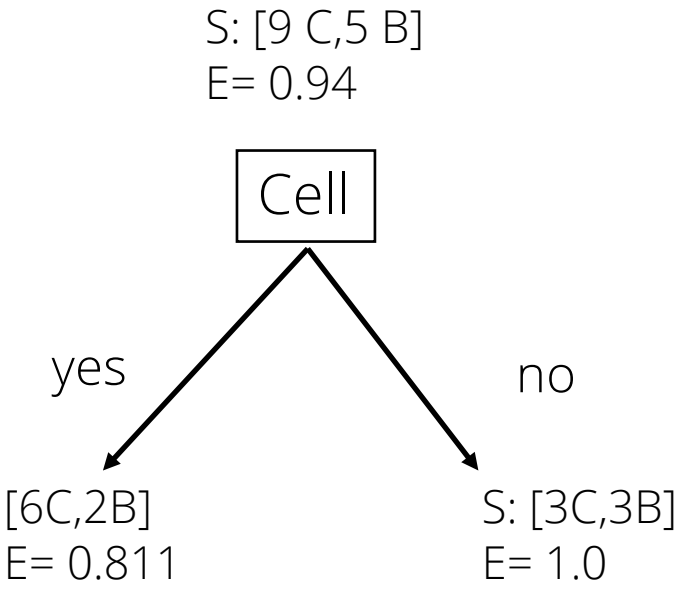
  - Many learning algorithms are making decisions based on information gain

Basic elements of information theory

Which feature is the best classifier?



Gain (S, Computer)=
0.94-7/14*0.985-7/14*0.592=
0.151



Gain (S, Cell)=
0.94-8/14*0.811-6/14*1=
0.048

Example

- Doc1 BIO cell structure growth study
- Doc2 CS computer network study
- Doc 3 CS structure information retrieval computer
- Doc4 BIO biology cell network distribution
- Doc 5 BIO growth structure evolution
- Doc 6 CS structure social network

- What is the information gain for “network”

Decision Trees

- Have the capability of generating rules:
  - IF Computer=yes and Network= yes
  - THEN topic = Computer
- Powerful – it is very hard to do that as a human
  - C4.5 (Quinlan)
  - Integral part of Weka (for Java)

Rocchio Text Classification

Rocchio Text Classification

- Use standard TF/IDF weighted vectors to represent text documents (normalized by maximum term frequency)
- For each category, compute a *prototype* vector by summing the vectors of the training documents in the category
- Assign test documents to the category with the closest prototype vector based on cosine similarity

