Distributed Representations

CMSC 473/673
UMBC

Some slides adapted from 3SLP
Outline

Recap
- Maxent models
- Basic neural language models

Continuous representations
- Motivation
- Key idea: represent words with vectors
- Two common counting types

Two (four) common continuous representation models

Evaluation
Maxent Objective: Log-Likelihood

\[
\log \prod_i p_\theta(x_i | y_i) = \sum_i \log p_\theta(x_i | y_i)
\]

\[
= \sum_i \theta^T f(x_i, y_i) - \log Z(y_i)
\]

\[
= F(\theta)
\]

The objective is implicitly defined with respect to (wrt) your data on hand.

Differentiating this becomes nicer (even though Z depends on \(\theta\)).
Log-Likelihood Gradient

Each component $k$ is the difference between:

the total value of feature $f_k$ in the training data

and

the total value the current model $p_\theta$ thinks it computes for feature $f_k$
N-gram Language Models

given some context...

compute beliefs about what is likely...

predict the next word

\[ p(w_i | w_{i-3}, w_{i-2}, w_{i-1}) \propto \text{count}(w_{i-3}, w_{i-2}, w_{i-1}, w_i) \]
Maxent Language Models

given some context...

compute beliefs about what is likely...

\[ p(w_i | w_{i-3}, w_{i-2}, w_{i-1}) \propto \text{softmax}(\theta \cdot f(w_{i-3}, w_{i-2}, w_{i-1}, w_i)) \]

predict the next word
Neural Language Models

given some context...

create/use
“distributed representations”...

combine these representations...

compute beliefs about what is likely...

\[ p(w_i \mid w_{i-3}, w_{i-2}, w_{i-1}) \propto \text{softmax}(\theta w_i \cdot f(w_{i-3}, w_{i-2}, w_{i-1})) \]

predict the next word
Neural Language Models

given some context...

create/use “distributed representations”...

combine these representations...

compute beliefs about what is likely...

\[ p(w_i | w_{i-3}, w_{i-2}, w_{i-1}) \propto \text{softmax}(\theta_{w_i} \cdot f(w_{i-3}, w_{i-2}, w_{i-1})) \]

predict the next word
Outline

Recap
  Maxent models
  Basic neural language models

Continuous representations
  Motivation
    Key idea: represent words with vectors
    Two common counting types

Two (four) common continuous representation models

Evaluation
How have we represented words?

Each word is a distinct item

- Bijection between the strings and unique integer ids:
  - "cat" --> 3, "kitten" --> 792, "dog" --> 17394

Are "cat" and "kitten" similar?
How have we represented words?

Each word is a distinct item

Bijection between the strings and unique integer ids:
"cat" --> 3, "kitten" --> 792 "dog" --> 17394

Are "cat" and "kitten" similar?

Equivalently: "One-hot" encoding

Represent each word type w with a vector the size of the vocabulary
This vector has V-1 zero entries, and 1 non-zero (one) entry
Word Similarity ➔ Plagiarism Detection

MAINFRAMES

Mainframes are primarily referred to large computers with rapid, advanced processing capabilities that can execute and perform tasks equivalent to many Personal Computers (PCs) machines networked together. It is characterized with high quantity Random Access Memory (RAM), very large secondary storage devices, and high-speed processors to cater for the needs of the computers under its service.

Consisting of advanced components, mainframes have the capability of running multiple large applications required by many and most enterprises and organizations. This is one of its advantages. Mainframes are also suitable to cater for those applications (programs) or files that are of very high demand by its users (clients). Examples of such organizations and enterprises using mainframes are online shopping websites such as Amazon, Microsoft, etc.

MAINFRAMES

Mainframes usually are referred those computers with fast, advanced processing capabilities that could perform by itself tasks that may require a lot of Personal Computers (PC) Machines. Usually mainframes would have lots of RAMs, very large secondary storage devices, and very fast processors to cater for the needs of those computers under its service.

