Text Classification

Jason Rennie
jrennie@ai.mit.edu
Text Classification

• Assign text document a label based on content.

• Examples:
  – E-mail filtering
  – Knowledge-base creation
  – E-commerce
  – Question Answering
  – Information Extraction
E-mail Filtering

- Filter e-mail into folders set up by user.
- Aids searching for old e-mails
- Can be used to prioritize incoming e-mails
  - High priority to e-mails concerning your Ph.D. thesis
  - Low priority to “FREE Pre-Built Home Business”
Knowledge-Base Creation

- Company web sites provide large amounts of information about products, marketing contact persons, etc.
- Categorization can be used to find companies’ web pages and organize them by industrial sector.
- This information can be sold to, e.g. person who wants to market “Flat Fixer” to tire company.
E-Commerce

- Users locate products in two basic ways: search and browsing.
- Browsing is best when user doesn’t know exactly what he/she wants.
- Text classification can be used to organize products into a hierarchy according to description.
- eBay: Classification can be used to ensure that product fits category given by user.
Question Answering

• “When did George Washington die?”
• Search document database for short strings with answer.
• Rank candidates
• Many features (question type, proper nouns, noun overlap, verb overlap, etc)
• Problem: learn if string is the answer based on its feature values.
Information Extraction

- Want to extract information from talk announcements (room, time, date, title, speaker, etc)

- Many features may identify the information (keyword, punctuation, capitalization, numeric tokens, etc.)

- Problem: scan over text of document, filling buckets with desired information.

- Freitag (1998) showed that this approach could identify speaker (63%), location (76%), start time (99%) and end time (96%).
Basics of Text Classification

- Canonical Problem: Set of training documents, \((d_1, \ldots, d_n)\), with labels, \((y_1, \ldots, y_n)\). Set of test documents, \((x_1, \ldots, x_n)\).
- Goal: Assign correct labels to test documents.
Representation

From: dyer@spdcc.com (Steve Dyer)
Subject: Re: food-related seizures?

My comments about the Feingold Diet have no relevance to your daughter’s purported FrostedFlakes-related seizures. I can’t imagine why you included it.

\[
\begin{array}{|c|c|}
\hline
\text{food} & 1 \\
\hline
\text{seizures} & 2 \\
\hline
\text{diet} & 1 \\
\hline
\text{catering} & 0 \\
\hline
\text{religion} & 0 \\
\hline
\end{array}
\]
Representation

• Punctuation is removed, case is ignored, words are separated into tokens. Known as “feature vector” or “bag-of-words” representation.

• Vector length is size of vocabulary. Common vocabulary size is 10,000-100,000. Classification problem is very high dimensional.
Why is text different?

- Near independence of features
- High dimensionality (often larger vocabulary than # of examples!)
- Importance of speed
Word Vector has Problems

- longer document $\Rightarrow$ larger vector
- words tend to occur a little or a lot
- rare words have same weight as common words
Text is Heavy Tailed

![Graph showing the relationship between Term Frequency and Probability, with lines for Data, Power law, and Multinomial distributions.](image-url)
SMART “ltc” Transform

- new-tf\(_i\) = log\((tf_i + 1.0)\)
  - Corresponds to a power law distribution:
  \[ p(tf_i) \propto (1 + tf_i)^{\log \theta} \]

- new-wt\(_i\) = new-tf\(_i\) * log \(\frac{\text{num-docs}}{\text{num-docs-with-term}}\) (“TFIDF”)

- norm-wt\(_i\) = \(\frac{\text{new-wt}_i}{\sqrt{\sum_i \text{new-wt}_{i}^2}}\) (unit length vectors)
Types of Classification Problems

- Binary: label each new document as positive or negative.
  *Is this a news article Tommy would want to read?*

- Multiclass: give one of $m$ labels to each new document.
  *Which customer support group should respond to this e-mail?*

- Multitopic: assign zero to $m$ topics to each new document.
  *Who are good candidates for reviewing this research paper?*

- Ranking: rank categories by relevance.
  *Help user annotate documents by suggesting good categories.*
Multiclass Classification

- Decision Theory: minimum error decision boundary lies where density of top two classes are equal.
- Problem: Learning densities is ineffective for classification
Multiclass Classification

- Simple approach: construct one binary classifier to discriminate each class from the rest.
- Problem: we can’t say anything about the middle regions.
Multiclass Classification

- Better approach: construct lots of binary classifiers that, together, approximate the true boundaries.
Error Correcting Output Coding

- Idea: Represent each label as a length $l$ binary code. Learn one binary classifier for each of the $l$ bits in the code.
- For each example, assign label with “closest” code.
- Motivation: errors can be corrected using more bits than are needed to partition labels.

