Roadmap

1. Background

2. Motivation of GloVe

3. What is GloVe? How does it work?

4. Results

5. Conclusion and Take-Aways
Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways
Word Embedding Algos

1. Matrix factorization methods
   ● LSA, HAL, PPMI, HPCA
Word Embedding Algos

1. Matrix factorization methods
   - LSA, HAL, PPMI, HPCA

2. Local context window methods
   - Bengio 2003, C&W 2008/2011, skip-gram & CBOW (aka word2vec)
Matrix Factorization Methods

- Co-occurrence counts = “latent semantics”
Matrix Factorization Methods

- Co-occurrence counts = “latent semantics”
- Latent semantic analysis (LSA):
  1. SVD factorization: $C = U \Sigma V^T$
  2. Low-rank approximation: $C_k = U_k \Sigma_k V_k^T$
Matrix Factorization Methods

- Co-occurrence counts = “latent semantics”
- Latent semantic analysis (LSA):
  1. SVD factorization: $C = U\Sigma V^T$
  2. Low-rank approximation: $C_k = U\Sigma_k V^T$
- Good approximation: the largest $k$ eigenvalues matter a lot more than the smaller ones
Matrix Factorization Methods

- Co-occurrence counts = “latent semantics”
- Latent semantic analysis (LSA):
  1. SVD factorization: $C = U\Sigma V^T$
  2. Low-rank approximation: $C_k = U\Sigma_k V^T$
- Good approximation: the largest $k$ eigenvalues matter a lot more than the smaller ones
- Useful for semantics: $C_k$ models co-occurrence counts
## LSA

<table>
<thead>
<tr>
<th></th>
<th>doc1</th>
<th>doc2</th>
<th>doc3</th>
<th>doc4</th>
<th>doc5</th>
<th>doc6</th>
</tr>
</thead>
<tbody>
<tr>
<td>ship</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>boat</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ocean</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>voyage</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>trip</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
### LSA

<table>
<thead>
<tr>
<th></th>
<th>doc1</th>
<th>doc2</th>
<th>doc3</th>
<th>doc4</th>
<th>doc5</th>
<th>doc6</th>
</tr>
</thead>
<tbody>
<tr>
<td>ship</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>boat</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ocean</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>voyage</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>trip</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\langle \text{doc1, doc2} \rangle = 1 \times 0 + 0 \times 1 + 1 \times 1 + 1 \times 0 + 0 \times 0 = 1
\]
### LSA

<table>
<thead>
<tr>
<th></th>
<th>doc1</th>
<th>doc2</th>
<th>doc3</th>
<th>doc4</th>
<th>doc5</th>
<th>doc6</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-1.62</td>
<td>-0.6</td>
<td>-0.44</td>
<td>-0.97</td>
<td>-0.7</td>
<td>-0.26</td>
</tr>
<tr>
<td>B</td>
<td>-0.46</td>
<td>-0.84</td>
<td>-0.30</td>
<td>1</td>
<td>0.35</td>
<td>0.65</td>
</tr>
</tbody>
</table>
### LSA

The inner product of documents 1 and 2 is calculated as follows:

\[
<\text{doc1}, \text{doc2}> = (-1.62)(-0.6) + (-0.46)(-0.84) = 1.36
\]

<table>
<thead>
<tr>
<th></th>
<th>doc1</th>
<th>doc2</th>
<th>doc3</th>
<th>doc4</th>
<th>doc5</th>
<th>doc6</th>
</tr>
</thead>
<tbody>
<tr>
<td>doc1</td>
<td>-1.62</td>
<td>-0.6</td>
<td>-0.44</td>
<td>-0.97</td>
<td>-0.7</td>
<td>-0.26</td>
</tr>
<tr>
<td>doc2</td>
<td>-0.46</td>
<td>-0.84</td>
<td>-0.30</td>
<td>1</td>
<td>0.35</td>
<td>0.65</td>
</tr>
</tbody>
</table>
Matrix Factorization Methods

