Document Classification
and
Word Representations

NLP ML Web
Fall 2013
Andrew Rosenberg
TA/Grader: David Guy Brizan
Document Classification

- News Aggregation
- Spam Filtering
- Automatic Labeling/Foldering
- Important vs. Regular Mail
- Safe Search
- Type of resource: Video, Audio, etc.
- Is a source “reliable”? 
Sentiment Classification

• Sentiment Classification
• Content-based Suggestion
  • We’ll come back to Sentiment Analysis
Google News
Classification

- Vanilla classification
- Document class is $t$, the target class.
- Document representation is $x$, the feature vector.
- Goal: Learn

$$f(x) = t$$
Clustering

- Find Similar Documents

- This is mostly **unsupervised** learning
What is a document?

- Newswire articles
- Blog posts
- Email
- What else?
Feature Representations

- Feature Representation is the “Perception” of your application

\[ f(x) = t \]
One-Hot Representation

- Many classifiers require $x$ to be numeric. $x \in \mathbb{R}^d$
- Assume you have a 3 word vocabulary.
  - $V = \{\text{SEE, SPOT, RUN}\}$
- Let every word in $V$ be a variable in $x$.
- To Represent SEE $x = [1, 0, 0]$
- To Represent SPOT $x = [0, 1, 0]$
- To Represent RUN $x = [0, 0, 1]$
One-Hot Representation

- SEE SPOT RUN RUN SPOT RUN
  \[1, 0, 0][0, 1, 0][0, 0, 1][0, 0, 1][0, 1, 0][0, 0, 1]\]
Bag-of-Words

• Now that we have a word representation, how do we represent a document? \( X = \{x_i\} \) where \( i \in \{1 \ldots N\} \)

• A document is a “Bag of Words”. Order doesn’t matter. Count doesn’t matter. Just tell me what’s present. \( g(X)_j = \max_i x_{ij} \)

Insert an image of a bag of words
Bag-of-Words

• SEE SPOT RUN RUN SPOT RUN

\[ [1, 0, 0] [0, 1, 0] [0, 0, 1] [0, 0, 1] [0, 1, 0] [0, 0, 1] \]

\[ g(X) = [1, 1, 1] \]
Bag-of-Words

- Now that we have a word representation, how do we represent a document? \( X = \{x_i\} \) where \( i \in \{1 \ldots N\} \)

- A document is a “Bag of Words”. Order doesn’t matter. Count doesn’t matter. Just tell me what’s present. \( g(X)_j = \max_i x_{ij} \)

Wordle.com of nlp wikipedia page
Term Vector

• We’re losing a lot of information.

• Let’s modify this a little.

• Use the number of times each term has been observed.

\[ g(X)_j = \sum_i x_{ij} \]
Term Vector

- SEE SPOT RUN RUN SPOT RUN
  \[ [1, 0, 0] [0, 1, 0] [0, 0, 1] [0, 0, 1] [0, 1, 0] [0, 0, 1] \]

\[ g(X) = [1, 1, 1] \] Bag of words

\[ g(X) = [1, 2, 3] \] Term Vector
Now that we have a way to express a document $x$, let’s do document classification.

Assume we have some labeled data for training. (i.e. known paired $x$ and $t$)

Naive Bayes

- Simple Classification Algorithm
- Direct extension of Bayes Rule
Bayes Rule

\[ p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)} \]
Interpretation of Bayes Rule

- **Prior**: Information we have before observation.