\[
\begin{array}{cccccccccc}
1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 \\
2 & 1 & -1 & -1 & -1 & 1 & -1 & -1 \\
3 & -1 & 1 & -1 & -1 & -1 & 1 & -1 \\
4 & -1 & -1 & 1 & -1 & -1 & -1 & 1 \\
\end{array}
\]

(1)

Code matrix
ECOC: The Loss Function

- ECOC works best when margin values are used

\[
\hat{H}(x) = \arg \min_{c \in \{1, \ldots, m\}} \sum_{i=1}^{l} g(f_i(x)M_{ci})
\]

- The loss function \((g)\) is a transform on the outputs:

  - Hamming
  - Hinge (SVM)
  - Logistic
ECOC: Some Results

• ECOC works better than using the usual multiclass approach for DTs and NNs. (Dietterich and Bakiri, 1995).
• Loss-based decoding works better than Hamming decoding using SVMs (Allwein et. al., 2000).
• ECOC w/ loss decoding very effective for text classification (Rennie and Rifkin, 2001).
Multiclass Classification: Interesting Questions

- Is a continuous code matrix useful? (Crammer & Singer 2001)
Multitopic Classification

- A document may be composed of many different topics.
- Zero or many topics per document.
- “Label” is a bit vector of topic indicators.
Multitopic Classification

- Basic approach: learn a binary classifier for each topic.

<table>
<thead>
<tr>
<th></th>
<th>Iraq vs. Non-Iraq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Politics vs. Non-Politics</td>
<td></td>
</tr>
<tr>
<td>Oil vs. Non-Oil</td>
<td></td>
</tr>
</tbody>
</table>

- Problem: “Iraq” document contains other things too.
Multitopic Classification

- How to identify part of document that gives it “Iraq” topic?
- Easier problem: How do we model a multi-topic document?
Multitopic Classification

• If we ignore word order, each word is randomly generated from one of $m$ topic-models.
• Problem becomes: how do we learn model for each topic?
• Ueda and Saito (2003) suggest modeling text as a multinomial and learning the models with an EM-like algorithm.
Parametric Mixture Model

- Let $\vec{y}$ be a label (bit vector)
- Let $\vec{\theta}_t = (\theta_{t1}, \ldots, \theta_{tV})$ be the model for topic $t$.
- Let $h_t(\vec{y})$ be the label $\vec{y}$ mixing proportion for topic $t$.

Model for a document with label $\vec{y}$ is

$$\phi(\vec{y}) = \sum_{t=1}^{m} h_t(\vec{y})\vec{\theta}_t.$$  \hspace{1cm} (3)

- Parameters for $\vec{y}$ are a convex combination
Parametric Mixture Model

- Simple case (PMM1): Assume $h_t(\tilde{y})$ equals $\frac{1}{k}$, $k$ is number of non-zero bits in $\tilde{y}$. (convex optimization)

- Harder case (PMM2): Learn $h_t(\tilde{y})$ via EM.

- Ueda and Saito: PMM1 works better than NB, SVM, kNN and NN. PMM2 useful in certain cases.

- PMM related to (McCallum 1999) and Latent Dirichlet Analysis (Blei, Ng, Jordan 2002)
Multitopic Classification: Interesting Problems

- Identify region(s) of document corresponding to topic(s)
- Capturing correlation between topics
- Hierarchy of topics (is parent or child more appropriate?)
Ranking

- How do you design a personalized search engine?
- Input: Ranking of documents based on relevance
- Want to learn a function that assigns rankings given a query
Ranking

• Option 1: Label documents rank $R$ or higher “relevant,” $R + 1$ or lower “not-relevant,” train a classifier. Rank based on classifier confidence values.

• Option 2: Train regression algorithm on rank values. Rank based on regression outputs.

• Option 3: Train a ranking algorithm.
Ranking

• A ranking algorithm has same form as classification and regression algorithms.

• Example: \( f(x) = \sum w_i x_i \) (linear)

• Difference is training

• Question: What constitutes a mistake?
Ranking: What is a Mistake?

- Classification: mistake if predicted rank, $r$, greater than $R$ and real rank, $r^t$ less than $R$ (or vice versa)
- Regression: error is difference between predicted value and true rank, $(r - r^t)^2$
- Ranking: mistake if documents are in wrong order
Ranking Loss: Examples

- Let \( \{d_1, \ldots, d_n\} \) be a set of documents.
- Let \( \{y^t_1, \ldots, y^t_n\} \) be the true ranks.
- Let \( \{\hat{y}_1, \ldots, \hat{y}_n\} \) be the predicted ranks.
- Let \( e_i = |y^t_i - \hat{y}_i| \).
- Loss = \( \sum_i e_i \).
Ranking Loss

- Ranking Loss better suited to a ranking problem
- Crammer and Singer (2002) show that using a ranking loss function works better on text than using the zero-one classification loss.
Review