- Term-term matrix methods: HAL, COALS, PPMI, HPCA
Matrix Factorization Methods

- Term-term matrix methods: HAL, COALS, PPMI, HPCA
- Takes advantage of global corpus stats
Matrix Factorization Methods

- Term-term matrix methods: HAL, COALS, PPMI, HPCA
- Takes advantage of global corpus stats
- Not the best approach for word embeddings (but often a reasonable baseline)
Local Context Window Methods

\[ Pr[w|context] = Pr[w_t|w_{t-1}, w_{t-2}, \ldots, w_{t-n+1}] \]
Local Context Window Methods

\[ Pr[w|\text{context}] = Pr[w_t|w_{t-1}, w_{t-2}, \ldots, w_{t-n+1}] \]

- Bengio, 2003 - neural language model
Local Context Window Methods

\[ Pr[w|\text{context}] = Pr[w_t|w_{t-1}, w_{t-2}, \ldots, w_{t-n+1}] \]

- Bengio, 2003 - neural language model
- Learning word representations stored lookup table/matrix or network weights
Word2Vec (Mikolov, 2013)

- **CBOW**
- **Skip-gram**
Local Context Window Methods

- Tailored to the task of learning useful embeddings
Local Context Window Methods

- Tailored to the task of learning useful embeddings
- Explicitly penalize models that poorly predict contexts given words (or words given contexts)
Local Context Window Methods

- Tailored to the task of learning useful embeddings
- Explicitly penalize models that poorly predict contexts given words (or words given contexts)
- Don’t utilize global corpus statistics
Local Context Window Methods

- Tailored to the task of learning useful embeddings
- Explicitly penalize models that poorly predict contexts given words (or words given contexts)
- Don’t utilize global corpus statistics
- Intuitively, a more globally-aware model should be able to do better
Good Embedding Spaces have Linear Substructure
Linear Substructure

- Want to “capture the meaningful linear substructures” prevalent in the embedding space
Linear Substructure

-Want to “capture the meaningful linear substructures” prevalent in the embedding space.
-Analogy tasks reflect linear relationships between words in the embedding space.
Linear Substructure

- Want to “capture the meaningful linear substructures” prevalent in the embedding space.
- Analogy tasks reflect linear relationships between words in the embedding space.

Paris - France + Germany = Berlin
Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways
Roadmap

1. Background

2. Motivation of GloVe

3. What is GloVe?

4. Results

5. Conclusion and Take-Aways
Motivation

- Learn embeddings useful for downstream tasks and outperform word2vec
Motivation

- Learn embeddings useful for downstream tasks and outperform word2vec
- Take advantage of global stats
Motivation

● Learn embeddings useful for downstream tasks and outperform word2vec
● Take advantage of global stats
● Analogies need linear substructure
Motivation

- Learn embeddings useful for downstream tasks and outperform word2vec
- Take advantage of global stats
- Analogies need linear substructure
- Embedding algos should exploit this substructure
Observation 1: Linear Substructure

- Analogy property is linear
Observation 1: Linear Substructure

- Analogy property is linear
- Vector differences seem to encode concepts
Observation 1: Linear Substructure

- Analogy property is linear
- Vector differences seem to encode concepts
- Man – woman should encode concept of gender
Observation 1: Linear Substructure

- Analogy property is linear
- Vector differences seem to encode concepts
- Man – woman should encode concept of gender
- France – Germany should encode them being different countries
Co-occurrence Matrix

$X$

$N = |V|$
Co-occurrence Matrix

$W_a = \text{France}$

$W_b = \text{Germany}$
Co-occurrence Matrix

$W_k = \text{Paris}$

$W_a = \text{France}$

$W_b = \text{Germany}$
Co-occurrence Matrix

$W_k = \text{Paris}$

$X_{ak} = \text{Num times } k \text{ appear in contexts with } a$

$X_a = \text{Num contexts with } a$

$W_a = \text{France}$

$W_b = \text{Germany}$
Observation 2: Co-Occurrence Ratios Matter
### Observation 2: Co-Occurrence Ratios Matter

