CSE574/EE517 – Lecture 3

• Announcements:
  – Lab 1 is due today; email your respective prof
  – Lab 2 deadline changed to Jan 21

• Today:
  – Vector vs. process models
  – Continuous vs. symbolic representations
Two Models of Language

• Process (sequence) model
  – Document = sequence of words (or tags, etc.)
  – \( S_{doc} = w_1, w_2, w_3, \ldots, w_T; \ T=\text{doc length} \)
  – \( w_i \) = index of \( i \)-th word in sequence

• Vector model
  – Document = bag of words (or other features)
  – \( X_{doc} = n_1, n_2, n_3, \ldots, n_V; \ V=\text{vocabulary size} \)
  – \( n_i \) = count of \( i \)-th word in vocabulary
The aim of a linguistic science is to be able to characterize and explain the multitude of linguistic observations circling around us, in conversations, writing, and other media. Part of that has to do with the cognitive side of how humans acquire, produce and understand language, part of it has to do with understanding the relationship between linguistic utterances and the world, and part of it has to do with understanding the linguistic structures by which language communicates. In order to approach the last problem, people have proposed that there are rules which are used to structure linguistic expressions. .... linguists .... language.
Two Models of Language

The aim of a **linguistic** science is to be able to characterize and explain the multitude of **linguistic** observations circling around us, in conversations, writing, and other media. …

- **Process (sequence) model**
  - \( S_{\text{doc}} = w_1, w_2, \ldots, w_T; \ T = 144 = \text{doc length (variable)} \)
  - \( w_5 = \text{“linguistic”} \) (\( w_5 = 45,739 = \text{index of “linguistic”} \))

- **Vector model**
  - \( X_{\text{doc}} = n_1, n_2, \ldots, n_V; \ V = 120k = \text{vocabulary size (fixed)} \)
  - \( n_{45739} = 5 \) (count of 5 instances in this paragraph)
More Generally…

• Vector representation of a document is any fixed set of measurements on it.
• Measurements might include:
  – Number of instances of a word (count)
  – Weighted count of a word
  – Number of instances of a word pair
  – Number of sentences
  – Number of sentences with past tense
  – Etc.
Vector vs. Process Models

• Process models: sequence matters
  – Good for characterizing fluency of language
  – Used in speech recognition, translation, summarization
  – Examples: Markov (n-gram), HMM, CRF, …

• Vector models: character matters
  – Good for text classification
  – Used in topic/IR and genre identification
  – Examples: BOW, min dist, SVM, decision tree, log-linear, MaxEnt, ….
Most Basic Vector Model

Bag of Words (BOW)

- Features = Word counts (or indicators or weighted word counts)
- Model = Naïve Bayes (assume features are independent)

\[
P(n_1, n_2, \ldots, n_V | \text{class}) = \prod_{i=1}^{V} P(n_i | \text{class})
\]

- Assumption: Word order doesn’t matter
BOW vs. Unigram

• Unigram (multinomial)
\[
\log P(w_1, w_2, \ldots, w_T | \text{class}) = \sum_{t=1:T} \log P(w_t | \text{class}) \\
= \sum_{i=1:V} n_i \log P(v_i | \text{class})
\]
where \( v_i = i\)-th word in vocab
small # adds (T), big prob table (V)

• BOW with word counts
\[
\log P(n_1, n_2, \ldots, n_V | \text{class}) = \sum_{i=1:V} \log P_i(n_i | \text{class})
\]
big # of adds (V), small prob tables (N)
Dimensionality Reduction

- Vocabularies can be 100k or more – that’s a big feature vector! (though sparse)
- Two pre-processing reduction methods
  - Discard “stop words” (e.g. a, the, to, … mostly function words, designed by hand)
  - Stemming (look, looked, looks, looking \(\rightarrow\) look)
- Two general auto reduction techniques:
  - Selection: throw out less important features
  - Transformation: transform first, then throw out
Warnings

• Rule-based warnings: makes sense for topic classification but not for all tasks
  – Stop-word examples: “uh” should be a stop word in conversational speech, pronouns are useful for genre/style
  – Stemming example: tense useful in genre

• Pre-processing (stemming) warning:
  – Stemmers aren’t perfect (beautiful → beauti)
  – Can introduce variability (booked → book N/V?)
Feature Selection Criteria

- Define: \( c = \text{class}, \ w = \text{word} \)
- Maximum Mutual Information (MMI)
  \[
  I(C;W) = H(C) - H(C|W)
  \]
  \[
  \text{Max } I(C;W) \equiv \text{Min } H(C|W) \text{ (min entropy)}
  \]
- Information Gain (IG):
  \[
  w/\neg w: \text{ word present/absent (1/0)}
  \]
  \[
  IG = H(C) - p(w)H(C|w) - p(\neg w)H(C|\neg w)
  \]
  \[
  = H(C) - H(C|I_w) = I(C;I_w)
  \]
- Assuming features are independent (Naïve Bayes -- NB), omit ones with worst scores
Feature Selection Example
Warnings (cont.)

