Text Categorization

CSE 454

Administrivia

- Mailing List
- Groups for PS1
- Questions on PS1?
  - See discussion & pseudocode for naive Bayes in "Information Retrieval" by Manning, Raghavan, and Schutze
  - Good textbook and available online for free

For Next Class

- Reading for Thurs
  - **Mercator: A Scalable, Extensible Web Crawler**,
  - by Allan Heydon & Mark Najork,
- Work on PS1
- Think about projects

Class Overview

- Other Cool Stuff
- Query processing
- Content Analysis
- Indexing
- Crawling
- Document Layer
- Network Layer

Class Overview

- Other Cool Stuff
- Query processing
- **Content Analysis**
- Indexing
- Crawling
- Document Layer
- Network Layer

Class Overview

- Other Cool Stuff
- Query processing
- Content Analysis
- Indexing
- Crawling
- **Document Layer**
- **Network Layer**

} Next Classes

Categorization

- Given:
  - A **description of an instance**, $x \in X$, where X is the *instance language* or *instance space*.
  - 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.

Sample Category Learning Problem

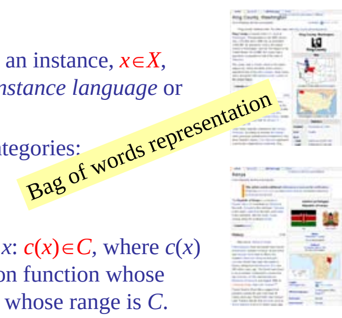
- Instance language: <size, color, shape>
  - size ∈ {small, medium, large}
  - color ∈ {red, blue, green}
  - shape ∈ {square, circle, triangle}
- C = {**positive**, negative}
- D:

Example	Size	Color	Shape	Category
1	**small**	**red**	**circle**	**positive**
2	**large**	**red**	**circle**	**positive**
3	small	red	triangle	negative
4	large	blue	circle	negative

Another Example: County *vs.* Country?

Example: County *vs.* Country?

- Given:
  - A description of an instance, $x \in X$, where X is the *instance language* or *instance space*.
  - 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.



Text Categorization

- Assigning documents to a fixed set of categories, *e.g.*
- Web pages
  - Yahoo-like classification
- What else?
- Email messages
  - Spam filtering
  - Prioritizing
  - Folderizing
- News articles
  - Personalized newspaper
- Web Ranking
  - Is page related to selling something?

Procedural Classification

- Approach:
  - Write a procedure to determine a document's class
  - E.g., Spam?

Learning for Text Categorization

- Hard to construct text categorization functions.
- Learning Algorithms:
  - **Bayesian (naïve)**
  - Neural network
  - Relevance Feedback (Rocchio)
  - Rule based (C4.5, Ripper, Slipper)
  - Nearest Neighbor (case based)
  - Support Vector Machines (SVM)

Applications of ML

- Credit card fraud
- Product placement / consumer behavior
- Recommender systems
- Speech recognition

Most mature & successful area of AI

Learning for Categorization

- A ***training example*** is an instance $x \in X$, paired with its correct category $c(x)$: **<x, c(x)>** for an unknown categorization function, *c*.
- Given a set of training examples, *D*.

{<  , county>, <  , country>,…

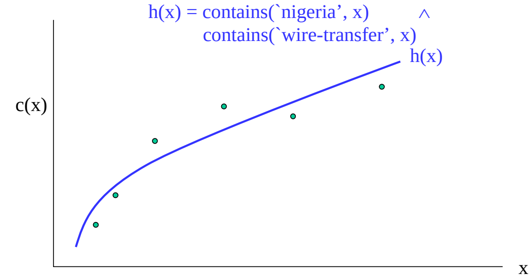
- Find a hypothesized categorization function, $h(x)$, such that: $\forall < x, c(x) > \in D : h(x) = c(x)$

Consistency

ML = Function Approximation

May not be any perfect fit
Classification ~ discrete functions

h(x) = contains(`nigeria', x) ^ contains(`wire-transfer', x)



Generalization

- Hypotheses must ***generalize*** to correctly classify instances not in the training data.
- Simply memorizing training examples is a consistent hypothesis ***that does not generalize***.

Why is Learning Possible?

