Logistics

• Next Tuesday: in addition to regular course content, TA will use half an hour to discuss the common problems seen in assignment 1.
  • Output format is incorrect, or no output at all
  • Code not runnable

• Grades, comments, and rubrics will be released by today. Feel free to reach out to TA during office hour if you have any question wrt grading.

Neural language models

• Skip-grams
• Continuous Bag of Words (CBOW)
• Math details can be found at https://cs224d.stanford.edu/lecture_notes/notes1.pdf (not required for this course)

Prediction-based models:
An alternative way to get dense vectors

• Skip-gram (Mikolov et al. 2013a), CBOW (Mikolov et al. 2013b)
• Idea: Learn embeddings as part of the process of word prediction
• Implementation: Train a neural network to predict neighboring words
• Advantages:
  • Fast, easy to train (much faster than SVD)
  • Available online in the word2vec package
  • Including sets of pretrained embeddings!

Word2vec

• Popular embedding method
• Very fast to train
• Code available on the web
  • Idea: predict rather than count

Word2vec

• Given a sentence:
  ... lemon, a tablespoon of apricot jam a pinch ...
• Instead of counting how often each word w occurs near "apricot"
• Train a classifier on a binary prediction task:
  • Is w likely to show up near "apricot"?

• We don’t actually care about this task
• But we’ll take the learned weights (will be discussed later) as the word embeddings
Brilliant insight: Use running text as implicitly supervised training data!

- A word near apricot
- Acts as gold 'correct answer' to the question
- "Is word w likely to show up near apricot?"
- No need for hand-labeled supervision
- The idea comes from neural language modeling
  - Bengio et al. (2003)
  - Collobert et al. (2011)

Word2Vec: Skip-Gram Task

- Now we have positive samples.
- Where do the "negative samples" come from?

Word2Vec: Skip-Gram Task

- Word2vec provides a variety of options. Let's do
  - "skip-gram with negative sampling" (SGNS)

Skip-gram algorithm

1. Treat the target word and a neighboring context word as positive examples.
2. Randomly sample other words in the lexicon to get negative samples
3. Use logistic regression (will discuss formulation later) to train a classifier to distinguish those two cases
4. Use the weights as the embeddings

Skip-gram Training Data

- Training sentence:
  ... lemon, a tablespoon of apricot jam a pinch ...
  c1  c2  target c3  c4

Assume context words are those in +/- 2 word window
Skip-gram Goal

- Given a tuple \((t, c)\) = target, context
  - \((\text{apricot, jam}) \rightarrow +\)
  - \((\text{apricot, aardvark}) \rightarrow -\)
- Return probability that \(c\) is a real context word (or not):
  - \(P(+|t,c)\) = positive
  - \(P(-|t,c) = 1 - P(+|t,c)\) = negative

How to compute \(p(\cdot|t,c)\)?

- Intuition:
  - Words are likely to appear near similar words
  - Model similarity with dot-product!
  - Similarity \((t,c) \propto t \cdot c\)
- Problem:
  - Dot product is not a probability!
  - \((\text{Neither is cosine})\)

Turning dot product into a probability

- The sigmoid lies between 0 and 1:
  \[
  \sigma(x) = \frac{1}{1 + e^{-x}}
  \]

For all the context words:

- Assume all context words are independent
  \[
  P(+|t,c_1 \ldots c_k) = \prod_{i=1}^{k} \frac{1}{1 + e^{-x_i}}
  \]
  \[
  \log P(+|t,c_1 \ldots c_k) = \sum_{i=1}^{k} \log \frac{1}{1 + e^{-x_i}}
  \]

Skip-gram Training Data

- Training sentence:
  \[
  \ldots \text{lemon, a tablespoon of apricot jam a pinch} \ldots
  \]
  \[
  c_1 \quad c_2 \quad t \quad c_3 \quad c_4
  \]
- Training data: input/output pairs centering on \text{apricot}
- Assume a +/- 2 word window
Let's start by considering a single piece of the training data, from the sentence...