- **Posterior**: The distribution of Y after observing X

- **Likelihood**: The likelihood of observing X given Y

\[
p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}
\]
Naive Bayes Classification

\[ p(t|x) = \frac{p(x|t)p(t)}{p(x)} \]

- Calculate the probability of a class \( t \) given a document representation \( x \)
- Naive Bayes assumes all features are conditionally independent given the class.
Naive Bayes Classification

- a short derivation

\[ t^* = \arg\max_t p(t|x_1, x_2, \ldots, x_n) \]

\[ t^* = \arg\max_t \frac{p(x_1, x_2, \ldots, x_n | t)p(t)}{p(x_1, x_2, \ldots, x_n)} \]

\[ p(x_1, x_2, \ldots, x_n | t) = p(x_1 | t)p(x_2 | t) \ldots p(x_n | t) \]

\[ = \prod_i p(x_i | t) \]
Naive Bayes Classification

• Assuming independent features simplifies the math

\[
t^\ast = \arg \max_t p(t|x_1, x_2, \ldots, x_n)
\]

\[
t^\ast = \arg \max_t \frac{p(x_1, x_2, \ldots, x_n|t)p(t)}{p(x_1, x_2, \ldots, x_n)}
\]

\[
t^\ast = \arg \max_t \frac{p(x_1|t)p(x_2|t)\ldots p(x_n|t)p(t)}{p(x_1, x_2, \ldots, x_n)}
\]

\[
t^\ast = \arg \max_t \frac{p(t) \prod_i p(x_i|t)}{p(x_1, x_2, \ldots, x_n)}
\]

\[
t^\ast = \arg \max_t p(t) \prod_{i=1}^{21} p(x_i|t)
\]
An Example

\[
t^* = \arg\max_t p(x_1 | t)p(x_2 | t) \ldots p(x_n | t)p(t)
\]
An Example

<table>
<thead>
<tr>
<th>HOT</th>
<th>LIGHT</th>
<th>SOFT</th>
<th>RED</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLD</td>
<td>HEAVY</td>
<td>SOFT</td>
<td>RED</td>
</tr>
<tr>
<td>HOT</td>
<td>HEAVY</td>
<td>FIRM</td>
<td>RED</td>
</tr>
<tr>
<td>HOT</td>
<td>LIGHT</td>
<td>FIRM</td>
<td>RED</td>
</tr>
<tr>
<td>COLD</td>
<td>LIGHT</td>
<td>SOFT</td>
<td>BLUE</td>
</tr>
<tr>
<td>COLD</td>
<td>HEAVY</td>
<td>SOFT</td>
<td>BLUE</td>
</tr>
<tr>
<td>HOT</td>
<td>HEAVY</td>
<td>FIRM</td>
<td>BLUE</td>
</tr>
<tr>
<td>HOT</td>
<td>LIGHT</td>
<td>FIRM</td>
<td>BLUE</td>
</tr>
<tr>
<td>HOT</td>
<td>HEAVY</td>
<td>FIRM</td>
<td>?????</td>
</tr>
</tbody>
</table>

\[
t^* = \arg\max_t p(x_1|t)p(x_2|t) \ldots p(x_n|t)p(t)
\]

\[
p(t = \text{red}) = 0.5
\]

\[
p(t = \text{blue}) = 0.5
\]
An Example

\[
t^* = \arg\max_t p(x_1|t)p(x_2|t) \ldots p(x_n|t)p(t)
\]

\[
p(\text{hot}|t = \text{red}) = 0.75 \quad p(\text{heavy}|t = \text{red}) = 0.75 \quad p(\text{firm}|t = \text{red}) = 0.5
\]

\[
p(\text{hot}|t = \text{blue}) = 0.5 \quad p(\text{heavy}|t = \text{blue}) = 0.5 \quad p(\text{firm}|t = \text{blue}) = 0.5
\]
An Example

<table>
<thead>
<tr>
<th>HOT</th>
<th>LIGHT</th>
<th>SOFT</th>
<th>RED</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLD</td>
<td>HEAVY</td>
<td>SOFT</td>
<td>RED</td>
</tr>
<tr>
<td>HOT</td>
<td>HEAVY</td>
<td>FIRM</td>
<td>RED</td>
</tr>
<tr>
<td>HOT</td>
<td>LIGHT</td>
<td>FIRM</td>
<td>RED</td>
</tr>
<tr>
<td>COLD</td>
<td>LIGHT</td>
<td>SOFT</td>
<td>BLUE</td>
</tr>
<tr>
<td>COLD</td>
<td>HEAVY</td>
<td>SOFT</td>
<td>BLUE</td>
</tr>
<tr>
<td>HOT</td>
<td>HEAVY</td>
<td>FIRM</td>
<td>BLUE</td>
</tr>
<tr>
<td>HOT</td>
<td>LIGHT</td>
<td>FIRM</td>
<td>BLUE</td>
</tr>
</tbody>
</table>
| HOT  | HEAVY | FIRM | ????

