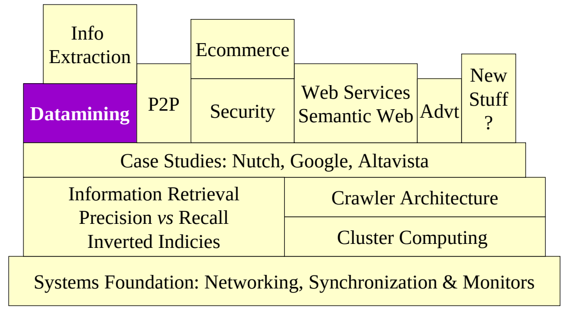
Text Categorization
(continued)

CSE 454

1

Course Overview



2

Immediate Organization

- Tues 11/1
  - Learning overview
  - Text categorization (Rocchio, nearest neighbor)
- Thurs 11/3
  - Text categorization (naïve Bayes); evaluation; topics
- Tues 11/8
  - Information extraction (HMMs)
- Thurs 11/10
  - KnowItAll (overview, rule learning, statistical model)
- Tues 11/15
  - KnowItAll (speedup, relational learning, opinion mining

Review: Checkers as ML

- Task T:
  - ***Playing checkers***
- Performance Measure P:
  - ***Percent of games won against opponents***
- Experience E:
  - ***Playing practice games against itself***
- Target Function
  - ***V: board -> R***
- Representation of approx. of target function

$$\hat{V}(b) = a + bx1 + cx2 + dx3 + ex4 + fx5 + gx6$$

4

Approximating the Target Function

- Profound Formulation:

  Can express any type of inductive learning as approximating a function
- E.g., Checkers
  - V: boards -> evaluation
- E.g., Handwriting recognition
  - V: image -> word
- E.g., Mushrooms
  - V: mushroom-attributes -> {E, P}

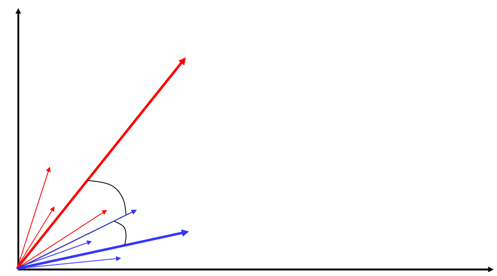
5

Supervised Learning

- **Inductive learning** or "**Prediction**":
  - **Given** examples of a function *(X, F(X))*
  - **Predict** function *F(X)* for new examples *X*
- Classification ("Categorization")
  - *F(X)* = Discrete
- Regression
  - *F(X)* = Continuous
- Probability estimation
  - *F(X)* = Probability*(X):*

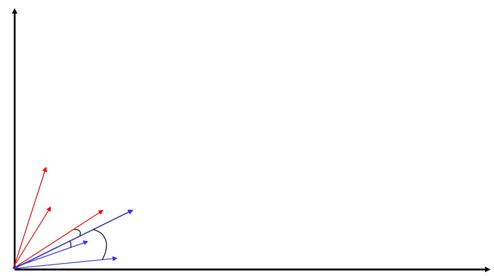
6

Illustration of Rocchio Text Categorization



7

Illustration of 3 Nearest Neighbor for Text



8

Learning ~ Prejudice meets Data

- 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?

9

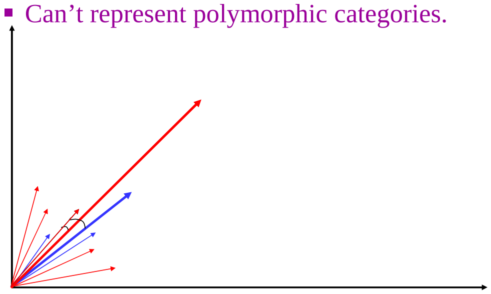
Two Strategies for ML

- Restriction bias: use prior knowledge to specify a restricted hypothesis space.
  - Rocchio
  - Naïve Bayes
- Preference bias: use a broad hypothesis space, but impose an ordering on the hypotheses.
  - General Bayesian learning

10

Rocchio Anomaly

- Prototype models ~ very strong bias
- Can't represent polymorphic categories.



11

Bayesian Methods

- Learning and classification methods based on probability theory.
  - Bayes theorem plays a critical role in probabilistic learning and classification.
  - Uses *prior* probability of each category given no information about an item.
- Categorization produces a ***posterior*** probability distribution over the possible categories given a description of an item.

12

Axioms of Probability Theory

- All probabilities between 0 and 1

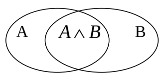
  $0 \le P(A) \le 1$

- True proposition has probability 1, false has probability 0.

  P(true) = 1   P(false) = 0.

- The probability of disjunction is:

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

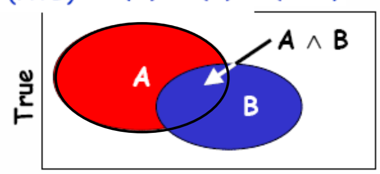


13

Probability: Simple & Logical

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

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



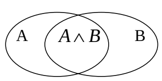
de Finetti (1931): an agent who bets according to probabilities that violate these axioms can be forced to bet so as to lose money regardless of outcome.