<table>
<thead>
<tr>
<th>k = paris</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr[k</td>
<td>france]</td>
</tr>
<tr>
<td>Pr[k</td>
<td>germany]</td>
</tr>
<tr>
<td>Pr[k</td>
<td>france]/Pr[k</td>
</tr>
<tr>
<td></td>
<td>( k = \text{paris} )</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>( \Pr[k \mid \text{france}] )</td>
<td>large</td>
</tr>
<tr>
<td>( \Pr[k \mid \text{germany}] )</td>
<td>small</td>
</tr>
<tr>
<td>( \frac{\Pr[k \mid \text{france}]}{\Pr[k \mid \text{germany}]} )</td>
<td>large</td>
</tr>
</tbody>
</table>
**Observation 2: Co-Occurrence Ratios Matter**

<table>
<thead>
<tr>
<th></th>
<th>k = paris</th>
<th>k = berlin</th>
<th>k = europe</th>
</tr>
</thead>
<tbody>
<tr>
<td>**Pr[k</td>
<td>france]**</td>
<td>large</td>
<td>small</td>
</tr>
<tr>
<td>**Pr[k</td>
<td>germany]**</td>
<td>small</td>
<td>large</td>
</tr>
<tr>
<td>**Pr[k</td>
<td>france]/Pr[k</td>
<td>germany]**</td>
<td>large</td>
</tr>
</tbody>
</table>
# Observation 2: Co-Occurrence Ratios Matter

<table>
<thead>
<tr>
<th></th>
<th>k = paris</th>
<th>k = berlin</th>
<th>k = europe</th>
<th>k = ostrich</th>
</tr>
</thead>
<tbody>
<tr>
<td>**Pr[k</td>
<td>france]**</td>
<td>large</td>
<td>small</td>
<td>large</td>
</tr>
<tr>
<td>**Pr[k</td>
<td>germany]**</td>
<td>small</td>
<td>large</td>
<td>large</td>
</tr>
<tr>
<td>**Pr[k</td>
<td>france] / Pr[k</td>
<td>germany]**</td>
<td>large</td>
<td>small</td>
</tr>
</tbody>
</table>
Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways
Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways
GloVe

- Model using a loss function that leverages:
  1. Global co-occurrence counts and their ratios
  2. Linear substructure for analogies
GloVe

- Model using a loss function that leverages:
  1. Global co-occurrence counts and their ratios
  2. Linear substructure for analogies
- Software package to build embedding models
GloVe

- Model using a loss function that leverages:
  1. Global co-occurrence counts and their ratios
  2. Linear substructure for analogies
- Software package to build embedding models
- Downloadable pre-trained word vectors created using a massive corpus
Deriving the GloVe model

- Co-occurrence values in matrix X should be the starting point
Deriving the GloVe model

- Co-occurrence values in matrix X should be the starting point

\[
F(w_a, w_b, w_k) = \frac{P[k|a]}{P[k|b]} = \frac{X_{ak}}{X_{ak}/X_a} = \frac{X_{ak}}{X_{bk}/X_b}
\]
Deriving the GloVe model

- Co-occurrence values in matrix X should be the starting point

\[ F(w_a, w_b, w_k) = \frac{P[k|a]}{P[k|b]} = \frac{X_{ak}/X_a}{X_{bk}/X_b} \]

- Suppose \( F(\text{france, germany, paris}) = \text{small} \)
Deriving the GloVe model

- Co-occurrence values in matrix $X$ should be the starting point

$$F(w_a, w_b, w_k) = \frac{P[k|a]}{P[k|b]} = \frac{X_{ak}/X_a}{X_{bk}/X_b}$$

- Suppose $F(\text{france, germany, paris}) = \text{small}$
- $\rightarrow$ update the vectors
Factoring in Vector Differences