Due to the advanced components mainframes have, these computers have the capability of running multiple large applications required by most enterprises, which is one of its advantage. Mainframes are also suitable to cater for those applications or files that are of very large demand by its users (clients). Examples of these include the large online shopping websites -i.e. : Ebay, Amazon, Microsoft, etc.
Distributional models of meaning
= vector-space models of meaning
= vector semantics

Zellig Harris (1954):
“oculist and eye-doctor ... occur in almost the same environments”
“If A and B have almost identical environments we say that they are synonyms.”

Firth (1957):
“You shall know a word by the company it keeps!”
Continuous Meaning

The paper reflected the truth.
Continuous Meaning

The paper reflected the truth.
The paper reflected the truth.
(Some) Properties of Embeddings

Capture “like” (similar) words

<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 Wash.</td>
<td>Vaclav Havel</td>
<td>ninja</td>
<td>spray paint</td>
<td>graffiti</td>
<td>capitulation</td>
</tr>
<tr>
<td>Redmond Washington</td>
<td>president Vaclav Havel</td>
<td>martial arts</td>
<td>graffiti</td>
<td>capitulated</td>
<td></td>
</tr>
<tr>
<td>Microsoft</td>
<td>Velvet Revolution</td>
<td>swordsmanship</td>
<td>taggers</td>
<td>capitulating</td>
<td></td>
</tr>
</tbody>
</table>

Mikolov et al. (2013)
(Some) Properties of Embeddings

Capture “like” (similar) words

<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 Wash.</td>
<td>Vaclav Havel</td>
<td>ninja</td>
<td>spray paint</td>
<td>capitulation</td>
<td></td>
</tr>
<tr>
<td>Redmond Washington</td>
<td>president Vaclav Havel</td>
<td>martial arts</td>
<td>grafitti</td>
<td>capitolated</td>
<td></td>
</tr>
<tr>
<td>Microsoft</td>
<td>Velvet Revolution</td>
<td>swordsmanship</td>
<td>taggers</td>
<td>capitolating</td>
<td></td>
</tr>
</tbody>
</table>

Capture relationships

\[
\text{vector('king')} - \text{vector('man')} + \text{vector('woman')} \approx \text{vector('queen')} \\
\text{vector('Paris')} - \text{vector('France')} + \text{vector('Italy')} \approx \text{vector('Rome')}\\
\]

Mikolov et al. (2013)
Outline

Recap
  Maxent models
  Basic neural language models
Continuous representations
  Motivation
    Key idea: represent words with vectors
    Two common counting types
Two (four) common continuous representation models
Evaluation
Key Idea

1. Acquire basic contextual statistics (counts) for each word type $w$

2. Extract a real-valued vector $v$ for each word $w$ from those statistics

3. Use the vectors to represent each word in later tasks
Outline

Recap
  Maxent models
  Basic neural language models
Continuous representations
  Motivation
  Key idea: represent words with vectors
  Two common counting types
Two (four) common continuous representation models
Evaluation
“You shall know a word by the company it keeps!” Firth (1957)

document (↓)-word (→) count matrix

<table>
<thead>
<tr>
<th></th>
<th>battle</th>
<th>soldier</th>
<th>fool</th>
<th>clown</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>As You Like It</em></td>
<td>1</td>
<td>2</td>
<td>37</td>
<td>6</td>
</tr>
<tr>
<td><em>Twelfth Night</em></td>
<td>1</td>
<td>2</td>
<td>58</td>
<td>117</td>
</tr>
<tr>
<td><em>Julius Caesar</em></td>
<td>8</td>
<td>12</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><em>Henry V</em></td>
<td>15</td>
<td>36</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>
“You shall know a word by the company it keeps!” Firth (1957)

