Hands-on Distributional Semantics

Part 2: The parameters of a DSM

Stefan Evert\textsuperscript{1} & Gabriella Lapesa\textsuperscript{2}
with Alessandro Lenci\textsuperscript{3} and Marco Baroni\textsuperscript{4}

\textsuperscript{1}Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
\textsuperscript{2}University of Stuttgart, Germany
\textsuperscript{3}University of Pisa, Italy
\textsuperscript{4}University of Trento, Italy

http://wordspace.collocations.de/doku.php/course:esslli2021:start

Copyright © 2009–2021 Evert, Lapesa, Lenci & Baroni | Licensed under CC-by-sa version 3.0
Outline

DSM parameters
- A taxonomy of DSM parameters
- Context type & size
- Feature scaling
- Measuring distance
- Dimensionality reduction

Building a DSM
- Sparse matrices
- Example: a verb-object DSM

Appendix
- Examples
- Three famous examples
General definition of DSMs

A **distributional semantic model** (DSM) is a scaled and/or transformed co-occurrence matrix $M$, such that each row $x$ represents the distribution of a target term across contexts.

<table>
<thead>
<tr>
<th>Term</th>
<th>get</th>
<th>see</th>
<th>use</th>
<th>hear</th>
<th>eat</th>
<th>kill</th>
</tr>
</thead>
<tbody>
<tr>
<td>knife</td>
<td>0.027</td>
<td>-0.024</td>
<td>0.206</td>
<td>-0.022</td>
<td>-0.044</td>
<td>-0.042</td>
</tr>
<tr>
<td>cat</td>
<td>0.031</td>
<td>0.143</td>
<td>-0.243</td>
<td>-0.015</td>
<td>-0.009</td>
<td>0.131</td>
</tr>
<tr>
<td>dog</td>
<td>-0.026</td>
<td>0.021</td>
<td>-0.212</td>
<td>0.064</td>
<td>0.013</td>
<td>0.014</td>
</tr>
<tr>
<td>boat</td>
<td>-0.022</td>
<td>0.009</td>
<td>-0.044</td>
<td>-0.040</td>
<td>-0.074</td>
<td>-0.042</td>
</tr>
<tr>
<td>cup</td>
<td>-0.014</td>
<td>-0.173</td>
<td>-0.249</td>
<td>-0.099</td>
<td>-0.119</td>
<td>-0.042</td>
</tr>
<tr>
<td>pig</td>
<td>-0.069</td>
<td>0.094</td>
<td>-0.158</td>
<td>0.000</td>
<td>0.094</td>
<td>0.265</td>
</tr>
<tr>
<td>banana</td>
<td>0.047</td>
<td>-0.139</td>
<td>-0.104</td>
<td>-0.022</td>
<td>0.267</td>
<td>-0.042</td>
</tr>
</tbody>
</table>

**Term** = word, lemma, phrase, morpheme, word pair, ...
General definition of DSMs

Mathematical notation:

- $k \times n$ co-occurrence matrix $M \in \mathbb{R}^{k \times n}$ (example: $7 \times 6$)
  - $k$ rows = target terms
  - $n$ columns = features or other dimensions

$$M = \begin{bmatrix}
    m_{11} & m_{12} & \cdots & m_{1n} \\
    m_{21} & m_{22} & \cdots & m_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    m_{k1} & m_{k2} & \cdots & m_{kn}
\end{bmatrix}$$

- distribution vector $m_i = i$-th row of $M$, e.g. $m_3 = m_{\text{dog}} \in \mathbb{R}^n$
- components $m_i = (m_{i1}, m_{i2}, \ldots, m_{in}) =$ features of $i$-th term:

$$m_3 = (-0.026, 0.021, -0.212, 0.064, 0.013, 0.014) = (m_{31}, m_{32}, m_{33}, m_{34}, m_{35}, m_{36})$$
Term-term matrix

**Term-term matrix** records co-occurrence frequencies with feature terms for each target term

\[ \mathbf{m}_{\text{dog}} = \text{collocational profile of } \text{dog} \ (\approx \text{word sketch}) \]

\[
\mathbf{M} = \begin{bmatrix} \cdots & \mathbf{m}_1 & \cdots \\
\cdots & \mathbf{m}_2 & \cdots \\
\vdots & \vdots & \vdots \\
\cdots & \mathbf{m}_k & \cdots 
\end{bmatrix}
\]

