CS4120: Natural Language Processing
Instructor: Prof. Lu Wang
Northeastern University
Webpage: www.ccs.neu.edu/home/luwang

Outline
• Vector Semantics
  • Sparse representation
    • Pointwise Mutual Information (PMI)
  • Dense representation
    • Singular Value Decomposition (SVD)
    • Neural Language Model

Sparse versus dense vectors
• PPMI vectors are
  • long (length |V| = 20,000 to 50,000)
  • sparse (most elements are zero)

Sparse versus dense vectors
• PPMI vectors are
  • long (length |V| = 20,000 to 50,000)
  • sparse (most elements are zero)
• Alternative: learn vectors which are
  • short (length 200-1000)
  • dense (most elements are non-zero)

Two methods for getting short dense vectors
• Singular Value Decomposition (SVD)
• ”Neural Language Model” – inspired by predictive models
Singular Value Decomposition (SVD)

Rank of a Matrix

• What is the rank of a matrix $A$?

• Number of linearly independent columns of $A$

$$A = \begin{bmatrix} 1 & 2 & 1 \\ -2 & -3 & 1 \\ 3 & 5 & 0 \end{bmatrix}$$

• Rank is 2
• We can rewrite $A$ as two “basis” vectors: $[1 2 1]$ and $[-2 -3 1]$

Rank as “Dimensionality”

• Think of point positions as a matrix:

$$\begin{bmatrix} -2 & -3 & 1 \\ 3 & 5 & 0 \end{bmatrix}$$

• Rewrite the coordinates in a more efficient way!
• Old basis vectors: $[1 0 0]$, $[0 1 0]$, $[0 0 1]$
• New basis vectors: $[1 2 1]$, $[-2 -3 1]$
Intuition of Dimensionality Reduction

- Approximate an N-dimensional dataset using fewer dimensions
- By first rotating the axes into a new space
- In which the highest order dimension captures the most variance in the original dataset
- And the next dimension captures the next most variance, etc.

Sample Dimensionality Reduction

Singular Value Decomposition

Any rectangular $w \times c$ matrix $X$ equals the product of 3 matrices:

- $W$: rows corresponding to original but $m$ columns represents a dimension in a new latent space, such that
  - $m$ column vectors are orthogonal to each other
  - Columns are ordered by the amount of variance in the dataset each new dimension accounts for

- $S$: diagonal $m \times m$ matrix of singular values expressing the importance of each dimension.

Sample Dimensionality Reduction

Singular Value Decomposition

(assuming the matrix has rank $m$, $m+c$)
Singular Value Decomposition

Any rectangular $w \times c$ matrix $X$ equals the product of 3 matrices:

$X = W R C$

$W$: rows corresponding to original but $m$ columns represents a dimension in a new latent space, such that:
- $m$ column vectors are orthogonal to each other
- Columns are ordered by the amount of variance in the dataset each new dimension accounts for

$R$: diagonal $m \times m$ matrix of singular values expressing the importance of each dimension.

$C$: columns corresponding to original but $m$ rows corresponding to singular values

---

SVD applied to term-document matrix:

Latent Semantic Analysis

- If instead of keeping all $m$ dimensions, we just keep the top $k$ singular values.
- Let's say 300.
- Each row of $W$ (keeping $k$ columns of the original $W$):
  - A $k$-dimensional vector
  - Representing word $w$

---

SVD on Term-Document Matrix: Example

- The matrix $X$

<table>
<thead>
<tr>
<th>word</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$d_3$</th>
<th>$d_4$</th>
<th>$d_5$</th>
<th>$d_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ship</td>
<td>1.00</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>boat</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>ocean</td>
<td>1.00</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>wood</td>
<td>1.00</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>tree</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

---

Matrix $W$

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>ship</td>
<td>-0.04</td>
<td>-0.30</td>
<td>0.30</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>boat</td>
<td>-0.30</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>ocean</td>
<td>-0.30</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>wood</td>
<td>-0.30</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>tree</td>
<td>-0.30</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Reduce dimension: The Matrix W