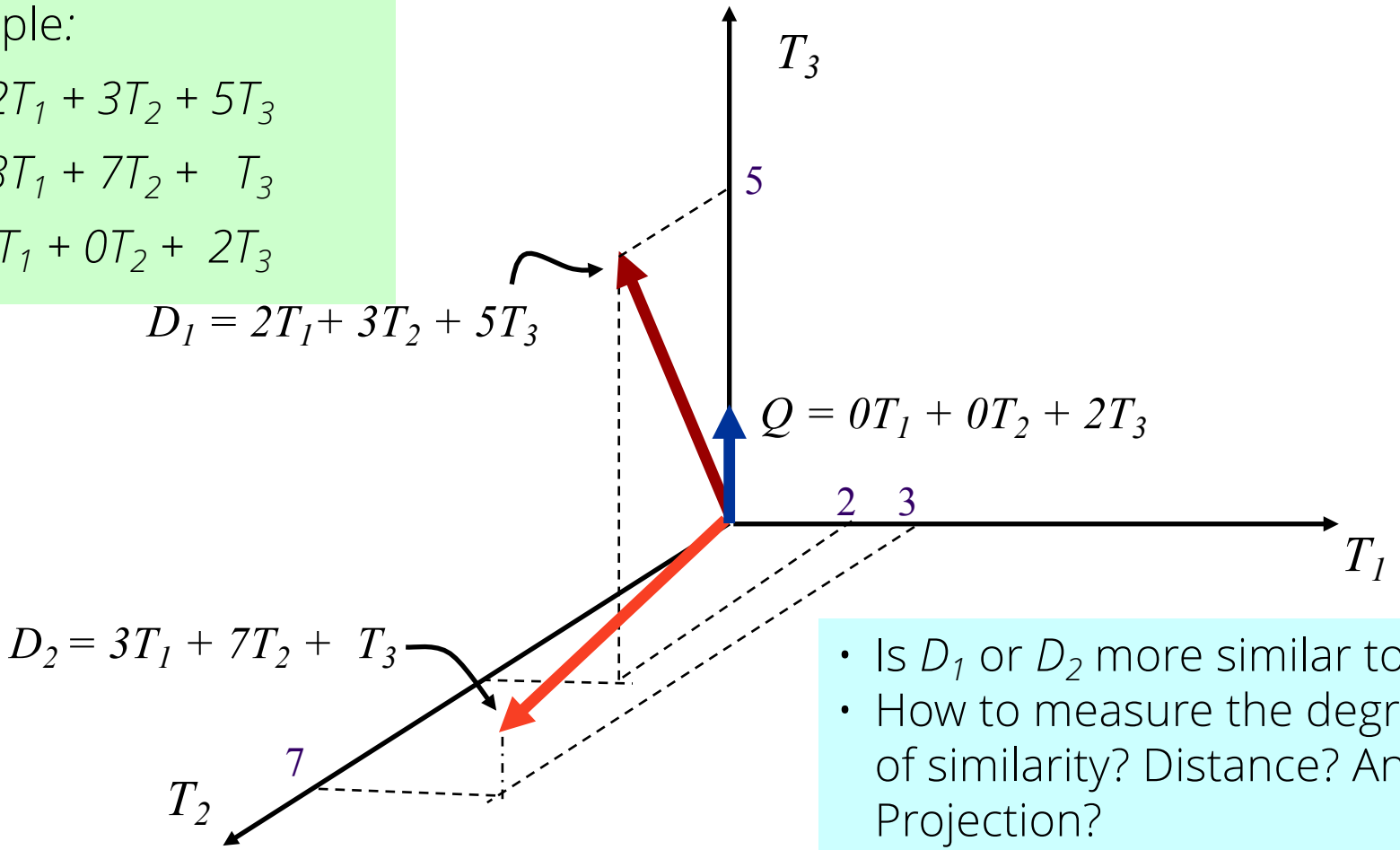
Recap: Vector-space Model

Example:

$D_1 = 2T_1 + 3T_2 + 5T_3$

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = 0T_1 + 0T_2 + 2T_3$



- Is D_1 or D_2 more similar to Q?
- How to measure the degree of similarity? Distance? Angle? Projection?

Recap: Document Collection Representation

- A collection of *n* documents can be represented in the vector space model by a term-document matrix.
- An entry in the matrix corresponds to the "weight" of a term in the document; zero means the term has no significance in the document or it simply doesn't exist in the document.

$$\begin{pmatrix} & T_1 & T_2 & .... & T_t \\ D_1 & w_{11} & w_{21} & ... & w_{t1} \\ D_2 & w_{12} & w_{22} & ... & w_{t2} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ D_n & w_{1n} & w_{2n} & ... & w_{tn} \end{pmatrix}$$

Recap: tf-idf Weighting

- The tf-idf weight of a term is the product of its tf weight and its idf weight.

$$w_{t,d} = \mathrm{tf}_{t,d} \times \log_{10}(N / \mathrm{df}_t)$$

- Increases with the number of occurrences within a document
- Increases with the rarity of the term in the collection

Rocchio Text Classification (Training)

Assume the set of categories is $\{c_1, c_2, \ldots c_k\}$

For i from 1 to k let $\mathbf{p}_i = <0, 0, \ldots, 0>$ *(init. prototype vectors)*