document (down)-word (right) count matrix

<table>
<thead>
<tr>
<th></th>
<th>battle</th>
<th>soldier</th>
<th>fool</th>
<th>clown</th>
</tr>
</thead>
<tbody>
<tr>
<td>As You Like It</td>
<td>1</td>
<td>2</td>
<td>37</td>
<td>6</td>
</tr>
<tr>
<td>Twelfth Night</td>
<td>1</td>
<td>2</td>
<td>58</td>
<td>117</td>
</tr>
<tr>
<td>Julius Caesar</td>
<td>8</td>
<td>12</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Henry V</td>
<td>15</td>
<td>36</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

basic bag-of-words counting

- it: 6
- l: 5
- the: 4
- to: 3
- and: 3
- seen: 2
- yet: 1
- would: 1
- whimsical: 1
- times: 1
- sweet: 1
- satirical: 1
- adventure: 1
- genre: 1
- fairy: 1
- humor: 1
- have: 1
- great: 1
“You shall know a word by the company it keeps!” Firth (1957)

document (↓)-word (→) count matrix

<table>
<thead>
<tr>
<th></th>
<th>battle</th>
<th>soldier</th>
<th>fool</th>
<th>clown</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>As You Like It</strong></td>
<td>1</td>
<td>2</td>
<td>37</td>
<td>6</td>
</tr>
<tr>
<td><strong>Twelfth Night</strong></td>
<td>1</td>
<td>2</td>
<td>58</td>
<td>117</td>
</tr>
<tr>
<td><strong>Julius Caesar</strong></td>
<td>8</td>
<td>12</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>Henry V</strong></td>
<td>15</td>
<td>36</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

Assumption: Two documents are similar if their vectors are similar
“You shall know a word by the company it keeps!” Firth (1957)

<table>
<thead>
<tr>
<th></th>
<th>battle</th>
<th>soldier</th>
<th>fool</th>
<th>clown</th>
</tr>
</thead>
<tbody>
<tr>
<td>As You Like It</td>
<td>1</td>
<td>2</td>
<td>37</td>
<td>6</td>
</tr>
<tr>
<td>Twelfth Night</td>
<td>1</td>
<td>2</td>
<td>58</td>
<td>117</td>
</tr>
<tr>
<td>Julius Caesar</td>
<td>8</td>
<td>12</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Henry V</td>
<td>15</td>
<td>36</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

Assumption: Two words are similar if their vectors are similar
“You shall know a word by the company it keeps!” Firth (1957)

![Document-word count matrix](image)

<table>
<thead>
<tr>
<th></th>
<th>battle</th>
<th>soldier</th>
<th>fool</th>
<th>clown</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>As You Like It</strong></td>
<td>1</td>
<td>2</td>
<td>37</td>
<td>6</td>
</tr>
<tr>
<td><strong>Twelfth Night</strong></td>
<td>1</td>
<td>2</td>
<td>58</td>
<td>117</td>
</tr>
<tr>
<td><strong>Julius Caesar</strong></td>
<td>8</td>
<td>12</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>Henry V</strong></td>
<td>15</td>
<td>36</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

**Assumption:** Two words are similar if their vectors are similar

**Issue:** Count word vectors are very large, sparse, and skewed!
"You shall know a word by the company it keeps!" Firth (1957)

**context** (↓)-**word** (→) count matrix

<table>
<thead>
<tr>
<th></th>
<th>apricot</th>
<th>pineapple</th>
<th>digital</th>
<th>information</th>
</tr>
</thead>
<tbody>
<tr>
<td>aardvark</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>computer</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>data</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>pinch</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>result</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>sugar</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

*Context: those other words within a small “window” of a target word*
“You shall know a word by the company it keeps!” Firth (1957)

context (\(\downarrow\))-word (\(\rightarrow\)) count matrix

<table>
<thead>
<tr>
<th></th>
<th>apricot</th>
<th>pineapple</th>
<th>digital</th>
<th>information</th>
</tr>
</thead>
<tbody>
<tr>
<td>aardvark</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>computer</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>data</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>pinch</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>result</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>sugar</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Context: those other words within a small “window” of a target word