\[
\begin{array}{cccccccc}
\text{breed} & \text{tail} & \text{feed} & \text{kill} & \text{important} & \text{explain} & \text{likely} \\
83 & 17 & 7 & 37 & - & 1 & - \\
561 & 13 & 30 & 60 & 1 & 2 & 4 \\
42 & 10 & 109 & 134 & 13 & 5 & 5 \\
19 & 9 & 29 & 117 & 81 & 34 & 109 \\
1 & - & 2 & 14 & 68 & 140 & 47 \\
- & 1 & - & 4 & 55 & 34 & 55 \\
- & - & 1 & 6 & 60 & 35 & 17 \\
\end{array}
\]

\[ \text{> TT <- DSM_TermTerm} \]
\[ \text{> head(TT, Inf)} \ # \text{extract full co-oc matrix from DSM object} \]
**Term-context matrix** records frequency of term in each individual context unit (e.g. document, tweet, encyclopaedia article)

$f_{\text{dog}} =$ texts related to or mentioning dogs

\[
F = \begin{bmatrix}
\cdots & f_1 & \cdots \\
\cdots & f_2 & \cdots \\
\cdots & \cdots & \cdots \\
\cdots & f_k & \cdots \\
\end{bmatrix}
\]

<table>
<thead>
<tr>
<th></th>
<th>Felidae</th>
<th>Pet</th>
<th>Feral</th>
<th>Bloat</th>
<th>Philosophy</th>
<th>Kant</th>
<th>Back pain</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>10</td>
<td>10</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>dog</td>
<td>-</td>
<td>10</td>
<td>4</td>
<td>11</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>animal</td>
<td>2</td>
<td>15</td>
<td>10</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>time</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>reason</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>cause</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>effect</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>

> TC <- DSM_TermContext
> head(TC, Inf)
Outline

DSM parameters
A taxonomy of DSM parameters
Context type & size
Feature scaling
Measuring distance
Dimensionality reduction

Building a DSM
Sparse matrices
Example: a verb-object DSM

Appendix
Examples
Three famous examples
Building a distributional model

pre-processed corpus with linguistic annotation
Building a distributional model

pre-processed corpus with linguistic annotation

term-term matrix

define target & feature terms
Building a distributional model

- Pre-processed corpus with linguistic annotation
- Term-term matrix
- Define target & feature terms
- Type & size of co-occurrence
Building a distributional model

1. Pre-processed corpus with linguistic annotation
2. Define target & feature terms
3. Type & size of co-occurrence

M
Building a distributional model

- pre-processed corpus with linguistic annotation
  - define target terms
  - define target & feature terms
  - type & size of co-occurrence
    - term-context matrix
    - term-term matrix

© Evert/Lapesa/Lenci/Baroni (CC-by-sa)
Building a distributional model

pre-processed corpus with linguistic annotation

define target terms
context units or aggregates

term-context matrix

define target & feature terms
type & size of co-occurrence

term-term matrix

M
Building a distributional model

1. Define target terms
2. Define target & feature terms
3. Pre-processed corpus with linguistic annotation
4. Term-context matrix
5. Term-term matrix
6. Context units or aggregates
7. Type & size of co-occurrence
8. M
9. Probabilistic analysis
10. Embedding learned by neural network
Building a distributional model

- Define target terms
- Define target & feature terms
- Pre-processed corpus with linguistic annotation
- Term-context matrix
- Term-term matrix
- Context units or aggregates
- Type & size of co-occurrence
- Geometric analysis
- Feature scaling
- Probabilistic analysis
- Embedding learned by neural network

© Evert/Lapesa/Lenci/Baroni (CC-by-sa)
Building a distributional model

pre-processed corpus with linguistic annotation

define target terms

term-context matrix

define target & feature terms

term-term matrix

type & size of co-occurrence

context units or aggregates

feature scaling

M

generic analysis

probabilistic analysis

embedding learned by neural network

similarity/distance measure + normalization
Building a distributional model

1. Pre-processed corpus with linguistic annotation
   - Define target terms
     - Context units or aggregates
   - Define target & feature terms
     - Type & size of co-occurrence

2. Term-context matrix
3. Term-term matrix

4. Geometric analysis
   - Feature scaling
5. Probabilistic analysis
   - Embedding learned by neural network

6. Similarity/distance measure + normalization
7. Dimensionality reduction

© Evert/Lapesa/Lenci/Baroni (CC-by-sa)
Building a distributional model

pre-processed corpus with linguistic annotation

define target terms
context units or aggregates

define target & feature terms
type & size of co-occurrence

term-context matrix
term-term matrix

M

feature scaling

geometric analysis

similarity/distance measure + normalization

dimensionality reduction

probabilistic analysis
embedding learned by neural network
Definition of target and feature terms