... lemon, a tablespoon of apricot jam a pinch ...

c1 c2 t c3 c4

**Training sentence:**

- **Training data:** input/output pairs centering on apricot
- **Assume a +/- 2 word window**

**Positive examples:**

- apricot tablespoon
- apricot preserves
- apricot or

**Negative examples:**

- apricot aardvark
- apricot twelve
- apricot dear
- apricot coaxial
- apricot forever

**Choosing noise words (we’ve seen this!)

- Could pick w according to their unigram frequency P(w)
- More common to chosen then according to p_i(w)
  
  \[
  p_i(w) = \frac{\text{count}(w)}{\sum \text{count}(w)}
  \]
- \(a=0.75\) works well because it gives rare words slightly higher probability
- To show this, imagine two events p(a)=.99 and p(b) = .01:
  
  \[
  p_a(a) = \frac{99}{99+1} = .97
  \]
  
  \[
  p_a(b) = \frac{1}{99+1} < .01
  \]

**Setup

- Let’s represent words as vectors of some length (say 300), randomly initialized.
- So we start with 300 * V random parameters
- Over the entire training set, we’d like to adjust those word vectors such that we:
  
  - Maximize the similarity of the target word, context word pairs (t,c) drawn from the positive data
  - Minimize the similarity of the (t,c) pairs drawn from the negative data
Formally

- We want to maximize the following objective
  \[ \sum_{(t,c) \in +} \log P(+|t,c) + \sum_{(t,c) \in -} \log P(-|t,c) \]
- Maximize the + label for the pairs from the positive training data, and the - label for the pairs sampled from the negative data.

Focusing on one target word \( t \):

\[
L(\theta) = \log P(+|t,c) + \sum_{i=1}^{k} \log P(-|t,w_i)
\]
\[
= \log \sigma(c \cdot t) + \sum_{i=1}^{k} \log \sigma(-w_i \cdot t)
\]
\[
= \log \frac{1}{1 + e^{-c \cdot t}} + \sum_{i=1}^{k} \log \frac{1}{1 + e^{-w_i \cdot t}}
\]

Summary: How to learn skip-gram embeddings

- Start with \( V \) random 300-dimensional vectors as initial embeddings
- Use logistic regression, the second most basic classifier used in machine learning after naïve bayes
- Take a corpus and take pairs of words that co-occur as positive examples
- Take pairs of words that don’t co-occur as negative examples
- Train the classifier to distinguish these by slowly adjusting all the embeddings to improve the classifier performance
- Throw away the classifier code and keep the embeddings.

(Dense) Word embeddings you can download!

- **Word2vec** [Mikolov et al.](https://code.google.com/archive/p/word2vec/)
- **Fasttext** [http://www.fasttext.cc/](http://www.fasttext.cc/)
- **Glove** [Pennington, Socher, Manning](http://nlp.stanford.edu/projects/glove/)
Evaluating embeddings

• Compare to human scores on word similarity-type tasks:
  • WordSim-353 (Finkelstein et al., 2002)
  • Stanford Contextual Word Similarity (SCWS) dataset (Huang et al., 2012)

• TOEFL dataset:
  • Levied is closest in meaning to:
    • imposed, believed, requested, correlated

Properties of embeddings

• Nearest words to some embeddings (Mikolov et al. 2013)

<table>
<thead>
<tr>
<th>Target</th>
<th>Redmond</th>
<th>Havel</th>
<th>ninjutsu</th>
<th>graffiti</th>
<th>capitulate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redmond</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Havel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ninjutsu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>graffiti</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>capitulate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Properties of embeddings

• Similarity depends on window size $C$

- $C = \pm 2$ The nearest words to Hogwarts:
  • Sunnydale
  • Evernight
- $C = \pm 5$ The nearest words to Hogwarts:
  • Dumbledore
  • Malfoy
  • halfblood

Analogy: Embeddings capture relational meaning!

- $\text{vector('king')} - \text{vector('man')} + \text{vector('woman')} \approx \text{vector('queen')}$
- $\text{vector('Paris')} - \text{vector('France')} + \text{vector('Italy')} \approx \text{vector('Rome')}$
Embeddings can help study word history!

- Train embeddings on old books to study changes in word meaning!!

Diachronic word embeddings for studying language change!