\[ t^* = \arg\max_t p(x_1 | t)p(x_2 | t) \ldots p(x_n | t)p(t) \]

\[ p(\text{hot} | t = \text{red})p(\text{heavy} | t = \text{red})p(\text{firm} | t = \text{red})p(c = \text{red}) = 0.75 \times 0.5 \times 0.5 \times 0.5 \]
\[ = 0.09375 \]

\[ p(\text{hot} | t = \text{blue})p(\text{heavy} | t = \text{blue})p(\text{firm} | t = \text{blue})p(c = \text{blue}) = 0.5 \times 0.5 \times 0.5 \times 0.5 \]
\[ = 0.0625 \]
• \( p(x|t) \) can be a gaussian (typically) or poisson or any other distribution in Naive Bayes.

\[
g(X) = [1, 1, 1] \quad \text{Bag of words}
\]

\[
g(X) = [1, 2, 3] \quad \text{Term Vector}
\]
What is the problem?

- Obviously these are naive representations.
- What is the problem?
What is the problem?

• Some problems
  1. No ordering information is included.
  2. All words are equally important.
  3. The vectors are too big.
  4. All words are equally different.
N-grams

• N-grams: each N-gram is a sequence of N ordered elements of a longer sequence.
  • 2-grams = bigrams
  • 3-grams = trigrams
  • 4-gram, 5-gram
• N-grams capture local ordering information.
N-Gram example

• SEE SPOT RUN RUN SPOT RUN
• SEE SPOT
• SPOT RUN
• RUN RUN
• RUN SPOT
• SPOT RUN
N-Gram example

- `<s>` SEE SPOT RUN RUN SPOT RUN `<s>`
- `<s>` SEE
- SEE SPOT
- SPOT RUN
- RUN RUN
- RUN SPOT
- SPOT RUN
- RUN `<s>`
Bag-of-N-grams

- If your vocabulary has $k$ words, how many $n$-grams can you have?

- Bag-of-words and Term-Vector representations use $N$-grams as the vocabulary.

- No difference in operation.
Sparse Features

- High order n-grams capture more information.

- The higher N gets, the less likely you will have seen the n-gram before
  - (or enough to generalize from)

- Sparse observations make learning impossible.
Learning with rare features

• Consider Naive Bayes.
• What is $p(x \mid t)$ if $x$ is a unique feature?
  • If $x$ is discrete (like an n-gram)?
  • If $x$ is continuous with a singular value modeled by a normal distribution?
• Drastic over or under estimation.
Black Swans

NEW YORK TIMES BESTSELLER
THE BLACK SWAN
The Impact of the Highly Improbable
Nassim Nicholas Taleb
Black Swans

• In the 17th Century, all known swans were white.
Black Swans

• In the 17th Century, all known swans were white.

• Based on evidence, it is impossible for a swan to be anything other than white.
Black Swans

• In the 17th Century, all known swans were white.
• Based on evidence, it is impossible for a swan to be anything other than white.
• In the 18th Century, black swans were discovered in Western Australia.
Black Swans

• In the 17\textsuperscript{th} Century, all known swans were white.

• Based on evidence, it is impossible for a swan to be anything other than white.

• In the 18\textsuperscript{th} Century, black swans were discovered in Western Australia.

• Black Swans are rare, sometimes unpredictable events, that have extreme impact.
Black Swans

• In the 17\textsuperscript{th} Century, all known swans were white.

• Based on evidence, it is impossible for a swan to be anything other than white.

• In the 18\textsuperscript{th} Century, black swans were discovered in Western Australia.