- The difference between the France and Germany vectors is what matters with w.r.t analogies
Factoring in Vector Differences

- The difference between the France and Germany vectors is what matters with w.r.t analogies

\[ F(w_a - w_b, w_k) = \frac{P[k|a]}{P[k|b]} \]
\[ F(w_a - w_b, w_k) = \frac{P[k|a]}{P[k|b]} \]
Real-Valued Input is Less Complicated
Real-Valued Input is Less Complicated

\[ F((w_a - w_b)^T w_k) = F(w_a \cdot w_k - w_b \cdot w_k) = \frac{P[k|a]}{P[k|b]} \]
\[ F((w_a - w_b)^T w_k) = F(w_a \cdot w_k - w_b \cdot w_k) = \frac{P[k|a]}{P[k|b]} \]
Reframing with Softmax

\[ F(w_a \cdot w_k - w_b \cdot w_k) = \frac{P[k|a]}{P[k|b]} = \frac{\exp(w_a \cdot w_k)}{\exp(w_b \cdot w_k)} \]
Reframing with Softmax

\[ F(w_a \cdot w_k - w_b \cdot w_k) = \frac{P[k|a]}{P[k|b]} = \frac{\exp(w_a \cdot w_k)}{\exp(w_b \cdot w_k)} \]

\[ \Rightarrow \]

\[ w_a \cdot w_k = \log(P[k|a]) \]
Reframing with Softmax

\[
F(w_a \cdot w_k - w_b \cdot w_k) = \frac{P[k|a]}{P[k|b]} = \frac{\exp(w_a \cdot w_k)}{\exp(w_b \cdot w_k)}
\]

\[\Rightarrow\]

\[w_a \cdot w_k = \log(P[k|a])\]

\[= \log\left(\frac{X_{ak}}{X_a}\right)\]
Reframing with Softmax

\[
F(w_a \cdot w_k - w_b \cdot w_k) = \frac{P[k|a]}{P[k|b]} = \frac{\exp(w_a \cdot w_k)}{\exp(w_b \cdot w_k)}
\]

\[\Rightarrow\]

\[w_a \cdot w_k = \log(P[k|a]) = \log\left(\frac{X_{ak}}{X_a}\right) = \log(X_{ak}) - \log(X_a)\]
Reframing with Softmax

\[ F(w_a \cdot w_k - w_b \cdot w_k) = \frac{P[k|a]}{P[k|b]} = \frac{\exp(w_a \cdot w_k)}{\exp(w_b \cdot w_k)} \]

\[ \implies w_a \cdot w_k = \log(P[k|a]) \]

\[ = \log\left(\frac{X_{ak}}{X_a}\right) \]

\[ = \log(X_{ak}) - \log(X_a) \]

\[ \implies w_a \cdot w_k + \text{bias} - \log(X_{ak}) = 0 \]
Least Squares Problem
Least Squares Problem

\[ w_a \cdot w_k + \text{bias} - \log(X_{ak}) = 0 \]
Least Squares Problem

\[ w_a \cdot w_k + bias - \log(X_{ak}) = 0 \]

\[ \Rightarrow \]

\[ (w_i \cdot w_j + biases - \log(X_{ij}))^2 \]
Final GloVe Model

\[ J = \sum_{i,j=1}^{V} f(X_{ij}) \left( w_i^T \tilde{v}_j + b_i + \tilde{b}_j - \log X_{ij} \right)^2 \]
Final GloVe Model

\[ J = \sum_{i,j=1}^{V} f(X_{ij}) \left( w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij} \right)^2 \]
Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways
Roadmap