- Choice of linguistic unit (targets $\neq$ features)
  - words
  - bigrams, trigrams, ...
  - multiword units, named entities, phrases, ...
  - morphemes
  - word pairs (analogy tasks)
Definition of target and feature terms

- **Choice of linguistic unit** (targets ≠ features)
  - words
  - bigrams, trigrams, ...
  - multiword units, named entities, phrases, ...
  - morphemes
  - word pairs (☞ analogy tasks)

- **Mapping to target/feature terms** (➔ linguistic annotation)
  - word forms (minimally requires tokenisation)
  - often lemmatisation or stemming to reduce data sparseness:
    \[ \text{go, goes, went, gone, going} \rightarrow \text{go} \]
  - POS disambiguation (\text{light/N vs. light/A vs. light/V})
  - word sense disambiguation (\text{bank}_{\text{river}} \ vs. \text{bank}_{\text{finance}})
  - abstraction: POS tags (or \(n\)-grams of POS tags) as features
Definition of target and feature terms

- Choice of linguistic unit (targets ≠ features)
  - words
  - bigrams, trigrams, ...
  - multiword units, named entities, phrases, ...
  - morphemes
  - word pairs (analogy tasks)

- Mapping to target/feature terms (linguistic annotation)
  - word forms (minimally requires tokenisation)
  - often lemmatisation or stemming to reduce data sparseness: 
    - go, goes, went, gone, going → go
  - POS disambiguation (light/N vs. light/A vs. light/V)
  - word sense disambiguation (bank\_river vs. bank\_finance)
  - abstraction: POS tags (or n-grams of POS tags) as features

❓ What is the effect of these choices?
## Effects of term mapping

### Nearest neighbours of *walk* (BNC)

<table>
<thead>
<tr>
<th>word forms</th>
<th>lemmatised + POS</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ stroll</td>
<td>▶ hurry</td>
</tr>
<tr>
<td>▶ walking</td>
<td>▶ stroll</td>
</tr>
<tr>
<td>▶ walked</td>
<td>▶ stride</td>
</tr>
<tr>
<td>▶ go</td>
<td>▶ trudge</td>
</tr>
<tr>
<td>▶ path</td>
<td>▶ amble</td>
</tr>
<tr>
<td>▶ drive</td>
<td>▶ wander</td>
</tr>
<tr>
<td>▶ ride</td>
<td>▶ walk (noun)</td>
</tr>
<tr>
<td>▶ wander</td>
<td>▶ walking</td>
</tr>
<tr>
<td>▶ sprinted</td>
<td>▶ retrace</td>
</tr>
<tr>
<td>▶ sauntered</td>
<td>▶ scuttle</td>
</tr>
</tbody>
</table>

[http://clic.cimec.unitn.it/infomap-query/](http://clic.cimec.unitn.it/infomap-query/)
Effects of term mapping

Nearest neighbours of *arrivare* (Repubblica)

<table>
<thead>
<tr>
<th>word forms</th>
<th>lemmatised + POS</th>
</tr>
</thead>
<tbody>
<tr>
<td>◦ giungere</td>
<td>◦ giungere</td>
</tr>
<tr>
<td>◦ raggiungere</td>
<td>◦ aspettare</td>
</tr>
<tr>
<td>◦ arrivi</td>
<td>◦ attendere</td>
</tr>
<tr>
<td>◦ raggiungimento</td>
<td>◦ arrivo (noun)</td>
</tr>
<tr>
<td>◦ raggiunto</td>
<td>◦ ricevere</td>
</tr>
<tr>
<td>◦ trovare</td>
<td>◦ accontentare</td>
</tr>
<tr>
<td>◦ raggiunge</td>
<td>◦ approdare</td>
</tr>
<tr>
<td>◦ arrivasse</td>
<td>◦ pervenire</td>
</tr>
<tr>
<td>◦ arriverà</td>
<td>◦ venire</td>
</tr>
<tr>
<td>◦ concludere</td>
<td>◦ piombare</td>
</tr>
</tbody>
</table>

http://clic.cimec.unitn.it/infomap-query/
Selection of target and feature terms

- Full-vocabulary models are often unmanageable
  - 762,424 distinct word forms in BNC, 605,910 lemmata
  - large Web corpora have > 10 million distinct word forms
  - low-frequency targets (and features) are not reliable ("noisy")
Selection of target and feature terms

- Full-vocabulary models are often unmanageable
  - 762,424 distinct word forms in BNC, 605,910 lemmata
  - large Web corpora have > 10 million distinct word forms
  - low-frequency targets (and features) are not reliable (“noisy”)