Visualizing changes

The evolution of sentiment words

Embeddings and bias

Embeddings reflect cultural bias

> Bolukbasi, Tolga, Kai-Wei Chang, Jamie Y. Zou, Venkatesh Saligrama, and Adam T. Kalai. “Man is to computer programmer as woman is to homemaker? Debiasing word embeddings.” In Advances in Neural Information Processing Systems, pp. 4349-4357. 2016.
Implicit Association test (Greenwald et al. 1998):
- How associated are concepts (flowers, insects) & attributes (pleasantness, unpleasantness)?
- Studied by measuring timing latencies for categorization.

Psychological findings on US participants:
- African-American names are associated with unpleasant words (more than European-American names)
- Male names associated more with math, female names with arts
- Old people’s names with unpleasant words, young people with pleasant words.

European American names are associated with unpleasant words (more than European-American names)
- Old people’s names with unpleasant words, young people with pleasant words.

Schiebinger, Leroy, Shaniqua

Change in linguistic framing 1910-1990
Change in association of Chinese names with adjectives framed as “othering” (barbaric, monstrous, biased)

Studied by measuring timing latencies for categorization.

Psychological findings on US participants:
- African-American names are associated with unpleasant words (more than European-American names)
- Old people’s names with unpleasant words, young people with pleasant words.

Garg, Nikhil, Aylin, Joanna J.

In this section we describe the datasets, embeddings, and word lists used,

SI Appendix, section B.1

Change in linguistic framing 1910-1990
Change in association of Chinese names with adjectives framed as “othering” (barbaric, monstrous, biased)

Stereotypes Beyond Census Data

Embeddings reflect cultural bias

Psychological findings on US participants:
- African-American names are associated with unpleasant words (more than European-American names)
- Old people’s names with unpleasant words, young people with pleasant words.

Schiebinger, Leroy, Shaniqua

A common challenge in historical analysis is that the written text in, say 1910, may not completely reflect the popular social occupations/adjectives and the intervals are tight. Similarly, for example, in word lists and types of measurements to demonstrate recall. For

SI Appendix, section A.5

A common challenge in historical analysis is that the written text in, say 1910, may not completely reflect the popular social

African American names (Ling, Shangpae) had a higher cosine with pleasant words (abuse, stunt, ugly)

European American names (Brad, Greg, Courtney) had a higher cosine with pleasant words (love, peace, mirror)

Embeddings reflect and replicate all sorts of pernicious biases.

In standard quantitative social science, machine learning is

In this section we describe the datasets, embeddings, and word lists used,

SI Appendix, section B.1

Change in linguistic framing 1910-1990
Change in association of Chinese names with adjectives framed as “othering” (barbaric, monstrous, biased)

Stereotypes Beyond Census Data

Changes in framing: adjectives associated with Chinese

<table>
<thead>
<tr>
<th>Adjective</th>
<th>1910</th>
<th>1950</th>
<th>1980</th>
</tr>
</thead>
<tbody>
<tr>
<td>Irresponsible</td>
<td>Immature</td>
<td>Unreliable</td>
<td>Unreliable</td>
</tr>
<tr>
<td>Embracing</td>
<td>Respectful</td>
<td>Admiring</td>
<td>Admiring</td>
</tr>
<tr>
<td>Barbarian</td>
<td>Savagery</td>
<td>Barbarism</td>
<td>Barbarism</td>
</tr>
<tr>
<td>Aggressive</td>
<td>Active</td>
<td>Dominant</td>
<td>Dominant</td>
</tr>
<tr>
<td>Transparent</td>
<td>Unobtrusive</td>
<td>Unnoticeable</td>
<td>Unnoticeable</td>
</tr>
<tr>
<td>Microminous</td>
<td>Unimportant</td>
<td>Unnecessary</td>
<td>Unnecessary</td>
</tr>
<tr>
<td>Hateful</td>
<td>Vicious</td>
<td>Vile</td>
<td>Vile</td>
</tr>
<tr>
<td>Cruel</td>
<td>Dishonest</td>
<td>Dishonest</td>
<td>Dishonest</td>
</tr>
<tr>
<td>Greedy</td>
<td>Greedy</td>
<td>Predatory</td>
<td>Predatory</td>
</tr>
<tr>
<td>Bizarre</td>
<td>Bizarre</td>
<td>Unusual</td>
<td>Unusual</td>
</tr>
</tbody>
</table>

Directions

- Debiasing algorithms for embeddings
- Use embeddings as a historical tool to study bias