For each training example $<x, c(x)> \in D$

Let $\mathbf{d}$ be the tf.idf term vector for doc x

Let $i = j: (c_j = c(x))$

(sum all the document vectors in c_i to get $\boldsymbol{p}_i$)

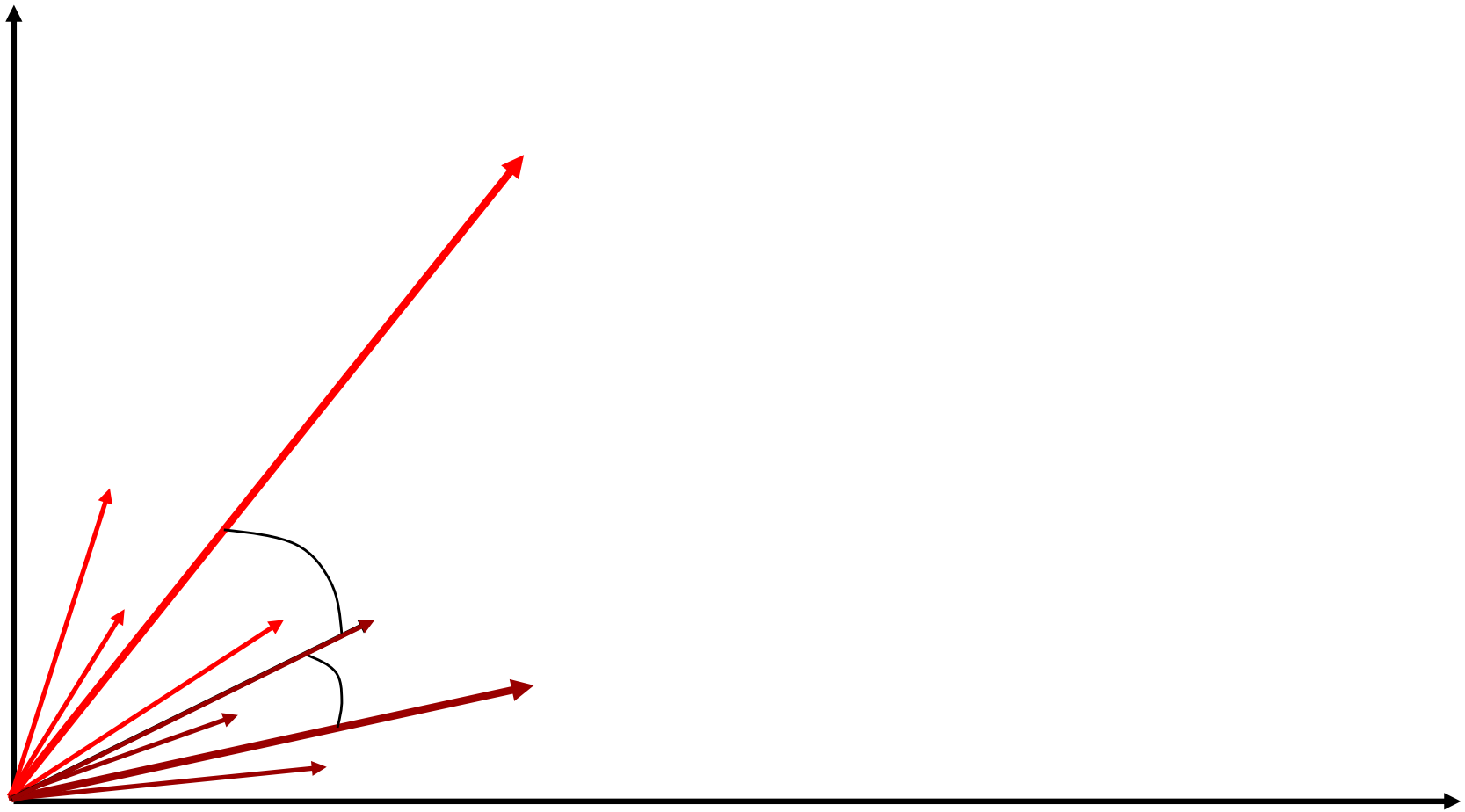
Let $\mathbf{p}_i = \mathbf{p}_i + \mathbf{d}$