A cloud [computer stores digital data on] a remote computer
“You shall know a word by the company it keeps!” Firth (1957)

**context** (↓)-**word** (→) count matrix

<table>
<thead>
<tr>
<th></th>
<th>apricot</th>
<th>pineapple</th>
<th>digital</th>
<th>information</th>
</tr>
</thead>
<tbody>
<tr>
<td>aardvark</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>computer</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>data</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>pinch</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>result</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>sugar</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The size of windows depends on your goals

The shorter the windows, the more **syntactic** the representation
± 1-3 more “syntax-y”

The longer the windows, the more **semantic** the representation
± 4-10 more “semantic-y”
“You shall know a word by the company it keeps!” Firth (1957)

context (↓-word (→) count matrix

<table>
<thead>
<tr>
<th></th>
<th>apricot</th>
<th>pineapple</th>
<th>digital</th>
<th>information</th>
</tr>
</thead>
<tbody>
<tr>
<td>aardvark</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>computer</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>data</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>pinch</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>result</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>sugar</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Context: those other words within a small “window” of a target word

Assumption: Two words are similar if their vectors are similar

Issue: Count word vectors are very large, sparse, and skewed!
Outline

Recap
  Maxent models
  Basic neural language models
Continuous representations
  Motivation
  Key idea: represent words with vectors
  Two common counting types
Two (four) common continuous representation models
Evaluation
Four kinds of vector models

Sparse vector representations
  1. Mutual-information weighted word co-occurrence matrices

Dense vector representations:
  2. Singular value decomposition/Latent Semantic Analysis
  3. Neural-network-inspired models (skip-grams, CBOW)
  4. Brown clusters
Shared Intuition

Model the meaning of a word by “embedding” in a vector space

The meaning of a word is a vector of numbers

Contrast: word meaning is represented in many computational linguistic applications by a vocabulary index (“word number 545”) or the string itself
What’s the Meaning of Life?
What’s the Meaning of Life?

LIFE’
What’s the Meaning of Life?

LIFE’

(.478, -.289, .897, ...)
“Embeddings” Did Not Begin In This Century


Deerwester et al. (1990): “Indexing by Latent Semantic Analysis”

Brown et al. (1992): “Class-based n-gram models of natural language”
Four kinds of vector models

Sparse vector representations

1. Mutual-information weighted word co-occurrence matrices

Dense vector representations:

2. Singular value decomposition/Latent Semantic Analysis
3. Neural-network-inspired models (skip-grams, CBOW)
4. Brown clusters

You already saw some of this in assignment 2 (question 2)!
Pointwise Mutual Information (PMI): Dealing with Problems of Raw Counts

Raw word frequency is not a great measure of association between words.

It’s very skewed: “the” and “of” are very frequent, but maybe not the most discriminative.

We’d rather have a measure that asks whether a context word is particularly informative about the target word.

(Positive) Pointwise Mutual Information ((P)PMI)
Pointwise Mutual Information (PMI): Dealing with Problems of Raw Counts

Raw word frequency is not a great measure of association between words.

It’s very skewed: “the” and “of” are very frequent, but maybe not the most discriminative.

We’d rather have a measure that asks whether a context word is particularly informative about the target word.

(Pointwise) Pointwise Mutual Information ((P)PMI)

Pointwise mutual information:

Do events $x$ and $y$ co-occur more than if they were independent?

$$PMI(x, y) = \log \frac{p(x, y)}{p(x)p(y)}$$
Pointwise Mutual Information (PMI): Dealing with Problems of Raw Counts

Raw word frequency is not a great measure of association between words.

It’s very skewed: “the” and “of” are very frequent, but maybe not the most discriminative.

We’d rather have a measure that asks whether a context word is particularly informative about the target word.