- “Text Classification” appears in many forms
- Multiclass classification
- Multitopic classification
- Ranking
Tokenization

- First step of text classification is tokenization.

\[
\text{Document} \rightarrow \text{Tokenization} \rightarrow \text{Stemming} \rightarrow \text{Feature Selection} \rightarrow \text{Bag of Words}
\]

“They just canceled them completely”

| canceled | completely | just | them | they |
Tokenization

- Tokenization determines the features for the classifier
- A bad classifier with good features can easily outperform a good classifier with bad features
- Very important step!
Tokenization

- Tokenization gets little attention
- Standard methods: separate on whitespace, alphabetic strings, alphanumerical strings.
- Problem: different tokenizations work best for different domains.
- Is there a better way?
Compression for Word Learning

- Can compression help tokenization?
- We want tokens to reflect features that appear in the documents.
- Compression encourages the construction of features that appear more frequently than their individual characters would imply.
Compression for Word Learning: An Idea

- Begin with individual characters as the tokens.
- Allow pairs of tokens to be compressed together.
- De Marcken (1995) did exactly this.
- Creates a hierarchical decomposition of documents.
## Compression: Examples

<table>
<thead>
<tr>
<th>Rank</th>
<th>$-\log p_G(w)$</th>
<th>$w$</th>
<th>rep($w$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4.589</td>
<td>.</td>
<td>terminal</td>
</tr>
<tr>
<td>1</td>
<td>4.890</td>
<td>,</td>
<td>terminal</td>
</tr>
<tr>
<td>100</td>
<td>10.333</td>
<td>[ two]</td>
<td>[ [two]]</td>
</tr>
<tr>
<td>101</td>
<td>10.342</td>
<td>[ it was]</td>
<td>[[ it][ was]]</td>
</tr>
<tr>
<td>502</td>
<td>12.469</td>
<td>[ling]</td>
<td>[l[ing]]</td>
</tr>
<tr>
<td>15000</td>
<td>16.684</td>
<td>[ pakistan]</td>
<td>[[ pa]k[ist][an]]</td>
</tr>
<tr>
<td>15001</td>
<td>16.684</td>
<td>[ creativity]</td>
<td>[ [creat][ivity]]</td>
</tr>
<tr>
<td>27167</td>
<td>18.006</td>
<td>[[ massachusetts][ institute of technology]]</td>
<td></td>
</tr>
</tbody>
</table>
Compression: Hierarchy Example
[[f[or]][t[he]][[p[ur]][[p[o]s[e][of]]][[m[a][n][t[a][n]]][[i[n]g]]
[[i[n][t[e][r]]][[n[a][t[i][o][n]]][[a][l]]][[p[e][a][c][e]]][[a][n][d]][[p[r][o]][[m[o][t]][[i[n]g]]
[t[h[e]][[a][d][v][a][n][c][e]]][[[m[e][n][t]][[o][f]][[a][l][l]]][[p[e][o][p][l][e]]][[t[h[e]]
[[[u][n][i][t]][[e][d]]][[[s][t][a][t][e][s]]][[o][f][a][m][e][r][i][c][a]]][[j[o][i][n]][[e][d]][i[n]
[f[o][u][n][d]][[i][n][g]][t[h[e]][[[u][n][i][t]][[e][d]][[n[a][t[i][o][n]]]]]]

- Tokens can be taken from any level of the hierarchy—from “ur” to “the united nations.”
- Much more useful than collecting all substrings.
- Compression object eliminates numerous meaningless strings.
Classification via Compression

Standard compression problem:

- Want to transmit labels with fewest number of bits.
- Documents can be used as background knowledge.
- What is fewest number of bits needed to transmit labels?
Examples of Learned Features

\begin{itemize}
\item \texttt{x} \quad \texttt{comp.os.xwindows}
\item \texttt{windows} \quad \texttt{comp.os.ms-windows.misc}
\item \texttt{car} \quad \texttt{rec.autos}
\item \texttt{for\_sale} \quad \texttt{misc.forsale}
\item \texttt{turk} \quad \texttt{talk.politics.mideast}
\item \texttt{486} \quad \texttt{comp.sys.ibm.pc.hardware}
\item \texttt{3.1} \quad \texttt{comp.os.ms-windows.misc}
\item \texttt{\$} \quad \texttt{misc.forsale}
\item \texttt{condition} \quad \texttt{misc.forsale}
\end{itemize}
String Kernels

- Kernel method
- Documents projected into feature space of substrings
- Requires discount factor (longer strings receive less weight)
- Lodhi et. al. (2001) successfully applied string kernels to text—found they work about as well as substrings.
Summary

- Text classification comes in many different flavors.
- Text presents interesting and unique problems.