One vector per category

Rocchio Text Classification (Test)

Given test document *x*
Let **d** be the tf.idf weighted term vector for *x*
Let *m* = –1 (*init. maximum cosSim*)
For *i* from 1 to *k*:
    (*compute similarity to prototype vector*)
    Let *s* = cosSim(**d**, $\mathbf{p}_i$)
    if *s* > *m*
        let *m* = *s*
        let *r* = c_i (*update most similar class prototype*)
Return class *r*

Illustration of Rocchio Text Categorization



Nearest Neigbor

Documents as Vectors

- Starting points for many learning algorithms
- Represent each document as a vector of weights, with length equal to vocabulary

$$\begin{array}{c|cccc|c} & T_1 & T_2 & \ldots & T_t & \\ D_1 & w_{11} & w_{21} & \ldots & w_{t1} & L_1 \\ D_2 & w_{12} & w_{22} & \ldots & w_{t2} & L_2 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ D_n & w_{1n} & w_{2n} & \ldots & w_{tn} & L_n \end{array}$$

- Augment this matrix with the class assigned to each document, $L_1..L_n$ in $\{c_1, c_2,...c_k\}$
- Can use this representation as input for a lot of ML algorithms (see Weka, Scikit, SVMLight, etc.)

Nearest Neighbour

- Classify a new instance based on the distance between current example and all examples in training
- Typical similarity measure for nearest neighbor:

$$d(X,Y) = \sqrt{\sum_{r=1}^{n} (x_r - y_r)^2}$$

What measure is this?

- Other similarity measures are also possible
  - Cosine, city distance, inner product, Jaccard

Classification in Nearest Neigbor

- Choose the category of the closest training example(s)
- 1-NN: use the category of the closest training examples
- 3-NN: do majority voting over the categories of the three closest training examples
- k-NN: k closest training examples

- Nearest Neigbor: lazy or eager?

Evaluation of Text Classification

Evaluating Text Classification

- Evaluation must be done on test data that are independent of the training data (usually a disjoint set of instances).
- *Classification accuracy*: c/n where n is the total number of test instances and c is the number of test instances correctly classified by the system.
- Precision / recall with respect to a given class
- Results can vary based on sampling error due to different training and test sets.
- N-fold cross-validation: Average results over multiple training and test sets (splits of the overall data) for the best results.