• Black Swans are rare, sometimes unpredictable events, that have extreme impact

• Almost all statistical models underestimate the likelihood of unseen events.
Long Tail

- Many events follow an exponential distribution
- These distributions have a very long “tail”.
  - I.e. A large region with significant probability mass, but low likelihood at any particular point.
- Often, interesting events occur in the Long Tail, but it is difficult to accurately model behavior in this region.

Images from wikipedia: exponential distribution, Zipf’s law
Google N-grams

- N-gram stats from millions of books in English and other languages (Chinese, German, Hebrew, French, Spanish, Italian, Russian)
- Part of speech tagging.
- Up to 5-grams.
- Customized viewing tool.
- [http://books.google.com/ngrams](http://books.google.com/ngrams)
- [http://storage.googleapis.com/books/ngrams/books/datasetsv2.html](http://storage.googleapis.com/books/ngrams/books/datasetsv2.html)
What is the problem?

• Some problems
  1. No ordering information is included.
  2. All words are equally important.
  3. The vectors are too big.
  4. All words are equally different.
Stop words

- These are high-frequency closed-class words
  - determiners, conjunctions, pronouns, prepositions
- Stop words are often filtered out common words before processing.
Stop words

• These are high-frequency closed-class words
  • determiners, conjunctions, pronouns, prepositions

Does high-frequency equal low-information?
Stop words

- These are high-frequency closed-class words
  - determiners, conjunctions, pronouns, prepositions
- Stop words are often filtered out before processing.

Does high-frequency equal low-information?

Shannon says “Yes”.

Yes.
TF*IDF

• Measure the relative importance of terms for classification.

• Every term gets assigned a TF*IDF score.

• TF : Term Frequency.
  • How frequent is this term *in this document*?

• IDF: Inverse Document Frequency
  • What inverse ratio of documents contain this term?

• Usually with a log applied to it.
TF*IDF is a weighted Term Vector.

Larger entries in a Term Vector are more important when calculating vector similarity.

To modify NaiveBayes, variables can be explicitly weighted by tf*idf.

TF*IDF is often used in similarity calculations.

\[
\begin{align*}
tf_{t,d} &= \frac{|t \in D_d|}{|D_d|} \\
idf_t &= \log \frac{|D|}{|\{d' \in D | t \in d'\}|}
\end{align*}
\]
Entropy

\[ H(X) = - \sum_{x \in X} p(x) \log p(x) \]

• Entropy measures the amount of “information” in a distribution.

• The flatter the distribution, the more information it holds.

• The more skewed, the lower the entropy, the less information.

• What does this have to do with tf*idf?
Entropy

\[ H(X) = - \sum_{x \in X} p(x) \log p(x) \]

- What does this have to do with tf*idf?
- tf is measuring how important a word is in a document.
- \( p(x) \) is normalized term frequency. \( tf/N \)
- But we don’t have any document frequency information.
Mutual Information

\[ I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \]

- What is the amount of shared information between two variables?
- What is the value of \( I(X;Y) \) if \( X \) and \( Y \) are independent?
- Harder: What is the maximum value of \( I(X;Y) \)?
Pointwise Mutual Information

\[ \text{pmi}(x, y) = \log \frac{p(x,y)}{p(x)p(y)} \]

- Mutual information is calculated over a whole joint distribution.
- Pointwise mutual information gives the mutual information of a single pair of outcomes
  - like a term document pair...
  - or a term and context...
Aside: Mixed type representations

- Say you have a classifier that can only use numerical features.
  \[ t = \tanh(c_0 + c_1x_1 + c_2x_2) \]
- But you have some categorical features.
  - Occupation, Word_ID, Bigram_ID, etc.
- How can you use this classifier?
  - How can you convert these categorical features into numeric features?
What is the problem?

• Some problems
  1. No ordering information is included.
  2. All words are equally important.
  3. The vectors are too big.
  4. All words are equally different.
Reducing Vocabulary Size

- Broadclassing
- Stopword removal / Appropriate capitalization
- Stemming
- Learning Word Classes
Content vs. Function

• Definitions.
  • What’s a function word?