• Auto-selection smoothing issues
  \[ H(C|W) = -\sum_{c,w} P(c,w) \log P(c|w) \]
  low entropy with small # samples, so bias towards including infrequent words

• NB independence assumption may not be valid, in which case treating features separately is suboptimal
Word Features

• 1/0 Indicators (Bernoulli model)
• Word counts (binned non-parametric dist, Poisson, etc.)
• Continuous space options:
  – Weighted word counts
  – Latent semantic analysis (LSA)
  – Probabilistic LSA
  – ...

Note: These continuous space methods grew out of Information Retrieval. They are well suited to topic classification, but not all text classification problems.
Weighted Counts

• IR vector space representation: TF-IDF
• Define
  – \( tf_{ij} \) : # of \( w_i \) in \( d_j \) (counts)
  – \( df_i \) : # of docs in collection that \( w_i \) occurs in
  – \( N \) : # of docs in collection
• Some options for weighted counts:
  – Natural: \( x_{ij} = tf_{ij}(N/df_i) \)
  – Log: \( x_{ij} = (1+\log(tf_{ij})) \log(N/df_i) \)
Weighted Counts Classifier

• TF-IDF is often used with the cosine distance for finding the similarity of two document vectors (or doc & query)
  – $\cos(x,y) = \frac{x^t y}{|x||y|}$
  – Useful for text clustering, finding nearest neighbors

• For normalized vectors, cosine dist is the same as min Euclidean distance
  – Useful for classification with local learning (e.g. KNN)
Latent Semantic Analysis* (LSA)

- Linear transformation of word count vector
- Basics:
  - Build a VxN document-word matrix $A$ for $N$ docs
    $w_{ij} =$ count of $i$-th word in $j$-th doc
  - Decompose $A$ using singular value decomposition (SVD) $A=TS\mathbf{D^t}$ ($S$ is $r\times r$)
  - $T^tA=SD \Rightarrow$ reduced dimension document representation ($y=T^tx$ gives $r$-dim vector)
  - $AD=TS \Rightarrow$ continuous space word representation ($z_i=[w_{i1}, ..., w_{iN}]D$ gives $r$-dim vec)

* Also called latent semantic indexing (LSI)
LSA in pictures

\[ d_i \quad w_j \quad A \]

# documents

V=\# word features

= T

S

D^t
### LSA Example (M&S)

<table>
<thead>
<tr>
<th></th>
<th>d₁</th>
<th>d₂</th>
<th>d₃</th>
<th>d₄</th>
<th>d₅</th>
<th>d₆</th>
</tr>
</thead>
<tbody>
<tr>
<td>cosmonaut</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>astronaut</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>moon</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>car</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>truck</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>d₁</th>
<th>d₂</th>
<th>d₃</th>
<th>d₄</th>
<th>d₅</th>
<th>d₆</th>
</tr>
</thead>
<tbody>
<tr>
<td>y₁</td>
<td>-1.62</td>
<td>-0.60</td>
<td>-0.04</td>
<td>-0.97</td>
<td>-0.71</td>
<td>-0.26</td>
</tr>
<tr>
<td>y₂</td>
<td>-0.46</td>
<td>-0.84</td>
<td>-0.30</td>
<td>1.00</td>
<td>0.35</td>
<td>0.65</td>
</tr>
</tbody>
</table>

\( T^iA = SD \rightarrow \) reduced dimension document representation \( (y = T^i x \) gives r-dim vector)
Probabilistic LSA

• Probabilistic mapping to a continuous space (avoid squared error criterion)

• Basic idea:
  – Train a topic mixture model
  – Represent document by vector of topic (mixture component) probabilities \( \{p(t|d)\} \)
  – Represent word by vector of topic (mixture component) probabilities \( \{p(t|w)\} \)