Experience alone never justifies any conclusion about any unseen instance.

Learning occurs when
PREJUDICE meets DATA!

Learning a "Frobnitz"

Bias

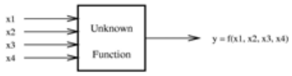
- The nice word for prejudice is “bias”.
- What kind of hypotheses will you ***consider***?
  - What is allowable ***range*** of functions you use when approximating?
- What kind of hypotheses do you ***prefer***?

Some Typical Biases

- Occam’s razor
  “It is needless to do more when less will suffice”
  – William of Occam,
  died 1349 of the Black plague
- MDL – Minimum description length
- Concepts can be approximated by
- ... **conjunctions** of predicates
  ... by **linear** functions
  ... by **short** decision trees

Frobnitz?

A Learning Problem



Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Hypothesis Spaces

- **Complete Ignorance.** There are $2^{16} = 65536$ possible boolean functions over four input features. We can't figure out which one is correct until we've seen every possible input-output pair. After 7 examples, we still have 2^9 possibilities.

x_1	x_2	x_3	x_4	y
0	0	0	0	?
0	0	0	1	?
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	?
1	0	0	0	?
1	0	0	1	1
1	0	1	0	?
1	0	1	1	?
1	1	0	0	0
1	1	0	1	?
1	1	1	0	?
1	1	1	1	?

Terminology

- **Training example.** An example of the form $\langle \mathbf{x}, f(\mathbf{x}) \rangle$.
- **Target function (target concept).** The true function f.
- **Hypothesis**. A proposed function h believed to be similar to f.
- **Concept**. A boolean function. Examples for which $f(\mathbf{x}) = 1$ are called **positive examples** or **positive instances** of the concept. Examples for which $f(\mathbf{x}) = 0$ are called **negative examples** or **negative instances.**
- **Classifier**. A discrete-valued function. The possible values $f(\mathbf{x}) \in \{1, \ldots, K\}$ are called the **classes** or **class labels**.
- **Hypothesis Space**. The space of all hypotheses that can, in principle, be output by a learning algorithm.
- **Version Space**. The space of all hypotheses in the hypothesis space that have not yet been ruled out by a training example.

General Learning Issues

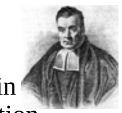
- Many hypotheses consistent with the training data.
- **Bias**
  - Any criteria other than consistency with the training data that is used to select a hypothesis.
- Classification accuracy
  - % of instances classified correctly
  - (Measured on independent test data.)
- Training time
  - Efficiency of training algorithm
- Testing time
  - Efficiency of subsequent classification

Two Strategies for ML

- Restriction bias: use prior knowledge to specify a restricted hypothesis space.
  - Naïve Bayes Classifier
- Preference bias: use a broad hypothesis space, but impose an ordering on the hypotheses.
  - Decision trees.

Bayesian Methods

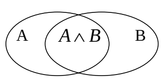
- Learning and classification methods based on probability theory.
  - Uses ***prior*** probability of each category
    Given no information about an item.
  - Produces a ***posterior*** probability distribution over possible categories
    Given a description of an item.



- Bayes theorem plays a critical role in probabilistic learning and classification.

Axioms of Probability Theory

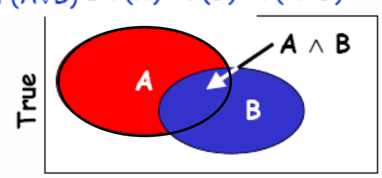
- All probabilities between 0 and 1
  $0 \le P(A) \le 1$
- Probability of truth and falsity
  P(true) = 1   P(false) = 0.
- The probability of disjunction is:
  $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$



Probability: Simple & Logical

- The definitions imply that certain logically related events must have related probabilities

E.g. $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$

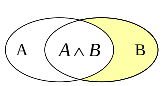


de Finetti (1931): an agent who bets according to probabilities that violate these axioms can be forced to bet so as to lose money regardless of outcome.