• What’s more important for document classification?
  • Function or Content words?
Pronouns and Function Words

• What’s more important function words or content words?
• Power/Social dynamics
• Depression, Emotion
• Deception (maybe)

• James Pennebaker, Lillian Lee, others

• Pennebaker’s Ted Talk: http://www.youtube.com/watch?v=PGsQwAu3PzU

• Why might this be true?
Word Classes

• **Broadclassing**: Represent a word not by its lexical form, but by some “broad class” that describes the *type* of word.

• Broadclassing reduced vocabulary size, enabling longer n-grams.

• e.g. bigrams of lexical items, vs. 4-grams of function/content words.
Common Word Classes

• Part of Speech
  • Nouns, Verbs, Adverbs, Adjectives, Cardinal Numbers, Function Words.
  • Penn Treebank: 36 tags
    http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html
Stemming

- Broadclassing gives ways to reduce the size of the vocabulary.
- Stemming/Lemmatization
  - friends -> friend
  - unfriend -> friend
    - NLTK has one. OpenNLP does too.
- Morphological Analysis
  - morphessor (unsupervised)
  - magead (supervised for Arabic)
Learning Word Classes

- Assume you know nothing about a language.
- How can you find Similar words?
Exchange Clustering

• Maximize class bigram probability.

• Objective Function:

  • $w$ is a word, $c$ is a word class, $d$ is the word class of the next word.

  $$J(V, C) = \sum_{i=n}^{N} \log p(w_i | c_i) p(c_i | d_i)$$

  $$\approx \sum_{w \in V} N(w) \log N(w)$$

  $$+ \sum_{c \in C, d \in C} N(c, d) \log N(c, d)$$

  $$- 2 \sum_{c \in C} N(c) \log N(c)$$

Martin et al. 1998 “Algorithms for bigram and trigram word clustering” 55
Exchange Clustering

**Input:** number of clusters $k$

Compute initial clustering

**while** clustering has changed

**forall** $w$ in $V$ **do**

**forall** $c$ in $C$ **do**

move word $w$ to cluster $c$

compute objective function

move word $w$ to the cluster $c'$ that maximizes the objective function

Uszkoreit and Brants 2008 “Distributed Word Clustering for Large Scale Language Modeling in Machine Translation”
Words as numbers

- We’ve already described how we represent a word or document as a vector of numbers.

- The size of the vector is the size of the vocabulary.

- How else can we reduce the size of this vector?
Dimensionality Reduction

• Take a vector in $\mathbb{R}^d$ and project it to $\mathbb{R}^k$, where $k < d$.

• Simple option:
  • Ignore some dimensions.
  • Which dimensions are important?
How many dimensions do you need?
Identify Dimensions of high variance

- Assumption: directions that show high variance represent the appropriate/useful dimension to represent a feature set.
Principal Components Analysis

- PCA identifies the dimensions of greatest variance
- Represent original data $X$ in this PCA space.
- And you can drop the low variance dimensions.
Principal Components Analysis

- For Language Data these are term-vectors, each representing a single document.
Eigenvectors

• Eigenvectors are orthogonal vectors that define a space, the eigenspace.

• Any data point can be described as a linear combination of eigenvectors.

• Eigenvectors, $v$, of a square matrix, $A$, have the following property.

$$A\vec{v} = \lambda \vec{v}$$

• The associated lambda is the eigenvalue.
PCA

• Write each data point in this new space

\[ \vec{x}_i \approx \vec{\mu} + \sum_{j=1}^{C} c_{ij} \vec{v}_j \]

• To do the dimensionality reduction, keep \( C < D \) dimensions.

• Each data point is now represented as a vector of c’s.