- Frequency-based selection
  - corpus frequency \( f \geq F_{\text{min}} \) or \( n_w \) most frequent terms
  - sometimes upper threshold for features: \( F_{\text{min}} \leq f \leq F_{\text{max}} \)
Selection of target and feature terms

- Full-vocabulary models are often unmanageable
  - 762,424 distinct word forms in BNC, 605,910 lemmata
  - large Web corpora have > 10 million distinct word forms
  - low-frequency targets (and features) are not reliable ("noisy")

- Frequency-based selection
  - corpus frequency $f \geq F_{\text{min}}$ or $n_w$ most frequent terms
  - sometimes upper threshold for features: $F_{\text{min}} \leq f \leq F_{\text{max}}$

- Relevance-based selection of features
  - criterion from information retrieval: document frequency $df$
    (high $df \Rightarrow$ uninformative / low $df \Rightarrow$ too sparse to be useful)
  - alternatives: entropy $H$ or chi-squared statistic $X^2$
Selection of target and feature terms

- Full-vocabulary models are often unmanageable
  - 762,424 distinct word forms in BNC, 605,910 lemmata
  - large Web corpora have > 10 million distinct word forms
  - low-frequency targets (and features) are not reliable (“noisy”)

- Frequency-based selection
  - corpus frequency $f \geq F_{\text{min}}$ or $n_w$ most frequent terms
  - sometimes upper threshold for features: $F_{\text{min}} \leq f \leq F_{\text{max}}$

- Relevance-based selection of features
  - criterion from information retrieval: document frequency $df$
    (high $df$ ➜ uninformative / low $df$ ➜ too sparse to be useful)
  - alternatives: entropy $H$ or chi-squared statistic $X^2$

- Other criteria
  - POS-based filter: no function words, only verbs, nouns, …
  - general dictionary, words required for particular task, …
Building a distributional model

1. Define target terms
2. Define target & feature terms
3. Context units or aggregates
4. Type & size of co-occurrence
5. Term-context matrix
6. Term-term matrix
7. Pre-processed corpus with linguistic annotation
8. Feature scaling
9. Similarity/distance measure + normalization
10. Dimensionality reduction
11. Geometric analysis
12. Probabilistic analysis
13. Embedding learned by neural network
Term-context matrix: choice of context unit

- Features are usually tokens of the selected context unit, i.e. individual instances of a
  - document, novel, Wikipedia article, Web page, ...
  - paragraph, sentence, tweet, ...
  - “co-occurrence” $f_{ij} =$ frequency of term $i$ in context token $j$
Term-context matrix: choice of context unit

- Features are usually **tokens** of the selected context unit, i.e. individual instances of a
  - document, novel, Wikipedia article, Web page, ...
  - paragraph, sentence, tweet, ...
  - “co-occurrence” $f_{ij}$ = frequency of term $i$ in context token $j$

- Similar context tokens can be **aggregated**, e.g.
  - feature = cluster of near-duplicate documents
  - feature = syntactic structure of sentence (ignoring content)
  - feature = all tweets from same author (“supertweet”)
  - $f_{ij}$ = pooled frequency count for aggregate $j$
Term-context matrix: choice of context unit

- Features are usually **tokens** of the selected context unit, i.e. individual instances of a
  - document, novel, Wikipedia article, Web page, ...
  - paragraph, sentence, tweet, ...
  - “co-occurrence” $f_{ij} = $ frequency of term $i$ in context token $j$

- Similar context tokens can be **aggregated**, e.g.
  - feature = cluster of near-duplicate documents
  - feature = syntactic structure of sentence (ignoring content)
  - feature = all tweets from same author (“supertweet”)
  - $f_{ij} = $ pooled frequency count for aggregate $j$

- Generalization: context **types**
  - e.g. pattern of POS tags around target word
  - e.g. subcategorisation pattern of target verb
Building a distributional model

pre-processed corpus with linguistic annotation

define target terms

term-context matrix

define target & feature terms

term-term matrix

classify units or aggregates

define target terms

type & size of co-occurrence

M

globular analysis

feature scaling

probabilistic analysis

similarities/distance measure + normalization

embedding learned by neural network

dimensionality reduction
Term-term matrix: definition of co-occurrence context

- Different types of co-occurrence (Evert 2008)
  - **surface context** (word or character window)
  - **textual context** (non-overlapping segments)
  - **syntactic context** (dependency relations)

☞ from research into collocations
Term-term matrix: definition of co-occurrence context

- Different types of co-occurrence (Evert 2008)
  - surface context (word or character window)
  - textual context (non-overlapping segments)
  - syntactic context (dependency relations)
  - from research into collocations

