Text Categorization

CSE 454

Sample Category Learning Problem

• Instance language: <size, color, shape>
  – size ∈ {small, medium, large}
  – color ∈ {red, blue, green}
  – shape ∈ {square, circle, triangle}
• \( C = \{ \text{positive, negative} \} \)
• \( D \):

<table>
<thead>
<tr>
<th>Example</th>
<th>Size</th>
<th>Color</th>
<th>Shape</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>small</td>
<td>red</td>
<td>circle</td>
<td>positive</td>
</tr>
<tr>
<td>2</td>
<td>large</td>
<td>red</td>
<td>circle</td>
<td>positive</td>
</tr>
<tr>
<td>3</td>
<td>small</td>
<td>red</td>
<td>triangle</td>
<td>negative</td>
</tr>
<tr>
<td>4</td>
<td>large</td>
<td>blue</td>
<td>circle</td>
<td>negative</td>
</tr>
</tbody>
</table>

Another Example: County vs. Country?

<table>
<thead>
<tr>
<th>Country</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>County</td>
<td>A county in the United States, typically in a state.</td>
</tr>
<tr>
<td>Country</td>
<td>A country in the United Nations system, which is the political entity of a nation.</td>
</tr>
</tbody>
</table>

Examples:
- King County, Washington
- Kenya

- \( D \):
  - Example 1: King County, Washington is a county in the United States.
  - Example 2: Kenya is a country in the United Nations system.
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
  – Newsgroup Messages
    – Recommending
    – Spam filtering
  – News articles
    – Personalized newspaper
  – Email messages
    – Routing
    – Prioritizing
    – Folderizing
    – Spam filtering

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, $D$.

$\{<\text{state}, \text{county}>, <\text{city}, \text{country}>, \ldots\}$

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

**Consistency**

Function Approximation

May not be any perfect fit
Classification $\sim$ discrete functions

General Learning Issues

- Many hypotheses are usually 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

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.
- **Occam’s razor**:
  - Finding a *simple* hypothesis helps ensure generalization.
Why is Learning Possible?

Experience alone never justifies any conclusion about any unseen instance.

Learning occurs when PREJUDICE meets DATA!

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

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?

A Learning Problem

Frobnitz?
Hypothesis Spaces

- **Complete Ignorance.** There are \(2^4 = 64\) 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 8 possibilities.

<table>
<thead>
<tr>
<th>(x_1), (x_2), (x_3), (x_4)</th>
<th>(y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>1</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>0</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>0</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>0</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>0</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>0</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>0</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>0</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>0</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>0</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>0</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>0</td>
</tr>
</tbody>
</table>

Terminology

- **Training example.** An example of the form \((x, f(x))\).
- **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 \(f(x) = 1\) are called positive examples or positive instances of the concept. Examples for which \(f(x) = 0\) are called negative examples or negative instances.
- **Classifier.** A discrete valued function. The possible values \(f(x) \in \{1, \ldots, K\}\) are called the classes or class labels.
- **Hypothesis Space.** The space of all hypotheses that our, 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.

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.
  - 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.
Axioms of Probability Theory

• All probabilities between 0 and 1
  \(0 \leq P(A) \leq 1\)

• Probability of truth and falsity
  \(P(\text{true}) = 1\) \(P(\text{false}) = 0\).

• The probability of disjunction is:
  \(P(A \lor B) = P(A) + P(B) - P(A \land B)\)

Conditional Probability

• \(P(A \mid B)\) is the probability of \(A\) given \(B\)

  Assumes:
  – \(B\) is all and only information known.

  Defined by:
  \(P(A \mid B) = \frac{P(A \land B)}{P(B)}\)

Independence

• \(A\) and \(B\) are independent iff:
  \(P(A \mid B) = P(A)\)

  These two constraints are logically equivalent
  \(P(B \mid A) = P(B)\)

• Therefore, if \(A\) and \(B\) are independent:
  \(P(A \mid B) = \frac{P(A \land B)}{P(B)} = P(A)\)

  \(P(A \land B) = P(A)P(B)\)
Independence

\[ P(A \land B) = P(A)P(B) \]

Conditional Independence

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

Bayes Theorem

\[
\begin{align*}
P(H \mid E) &= \frac{P(E \mid H)P(H)}{P(E)} \\
P(E \mid H) &= \frac{P(H \land E)}{P(H)} \\
P(H \land E) &= P(E \mid H)P(H) \\
\text{QED: } P(H \mid E) &= \frac{P(E \mid H)P(H)}{P(E)}
\end{align*}
\]
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 \( c_i \):
  \[
P(c_i | E) = \frac{P(c_i)P(E | c_i)}{P(E)}
  \]
- \( P(E) \) can be determined since categories are complete and disjoint.
  \[
  P(E) = \sum_{c_i} P(c_i)P(E | c_i)
  \]

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 \( P(E | c_i) \)

Naïve Bayesian Motivation

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

Naïve Bayes Example

- \( C = \{\text{allergy, cold, well}\} \)
- \( e_1 = \text{sneeze}; e_2 = \text{cough}; e_3 = \text{fever} \)
- \( E = \{\text{sneeze, cough, ~fever}\} \)

<table>
<thead>
<tr>
<th>Prob</th>
<th>Well</th>
<th>Cold</th>
<th>Allergy</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P(c_i) )</td>
<td>0.9</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>( P(\text{sneeze}</td>
<td>c_i) )</td>
<td>0.1</td>
<td>0.9</td>
</tr>
<tr>
<td>( P(\text{cough}</td>
<td>c_i) )</td>
<td>0.1</td>
<td>0.8</td>
</tr>
<tr>
<td>( P(\text{fever}</td>
<td>c_i) )</td>
<td>0.01</td>
<td>0.7</td>
</tr>
</tbody>
</table>
Naïve Bayes Example (cont.)

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

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:
  \[
P(e_j | c_i) = \frac{n_{ij}}{n_i}
\]
- 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, \(P(e_k | c_i) = 0\).
- If \(e_k\) then occurs in a test example, \(E\), the result is that \(P(E | c_i) = 0 \) and \(P(c_i | E) = 0\).

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 \(m\).
  \[
P(e_j | c_i) = \frac{n_{ij} + mp}{n_i + m} = (n_{ij} + 1) / (n_i + 2)
\]
- 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 \(P(w_j | c_i)\).
- 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 $c_i$.
- Let $P(c_i) = |D_i| / |D|$.
- 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 $P(w_j \mid c_i) = (n_{ij} + 1) / (n_i + |V|)$.

Text Naïve Bayes Algorithm (Test)

Given a test document $X$:
- Let $n$ be the number of word occurrences in $X$.
- Return the category:
  \[
  \arg\max_{c_i \in C} P(c_i) \prod_{a_i \in X} P(a_i \mid c_i)
  \]
  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 $D_i$, $n_*$, 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!
    - \(0.5^{57} = 7 \times 10^{-18}\)
- Solution? Use logs and add!
  - \(p_1 \times p_2 = e^{\log(p_1) + \log(p_2)}\)
  - 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)

Entity
- Person
  - Scientist
  - Artist
- Location
  - City
  - County
  - Country