• Excellent tutorial by Lindsay Smith
Eigenvectors to PCA

- PCA is easy once we have eigenvectors and the mean.
- Identifying the mean is easy.

\[ \mathbf{x}_i \approx \mu + \sum_{j=1}^{C} c_{ij} \mathbf{v}_j \]

- Eigenvectors of the covariance matrix, represent a set of direction of variance.
- Covariance between two X dimensions, i.e. Two words.
- Eigenvalues represent the degree of the variance.
Eigenvectors of the covariance matrix

\[
\begin{bmatrix}
\Sigma(1,1) & \Sigma(1,2) & \Sigma(1,3) \\
\Sigma(1,2) & \Sigma(2,2) & \Sigma(2,3) \\
\Sigma(1,3) & \Sigma(2,3) & \Sigma(3,3)
\end{bmatrix}
= \begin{bmatrix}
\vec{v}_1 & \vec{v}_2 & \vec{v}_3
\end{bmatrix}
\begin{bmatrix}
\lambda_1 & 0 & 0 \\
0 & \lambda_2 & 0 \\
0 & 0 & \lambda_3
\end{bmatrix}
\begin{bmatrix}
\vec{v}_1 & \vec{v}_2 & \vec{v}_3
\end{bmatrix}
\]

- Eigenvectors are orthonormal
- All eigenvalues are non-negative.
- Eigenvalues are sorted. \( \lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \)
- Larger eigenvalues, higher variance

\[\Sigma = V \Lambda V^{-1}\]
Dimensionality Reduction with PCA

• To convert from an original data point to PCA

\[ c_{i,j} = (\vec{x}_i - \vec{\mu})^T \vec{v}_j \]

• To reconstruct a point

\[
\left\{
\begin{align*}
    x'_1 &= \mu + \sum_{j=1}^{C} c_{1,j} \vec{v}_j \\
    &\ldots\\
    x'_n &= \mu + \sum_{j=1}^{C} c_{n,j} \vec{v}_j 
\end{align*}
\right\}
\]
Important Aside: Feature Normalization

• Assume 2 features:
  • Percentile GPA
  • Height in cm.
• Which dimension shows greater variability?
Important Aside: Feature Normalization

- Assume 2 features:
  - Percentile GPA
  - Height in cm.
- Which dimension shows greater variability?
Important Aside: Feature Normalization

• Assume 2 features:
  • Percentile GPA
  • Height in cm.

• Which dimension shows greater variability?
Normalization

- How could you make these features more comparable?
Normalization

- Range normalization

\[ x' = \frac{x - \min(X)}{\max(X) - \min(X)} \]
Normalization

• **Z-score normalization**
  
  • $x'$ has 0 mean
  
  • $x'$ has stdev of 1

\[
x' = \frac{x - \text{mean}(X)}{\text{std}(X)} = \frac{x - \mu}{\sigma}
\]
Latent Semantic Analysis

- PCA operates on word covariance matrix
- Consider a Term-document matrix.
- Singular Value Decomposition (very similar to eigenvector decomposition)

\[ X = U \Sigma V^T \]

![Diagram showing the relationship between the original matrix and the low-dimensional matrices.](image)
Autoencoders

- Learn mappings:
  \[ \tilde{y} = W^{(1)} \tilde{x} \]
  \[ \tilde{x'} = W^{(2)} \tilde{y} \]
Stacked Autoencoders

- Learn mappings:

\[ \tilde{y} = W^{(2)} W^{(1)} \tilde{x} \]
\[ \tilde{x}' = W^{(4)} W^{(3)} \tilde{y} \]
Stacked Autoencoders

- Learn mappings:

\[ \tilde{y} = W^{(2)} W^{(1)} \tilde{x} \]

\[ \tilde{x}' = W^{(4)} W^{(3)} \tilde{y} \]

We’ll build to this next time

Monday, September 16, 13
• Linear Discriminant Analysis
  http://courses.cs.tamu.edu/rgutier/cs790_w02/l6.pdf

• Multi-Dimensional Scaling
  http://homepages.uni-tuebingen.de/florian.wickelmaier/pubs/Wickelmaier2003SQRU.pdf

• t-SNE
  http://homepage.tudelft.nl/19j49/t-SNE.html