(Positive) Pointwise Mutual Information ((P)PMI)

Pointwise mutual information:

Do events x and y co-occur more than if they were independent?

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

PMI between two words: (Church & Hanks 1989)

Do words x and y co-occur more than if they were independent?
“Noun Classification from Predicate-Argument Structure,” Hindle (1990)

“drink it” is more common than “drink wine”

“wine” is a better “drinkable” thing than “it”

<table>
<thead>
<tr>
<th>Object of “drink”</th>
<th>Count</th>
<th>PMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>it</td>
<td>3</td>
<td>1.3</td>
</tr>
<tr>
<td>anything</td>
<td>3</td>
<td>5.2</td>
</tr>
<tr>
<td>wine</td>
<td>2</td>
<td>9.3</td>
</tr>
<tr>
<td>tea</td>
<td>2</td>
<td>11.8</td>
</tr>
<tr>
<td>liquid</td>
<td>2</td>
<td>10.5</td>
</tr>
</tbody>
</table>
Four kinds of vector models

Sparse vector representations
1. Mutual-information weighted word co-occurrence matrices

Dense vector representations:
2. Singular value decomposition/Latent Semantic Analysis
3. Neural-network-inspired models (skip-grams, CBOW)
4. Brown clusters

Learn more in:
• Your project
• Paper (673)
• Other classes (478/678)
Four kinds of vector models

Sparse vector representations

1. Mutual-information weighted word co-occurrence matrices

Dense vector representations:

2. Singular value decomposition/Latent Semantic Analysis
3. Neural-network-inspired models (skip-grams, CBOW)
4. Brown clusters
Brown clustering (Brown et al., 1992)

An agglomerative clustering algorithm that clusters words based on which words precede or follow them

These word clusters can be turned into a kind of vector (binary vector)
Brown Clusters as vectors

Build a binary tree from bottom to top based on how clusters are merged

Each word represented by binary string = path from root to leaf

Each intermediate node is a cluster

In practice, use an available implementation: https://github.com/percyliang/brown-cluster
Brown cluster examples
Outline

Recap
- Maxent models
- Basic neural language models

Continuous representations
- Motivation
- Key idea: represent words with vectors
- Two common counting types

Two (four) common continuous representation models

Evaluation
Evaluating Similarity

Extrinsic (task-based, end-to-end) Evaluation:
  Question Answering
  Spell Checking
  Essay grading

Intrinsic Evaluation:
  Correlation between algorithm and human word similarity ratings
    WordSim353: 353 noun pairs rated 0-10.
      \( \text{sim}(\text{plane}, \text{car}) = 5.77 \)
  
  Taking TOEFL multiple-choice vocabulary tests
Cosine: Measuring Similarity

Given 2 target words \( v \) and \( w \) how similar are their vectors?

**Dot product** or **inner product** from linear algebra

\[
\text{dot-product}(\vec{v}, \vec{w}) = \vec{v} \cdot \vec{w} = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \ldots + v_N w_N
\]

High when two vectors have large values in same dimensions, low for **orthogonal vectors** with zeros in complementary distribution
Cosine: Measuring Similarity

Given 2 target words $v$ and $w$ how similar are their vectors?

**Dot product** or **inner product** from linear algebra

$$\text{dot-product}(\vec{v}, \vec{w}) = \vec{v} \cdot \vec{w} = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \ldots + v_N w_N$$

High when two vectors have large values in same dimensions, low for **orthogonal vectors** with zeros in complementary distribution

Correct for high magnitude vectors

$$\frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| ||\vec{b}||}$$
Cosine Similarity

Divide the dot product by the length of the two vectors

\[
\frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \cdot ||\vec{b}||}
\]

This is the cosine of the angle between them

\[
\vec{a} \cdot \vec{b} = ||\vec{a}|| ||\vec{b}|| \cos \theta
\]

\[
\frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \cdot ||\vec{b}||} = \cos \theta
\]
Cosine as a similarity metric