Conditional Probability

- P(*A* | *B*) is the probability of *A* given *B*
- Assumes:
  - *B* is all and only information known.
- Defined by:

$$P(A \mid B) = \frac{P(A \wedge B)}{P(B)}$$



Independence

- *A* and *B* are *independent* iff:

$$P(A \mid B) = P(A)$$
$$P(B \mid A) = P(B)$$

These constraints are logically equivalent

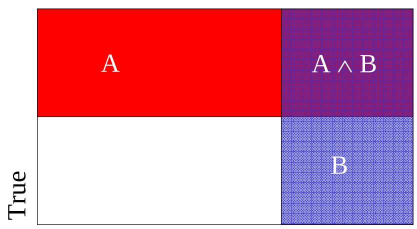
- Therefore, if *A* and *B* are independent:

$$P(A \mid B) = \frac{P(A \wedge B)}{P(B)} = P(A)$$

$$P(A \wedge B) = P(A)P(B)$$

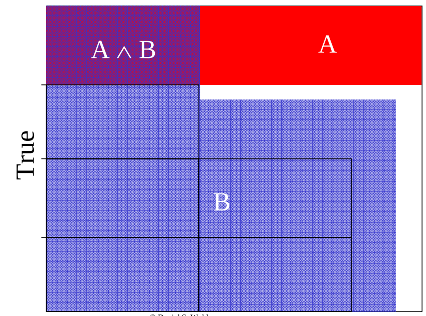
Independence

$P(A \wedge B) = P(A)P(B)$



Independence Is Rare

A&B *not* independent, since $P(A|B) \neq P(A)$



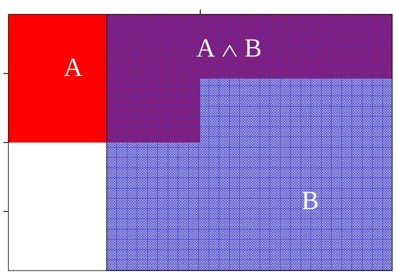
$P(A) = 25\%$

$P(B) = 80\%$

$P(A|B) \leq 1/7$

Conditional Independence

Are A & B independent? $P(A|B) \leq P(A)$



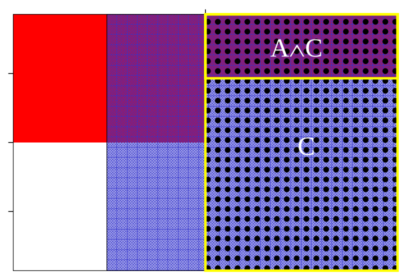
$P(A)=(.25+.5)/2$
$= .375$

$P(B)= .75$

$P(A|B)=(.25+.25+.5)/3$
$=.3333$

A, B Conditionally Independent Given C

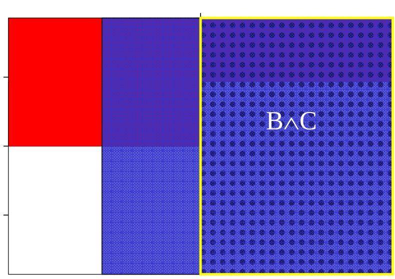
$P(A|B,C) = P(A|C)$ C = spots



$P(A|C) = .25$

A, B Conditionally Independent Given C

$P(A|B,C) = P(A|C)$ C = spots

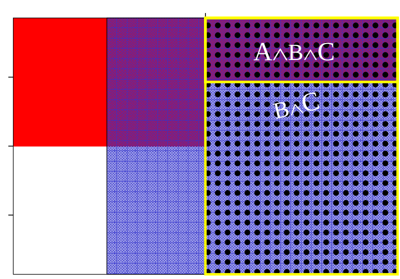


$P(A|C) = .25$

$P(B|C) = 1.0$

A, B Conditionally Independent Given C

$P(A|B,C) = P(A|C)$ C = spots



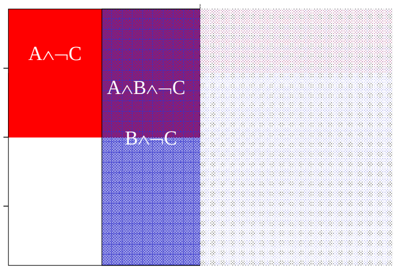
$P(A|C) = .25$

$P(B|C) = 1.0$

$P(A|B,C)=.25$

A, B Conditionally Independent Given C

P(A|B,C) = P(A|C)   C = spots

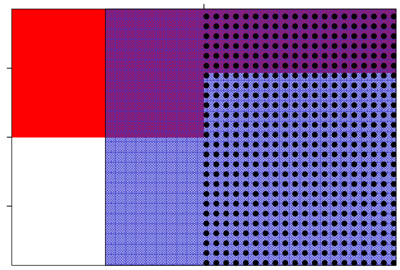


P(A|C) =.25
P(B|C) = 1.0
P(A|B,C)=.25

P(A|¬C) =.5
P(B|¬C) = .5
P(A|B,¬C)=.5

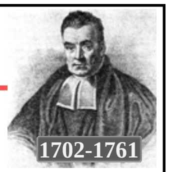
Conditional Independence = The Next Best Thing to Independence

A, B Conditionally Independent Given C



P(A|B,C) = P(A|C)
= P(A|C)

Bayes Theorem



$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}$$

Simple proof from definition of conditional probability:

$$P(H \mid E) = \frac{P(H \wedge E)}{P(E)}$$ (Def. cond. prob.)

$$\frac{P(H \wedge E)}{P(H)} = P(E \mid H)$$ (Def. cond. prob.)

$$P(H \wedge E) = P(E \mid H)P(H)$$ (Mult both sides of 2 by P(H).)

QED:

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}$$ (Substitute 3 in 1.)

Bayesian Categorization

- Let set of categories be $\{c_1, c_2, \ldots c_n\}$
- Let E be description of an instance.
- Determine category of E by determining for each c_i

$$P(c_i \mid E) = \frac{P(c_i)P(E \mid c_i)}{P(E)}$$

- P(E) can be ignored since is factor ∀ categories

$$P(c_i \mid E) \sim P(c_i)P(E \mid c_i)$$

Bayesian Categorization

$$P(c_i \mid E) \sim P(c_i)P(E \mid c_i)$$

- Need to know:
  - Priors: P(c_i)
  - Conditionals: P(E | c_i)

Problem!

- P(c_i) are easily estimated from data.
  - If n_i of the examples in D are in c_i,then P(c_i) = n_i / |D|
- Assume instance is a conjunction of binary features:

$$E = e_1 \wedge e_2 \wedge \cdots \wedge e_m$$

- Too many possible instances (exponential in m) to estimate all P(E | c_i)

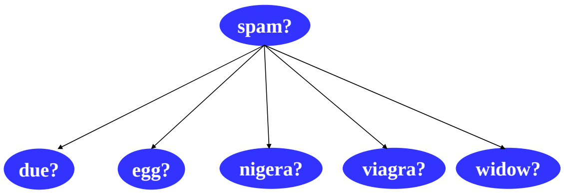
Naïve Bayesian Motivation

- Problem: Too many possible instances to estimate all P(E | c_i) **(exp in *m*)**
- Assume features of an instance are **conditionally independent** given the category (c_i)

$$P(E \mid c_i) = P(e_1 \wedge e_2 \wedge \cdots \wedge e_m \mid c_i) = \prod_{j=1}^{m} P(e_j \mid c_i)$$

- Now we only need to know P(e_j | c_i) for each feature and category.

Conditional Independence??



P(nigeria | spam) = P(nigeria | spam, widow)

P(nigeria | spam) = P(nigeria | spam, viagra)

Naïve Bayes Example

- C = {allergy, cold, well}
- e_1 = sneeze; e_2 = cough; e_3 = fever
- E = {sneeze, cough, ¬fever}

Prob	Well	Cold	Allergy
$P(c_i)$	0.9	0.05	0.05
$P(\text{sneeze}|c_i)$	0.1	0.9	0.9
$P(\text{cough}|c_i)$	0.1	0.8	0.7
$P(\text{fever}|c_i)$	0.01	0.7	0.4

Naïve Bayes Example (cont.)