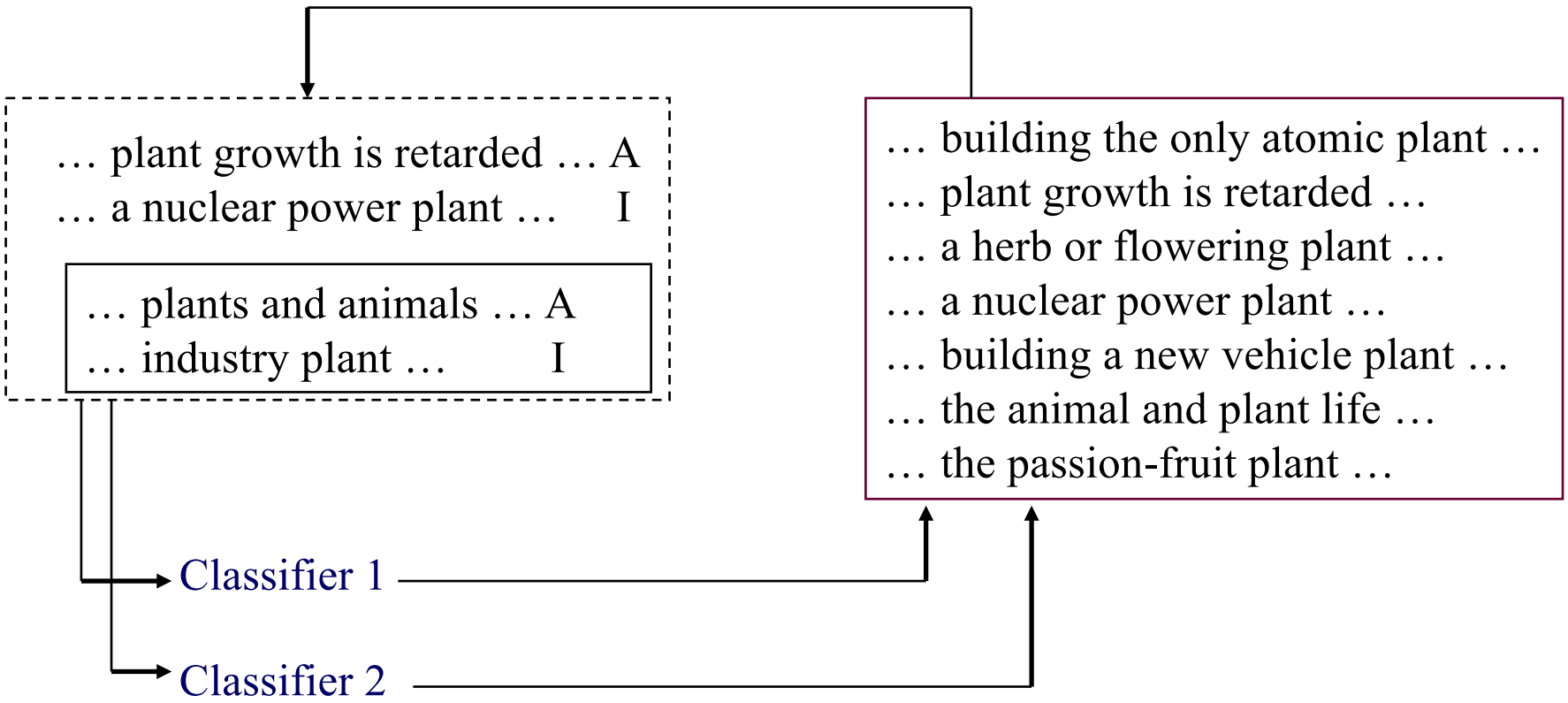
Learning Curves

- In practice, labeled data is usually rare and expensive.
- Would like to know how performance varies with the number of training instances.
- *Learning curves* plot classification accuracy on independent test data (*Y* axis) versus number of training examples (*X* axis).
- Want learning curves averaged over multiple trials.
- Use *N*-fold cross validation to generate *N* full training and test sets.
- Alternatively, use leave-one-out cross-validation, to train on all examples minus one, and test on the remaining one

Bootstrapping

Bootstrapping

- Use a few seed labeled documents, and a lot of raw documents
- Automatically classify documents and add the most confidently labeled ones to the training set
- Repeat
  - Automatically grow the training set
- Bootstrapping recipe:
  - Ingredients
    - (Some) labeled data
    - (Large amounts of) unlabeled data
    - (One or more) basic classifiers
  - Output
    - Classifier that improves over the basic classifiers



Assume two classes: A[griculture] and I[ndustry]

Co-training / Self-training

- A set L of labeled training examples
- A set U of unlabeled examples
- Classifiers C_i

- 1. Create a pool of examples U'
  - choose P random examples from U
- 2. Loop for I iterations
  - Train C_i on L and label U'
  - Select G most confident examples and add to L
    - maintain distribution in L
  - Refill U' with examples from U
    - keep U' at constant size P

Co-training / Self-training

Co-training

- (Blum and Mitchell 1998)
- Two classifiers

  independent views

  [independence condition can be relaxed]

  Example for the classification of a web page

Self-training

- (Nigam and Ghani 2000)
- One single classifier

  Retrain on its own output

  Use confidence value to select examples to add to training data