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>ship</td>
<td>-0.44</td>
<td>-0.30</td>
<td>0.57</td>
<td>0.58</td>
<td>0.25</td>
</tr>
<tr>
<td>boat</td>
<td>-0.13</td>
<td>-0.33</td>
<td>-0.59</td>
<td>0.00</td>
<td>0.73</td>
</tr>
<tr>
<td>ocean</td>
<td>-0.48</td>
<td>-0.51</td>
<td>-0.37</td>
<td>0.00</td>
<td>-0.61</td>
</tr>
<tr>
<td>wood</td>
<td>-0.70</td>
<td>0.35</td>
<td>0.15</td>
<td>-0.58</td>
<td>0.16</td>
</tr>
<tr>
<td>tree</td>
<td>-0.26</td>
<td>0.65</td>
<td>-0.41</td>
<td>0.58</td>
<td>-0.09</td>
</tr>
</tbody>
</table>

Reduce dimension: The Matrix S

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>ship</td>
<td>2.16</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>boat</td>
<td>1.59</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>ocean</td>
<td>0.00</td>
<td>1.28</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>wood</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>tree</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.39</td>
</tr>
</tbody>
</table>

Reduce dimension: The Matrix C

<table>
<thead>
<tr>
<th></th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
<th>d6</th>
<th>d7</th>
</tr>
</thead>
<tbody>
<tr>
<td>ship</td>
<td>-0.75</td>
<td>-0.38</td>
<td>-0.50</td>
<td>-0.45</td>
<td>-0.31</td>
<td>-0.12</td>
<td>0.00</td>
</tr>
<tr>
<td>boat</td>
<td>-0.29</td>
<td>-0.53</td>
<td>-0.19</td>
<td>0.63</td>
<td>0.22</td>
<td>0.41</td>
<td>0.00</td>
</tr>
<tr>
<td>ocean</td>
<td>0.28</td>
<td>0.76</td>
<td>0.45</td>
<td>-0.20</td>
<td>-0.12</td>
<td>-0.33</td>
<td>0.00</td>
</tr>
<tr>
<td>wood</td>
<td>0.00</td>
<td>0.00</td>
<td>0.58</td>
<td>0.00</td>
<td>-0.58</td>
<td>0.58</td>
<td>0.00</td>
</tr>
<tr>
<td>tree</td>
<td>-0.53</td>
<td>0.29</td>
<td>0.63</td>
<td>0.19</td>
<td>0.41</td>
<td>-0.22</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Reduce dimension: The Matrix W

<table>
<thead>
<tr>
<th></th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
<th>d6</th>
<th>d7</th>
</tr>
</thead>
<tbody>
<tr>
<td>ship</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>boat</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>ocean</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>wood</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>tree</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Similarity between ship and boat or ship and wood?
More details

- 300 dimensions are commonly used
- The cells are commonly weighted by a product of two weights (TF-IDF)
- Local weight: term frequency (or log version)
- Global weight: idf

Let’s return to PPMI word-word matrices

- Can we apply SVD to them?

SVD applied to term-term matrix

\[
X = W C
\]

(assuming the matrix has rank |V|, may not be true)

Truncated SVD on term-term matrix

\[
X = W C
\]

Truncated SVD produces embeddings

- Each row of W matrix is a k-dimensional representation of each word w
- K might range from 50 to 1000
- Generally we keep the top k dimensions, but some experiments suggest that getting rid of the top 1 dimension or even the top 50 dimensions is helpful (Lapesa and Evert 2014).
Embeddings versus sparse vectors

• Dense SVD embeddings sometimes work better than sparse PPMI matrices at tasks like word similarity
  • Denoising: low-order dimensions may represent unimportant information
  • Truncation may help the models generalize better to unseen data.
  • Having a smaller number of dimensions may make it easier for classifiers to properly weight the dimensions for the task.
  • Dense models may do better at capturing higher order co-occurrence.