14

Conditional Probability

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

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



15

Independence

- A and B are *independent* iff:

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

These two constraints are logically equivalent

- Therefore, if A and B are independent:

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

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

16

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.)

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

17

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 determined since categories are complete and disjoint.

$$\sum_{i=1}^{n} P(c_i \mid E) = \sum_{i=1}^{n} \frac{P(c_i)P(E \mid c_i)}{P(E)} = 1$$

$$P(E) = \sum_{i=1}^{n} P(c_i)P(E \mid c_i)$$

18

Bayesian Categorization (cont.)

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

19

Naïve Bayesian Motivation

- Too many possible instances (exponential in m) to estimate all P(E | c_i)
- If we assume features of an instance are independent given the category (c_i) (*conditionally independent*).

  $$P(E \mid c_i) = P(e_1 \wedge e_2 \wedge \cdots \wedge e_m \mid c_i) = \prod_{j=1}^{m} P(e_j \mid c_i)$$
- Therefore, we then only need to know P(e_j | c_i) for each feature and category.

20

Naïve Bayesian Categorization

- If we assume features of an instance are independent given the category (c_i) (*conditionally independent*).

  $$P(E \mid c_i) = P(e_1 \wedge e_2 \wedge \cdots \wedge e_m \mid c_i) = \prod_{j=1}^{m} P(e_j \mid c_i)$$
- Therefore, we then only need to know P(e_j | c_i) for each feature and category.

21

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(sneeze|c_i)	0.1	0.9	0.9
P(cough|c_i)	0.1	0.8	0.7
P(fever|c_i)	0.01	0.7	0.4

22

Naïve Bayes Example (cont.)

Probability	**Well**	**Cold**	**Allergy**
P(c_i)	0.9	0.05	0.05
P(sneeze | c_i)	0.1	0.9	0.9
P(cough | c_i)	0.1	0.8	0.7
P(fever | c_i)	0.01	0.7	0.4

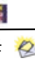
E={sneeze, cough, ¬fever}

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

23

Evidence is Easy?

P(c_i | E) = #  / (#  + # )

- Or…. Are their problems?

Assume evidence is words in document

Smooth with a Prior

P(c_i | E) = (#  + *mp*) / (#  + #  + *m*)

p = prior probability
m = weight

Note that if *m* = 10, it means "I've seen 10 samples that make me believe P(X_i | S) = *p*"

Hence, m is referred to as the equivalent sample size

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 | c_i) = 0.
- If e_k then occurs in a test example, *E*, the result is that $\forall c_i$: P(*E* | c_i) = 0 and $\forall c_i$: P(c_i | *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,…w_m} based on the probabilities P(w_j | 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 | 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 *i*th 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.
- Similar to Rocchio time complexity.

31

Easy to Implement

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

32

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.

34

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.

35

Evaluating Categorization

- 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,
  - c is the number of ***correctly classified*** test instances.
- Results can vary based on sampling error due to different training and test sets.
  - Bummer… what should we do?
- Average results over multiple training and test sets (splits of the overall data) for the best results.
  - Bummer… that means we need ***lots*** of labeled data…

36

N-Fold Cross-Validation

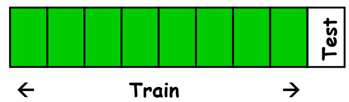
- Ideally: test, training sets are independent on each trial.
  - But this would require too much labeled data.
- Cool idea:
  - Partition data into *N* equal-sized disjoint segments.
  - Run *N* trials, each time hold back a different segment for testing
  - Train on the remaining *N*–1 segments.
- This way, at least test-sets are independent.
- Report average classification accuracy over the *N* trials.
- Typically, *N* = 10.

Also nice to report standard deviation of averages

37

Cross validation

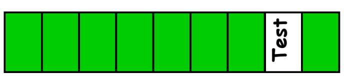
- Partition examples into *k* disjoint equiv classes
- Now create *k* training sets
  - Each set is union of all equiv classes ***except one***
  - So each set has (k-1)/k of the original training data



38

Cross Validation

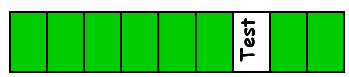
- Partition examples into *k* disjoint equiv classes
- Now create *k* training sets
  - Each set is union of all equiv classes ***except one***
  - So each set has (k-1)/k of the original training data



39

Cross Validation

- Partition examples into *k* disjoint equiv classes
- Now create *k* training sets
  - Each set is union of all equiv classes ***except one***
  - So each set has (k-1)/k of the original training data



40

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).

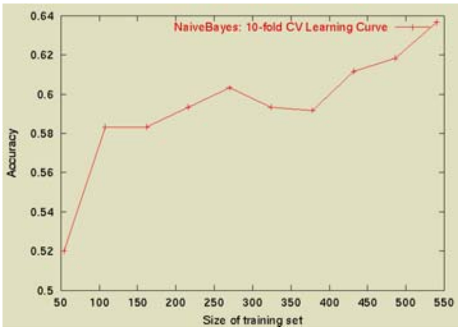
41

N-Fold Learning Curves

- Want learning curves averaged over multiple trials.
- Use *N*-fold cross validation to generate *N* full training and test sets.
- For each trial,
  - train on increasing fractions of the training set
  - measure accuracy on the test data
    - for each point on the desired learning curve.

42

Sample Learning Curve
(Yahoo Science Data)



43