Probability	Well	Cold	Allergy
$P(c_i)$	0.9	0.05	0.05
$P(\text{sneeze} \mid c_i)$	0.1	0.9	0.9
$P(\text{cough} \mid c_i)$	0.1	0.8	0.7
$P(\text{fever} \mid c_i)$	0.01	0.7	0.4

E={sneeze, cough, ¬fever}

P(well | E) = (0.9)(0.1)(0.1)(0.99)/P(E)=0.0089/P(E)
P(cold | E) = (0.05)(0.9)(0.8)(0.3)/P(E)=0.01/P(E)
P(allergy | E) = (0.05)(0.9)(0.7)(0.6)/P(E)=0.019/P(E)

Most probable category: allergy
P(E) = 0.089 + 0.01 + 0.019 = 0.0379
P(well | E) = 0.23
P(cold | E) = 0.26
P(allergy | E) = 0.50

Estimating Probabilities

- Normally, probabilities are estimated based on observed frequencies in the training data.
- If D contains n_i examples in category c_i, and n_{ij} of these n_i examples contains feature e_j, then:

$$P(e_j \mid c_i) = \frac{n_{ij}}{n_i}$$

- However, estimating such probabilities from small training sets is error-prone.
- If due only to chance, a rare feature, e_k, is always false in the training data, $\forall c_i$:$P(e_k \mid c_i) = 0$.
- If e_k then occurs in a test example, E, the result is that $\forall c_i$: $P(E \mid c_i) = 0$ and $\forall c_i$: $P(c_i \mid E) = 0$

Smoothing

- To account for estimation from small samples, probability estimates are adjusted or *smoothed*.
- **Laplace smoothing** using an *m*-estimate assumes that each feature is given a prior probability, *p*, that is assumed to have been previously observed in a "virtual" sample of size *m*.

$$P(e_j \mid c_i) = \frac{n_{ij} + mp}{n_i + m} \quad = (n_{ij} + 1) / (n_i + 2)$$

- For binary features, *p* is simply assumed to be 0.5.

Naïve Bayes for Text

- Modeled as generating a bag of words for a document in a given category by repeatedly sampling with replacement from a vocabulary $V = \{w_1, w_2, \ldots w_m\}$ based on the probabilities $P(w_j \mid c_i)$.
- Smooth probability estimates with Laplace *m*-estimates assuming a uniform distribution over all words ($p = 1/|V|$) and $m = |V|$
  - Equivalent to a virtual sample of seeing each word in each category exactly once.

Text Naïve Bayes Algorithm (Train)

Let V be the vocabulary of all words in the documents in D
For each category $c_i \in C$
  Let D_i be the subset of documents in D in category c_i
  $P(c_i) = |D_i| / |D|$
  Let T_i be the concatenation of all the documents in D_i
  Let n_i be the total number of word occurrences in T_i
  For each word $w_j \in V$
    Let n_{ij} be the number of occurrences of w_j in T_i
    Let $P(w_i \mid c_i) = (n_{ij} + 1) / (n_i + |V|)$

Text Naïve Bayes Algorithm (Test)

Given a test document X
Let n be the number of word occurrences in X
Return the category:

$$\underset{c_i \in C}{\operatorname{argmax}} P(c_i) \prod_{i=1}^{n} P(a_i \mid c_i)$$

where a_i is the word occurring the ith position in X

Naïve Bayes Time Complexity

- Training Time: $O(|D|L_d + |C||V|))$
  where L_d is the average length of a document in D.
  - Assumes V and all D_i , n_i, and n_{ij} pre-computed in $O(|D|L_d)$ time during one pass through all of the data.
  - 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.

Easy to Implement

- But…
- If you do… it probably won't work…

Probabilities: Important Detail!

- $P(\text{spam} \mid E_1 \ldots E_n) = \prod_i P(\text{spam} \mid E_i)$

Any more potential problems here?

- We are multiplying lots of small numbers
  Danger of underflow!
  - $0.5^{57} = 7$ E -18
- Solution? Use logs and add!
  - $p_1 * p_2 = e^{\log(p1)+\log(p2)}$
  - Always keep in log form

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.
- Class with highest final un-normalized log probability score is still the most probable.

Naïve Bayes Posterior Probabilities

- Classification results of naïve Bayes
  - I.e. the class with maximum posterior probability…
  - Usually fairly accurate (?!?!?)
- However, due to the inadequacy of the conditional independence assumption…
  - Actual posterior-***probability*** estimates ***not*** accurate.
  - Output probabilities generally very close to 0 or 1.

Multi-Class Categorization

- Pick the category with max probability
- Create many 1 vs other classifiers
- Use a hierarchical approach (wherever hierarchy available)