1. Background

2. Motivation of GloVe

3. What is GloVe?

4. Results

5. Conclusion and Take-Aways
Performance on Analogy Tasks

- Semantic: “Paris is France as Berlin is to ____”
- Syntactic: “Fly is to flying as dance is to ____”
## Performance on Analogy Tasks

<table>
<thead>
<tr>
<th>Model</th>
<th>Dimensions</th>
<th>Corpus Size</th>
<th>Semantic</th>
<th>Syntactic</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBOW</td>
<td>1000</td>
<td>6B</td>
<td>57.3</td>
<td>68.9</td>
<td>63.7</td>
</tr>
<tr>
<td>Skip-Gram</td>
<td>1000</td>
<td>GB</td>
<td>66.1</td>
<td>65.1</td>
<td>65.6</td>
</tr>
<tr>
<td>SVD-L</td>
<td>300</td>
<td>42B</td>
<td>38.4</td>
<td>58.2</td>
<td>49.2</td>
</tr>
<tr>
<td>GloVe</td>
<td>300</td>
<td>42B</td>
<td><strong>81.9</strong></td>
<td><strong>69.3</strong></td>
<td><strong>75.0</strong></td>
</tr>
</tbody>
</table>
Speed

![Graphs showing training time vs. accuracy for different models and negative samples.](image)

- **Left Graph**: Accuracy [%] vs. Iterations (GloVe) and Negative Samples (CBOW).
  - Orange line: GloVe
  - Blue line: CBOW

- **Right Graph**: Accuracy [%] vs. Training Time (hrs) and Negative Samples (Skip-Gram).
  - Orange line: GloVe
  - Green line: Skip-Gram
Limitations

- Superior for analogy tasks; almost the same performance as word2vec and fastText for retrieval tasks
Limitations

● Superior for analogy tasks; almost the same performance as word2vec and fastText for retrieval tasks
● Co-occurrence construction takes 1 hr +
Limitations

- Superior for analogy tasks; almost the same performance as word2vec and fastText for retrieval tasks
- Co-occurrence construction takes 1 hr +
- Each training iterations takes 10 minutes +
Limitations

- Superior for analogy tasks; almost the same performance as word2vec and fastText for retrieval tasks
- Co-occurrence construction takes 1 hr +
- Each training iterations takes 10 minutes +
- Not online
Limitations

- Superior for analogy tasks; almost the same performance as word2vec and fastText for retrieval tasks
- Co-occurrence construction takes 1 hr +
- Each training iterations takes 10 minutes +
- Not online
- Doesn’t allow different entities to be embedded (a la StarSpace)
Limitations

- Superior for analogy tasks; almost the same performance as word2vec and fastText for retrieval tasks
- Co-occurrence construction takes 1 hr +
- Each training iterations takes 10 minutes +
- Not online
- Doesn’t allow different entities to be embedded (a la StarSpace)
Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways
Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways
Conclusion and Take-Aways

- Embeddings algos all take advantage of co-occurrence stats
Conclusion and Take-Aways

- Embeddings algos all take advantage of co-occurrence stats
- Leveraging global stats can provide performance boost
Conclusion and Take-Aways

- Embeddings algos all take advantage of co-occurrence stats
- Leveraging global stats can provide performance boost
- Keep the linear substructure in mind when designing embedding algorithms
Conclusion and Take-Aways

- Embeddings algs all take advantage of co-occurrence stats
- Leveraging global stats can provide performance boost
- Keep the linear substructure in mind when designing embedding algorithms
- Simpler models can work well (SVD-L performed very well)
Conclusion and Take-Aways

- Embeddings algos all take advantage of co-occurrence stats
- Leveraging global stats can provide performance boost
- Keep the linear substructure in mind when designing embedding algorithms
- Simpler models can work well (SVD-L performed very well)
- More iterations seems to be most important for embedding models:
  - faster iterations $\rightarrow$ train on a larger corpus $\rightarrow$ create better embeddings
- Demonstrated by word2vec, GloVe, and SVD-L
Questions?