-1: vectors point in opposite directions
+1: vectors point in same directions
0: vectors are orthogonal
Example: Word Similarity

\[ \cos(x, y) = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2} \sqrt{\sum y_i^2}} \]

<table>
<thead>
<tr>
<th></th>
<th>large</th>
<th>data</th>
<th>computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>apricot</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>digital</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>information</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>
Example: Word Similarity

\[
\cos(x, y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2} \sqrt{\sum_i y_i^2}}
\]

<table>
<thead>
<tr>
<th></th>
<th>large</th>
<th>data</th>
<th>computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>apricot</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>digital</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>information</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\cosine(\text{apricot,information}) =
\]

\[
\cosine(\text{digital,information}) =
\]

\[
\cosine(\text{apricot,digital}) =
\]
Example: Word Similarity

\[
\cos(x, y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2} \sqrt{\sum_i y_i^2}}
\]

<table>
<thead>
<tr>
<th></th>
<th>large</th>
<th>data</th>
<th>computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>apricot</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>digital</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>information</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\cosine(\text{apricot, information}) = \frac{2 + 0 + 0}{\sqrt{4 + 0 + 0} \sqrt{1 + 36 + 1}} = 0.1622
\]

\[
\cosine(\text{digital, information}) =
\]

\[
\cosine(\text{apricot, digital}) =
\]
Example: Word Similarity

\[ \cos(x, y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2} \sqrt{\sum_i y_i^2}} \]

<table>
<thead>
<tr>
<th></th>
<th>large</th>
<th>data</th>
<th>computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>apricot</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>digital</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>information</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\text{cosine(apricot,information)} = \frac{2 + 0 + 0}{\sqrt{4 + 0 + 0} \sqrt{1 + 36 + 1}} = 0.1622
\]

\[
\text{cosine(digital,information)} = \frac{0 + 6 + 2}{\sqrt{0 + 1 + 4} \sqrt{1 + 36 + 1}} = 0.5804
\]

\[
\text{cosine(apricot,digital)} = \frac{0 + 0 + 0}{\sqrt{4 + 0 + 0} \sqrt{0 + 1 + 4}} = 0.0
\]
Other Similarity Measures

\[
\text{sim}_{\text{cosine}}(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}| |\vec{w}|} = \frac{\sum_{i=1}^{N} v_i \times w_i}{\sqrt{\sum_{i=1}^{N} v_i^2} \sqrt{\sum_{i=1}^{N} w_i^2}}
\]

\[
\text{sim}_{\text{Jaccard}}(\vec{v}, \vec{w}) = \frac{\sum_{i=1}^{N} \min(v_i, w_i)}{\sum_{i=1}^{N} \max(v_i, w_i)}
\]

\[
\text{sim}_{\text{Dice}}(\vec{v}, \vec{w}) = \frac{2 \times \sum_{i=1}^{N} \min(v_i, w_i)}{\sum_{i=1}^{N} (v_i + w_i)}
\]

\[
\text{sim}_{\text{JS}}(\vec{v} \mid \vec{w}) = D(\vec{v} \mid \frac{\vec{v} + \vec{w}}{2}) + D(\vec{w} \mid \frac{\vec{v} + \vec{w}}{2})
\]
Adding Morphology, Syntax, and Semantics to Embeddings

Lin (1998): “Automatic Retrieval and Clustering of Similar Words”


Cotterell and Schütze (2015): “Morphological Word Embeddings”

Ferraro et al. (2017): “Frame-Based Continuous Lexical Semantics through Exponential Family Tensor Factorization and Semantic Proto-Roles”

and many more...
Outline

Recap
  Maxent models
  Basic neural language models

Continuous representations
  Motivation
  Key idea: represent words with vectors
  Two common counting types

Two (four) common continuous representation models

Evaluation