- Context size
  - small context (few words, syntactic relation) ➔ more specific
  - large context (many words, entire document) ➔ more general
Term-term matrix: definition of co-occurrence context

▶ Different types of co-occurrence (Evert 2008)
  ▶ **surface context** (word or character window)
  ▶ **textual context** (non-overlapping segments)
  ▶ **syntactic context** (dependency relations)
☞ from research into collocations

▶ Context size
  ▶ small context (few words, syntactic relation) → more specific
  ▶ large context (many words, entire document) → more general

▶ Different roles of co-occurrence context
  ▶ unstructured context → acts as a **filter** for counts
  ▶ **structured** context → subcategorizes feature terms
Term-term matrix: definition of co-occurrence context

- Different types of co-occurrence (Evert 2008)
  - surface context (word or character window)
  - textual context (non-overlapping segments)
  - syntactic context (dependency relations)

- Context size
  - small context (few words, syntactic relation) ➔ more specific
  - large context (many words, entire document) ➔ more general

- Different roles of co-occurrence context
  - unstructured context ➔ acts as a filter for counts
  - structured context ➔ subcategorizes feature terms

What effects do you expect from these choices?
Surface context

Context term occurs within a span of $k$ words around target.

The silhouette of the sun beyond a wide-open bay on the lake; the sun still glitters although evening has arrived in Kuhmo. It’s midsummer; the living room has its instruments and other objects in each of its corners. \[\text{[L3/R3 span, } k = 6\]\n
Parameters:

- span size (in words or characters)
- symmetric vs. one-sided span
- uniform or “triangular” (distance-based) weighting (don’t!)
- spans clamped to sentences or other textual units?
Effect of span size

Nearest neighbours of *dog* (BNC)

<table>
<thead>
<tr>
<th>2-word span</th>
<th>30-word span</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ cat</td>
<td>▶ kennel</td>
</tr>
<tr>
<td>▶ horse</td>
<td>▶ puppy</td>
</tr>
<tr>
<td>▶ fox</td>
<td>▶ pet</td>
</tr>
<tr>
<td>▶ pet</td>
<td>▶ bitch</td>
</tr>
<tr>
<td>▶ rabbit</td>
<td>▶ terrier</td>
</tr>
<tr>
<td>▶ pig</td>
<td>▶ rottweiler</td>
</tr>
<tr>
<td>▶ animal</td>
<td>▶ canine</td>
</tr>
<tr>
<td>▶ mongrel</td>
<td>▶ cat</td>
</tr>
<tr>
<td>▶ sheep</td>
<td>▶ to bark</td>
</tr>
<tr>
<td>▶ pigeon</td>
<td>▶ Alsatian</td>
</tr>
</tbody>
</table>

http://clic.cimec.unitn.it/infomap-query/
Textual context

Context term is in the same linguistic unit as target.

The silhouette of the sun beyond a wide-open bay on the lake; the sun still glitters although evening has arrived in Kuhmo. It’s midsummer; the living room has its instruments and other objects in each of its corners.

Parameters:

- choice of linguistic unit
  - sentence
  - paragraph
  - turn in a conversation
  - Web page
  - tweet

☞ similar to large surface spans, but more self-contained
Syntactic context

Context term is linked to target by a **syntactic dependency** (e.g. subject, modifier, ...).

The *silhouette* of the *sun* beyond a wide-open *bay* on the lake; the *sun* still *glitters* although evening has arrived in Kuhmo. It’s midsummer; the living room has its instruments and other objects in each of its corners.

Parameters:

- types of syntactic dependency (Padó & Lapata 2007)
- maximal length of dependency path (1 for direct relation)
- homogeneous data (e.g. only verb-object) vs. heterogeneous data (e.g. all children and parents of the verb)
“Knowledge pattern” context

Context term is linked to target by a **lexico-syntactic pattern** (text mining, cf. Hearst 1992, Pantel & Pennacchiotti 2008, etc.).

In Provence, Van Gogh painted with bright **colors** such as **red** and **yellow**. These **colors** produce incredible **effects** on anybody looking at his paintings.

Parameters:

- inventory of lexical patterns
  - lots of research to identify semantically interesting patterns (cf. Almuhareb & Poesio 2004, Veale & Hao 2008, etc.)
- fixed vs. flexible patterns
  - patterns are mined from large corpora and automatically generalised (optional elements, POS tags or semantic classes)
Comparison of co-occurrence contexts

Contexts range from general/implicit to specific/explicit:

<table>
<thead>
<tr>
<th>features are</th>
</tr>
</thead>
<tbody>
<tr>
<td>textual / large span</td>
</tr>
<tr>
<td>from same topic domain</td>
</tr>
</tbody>
</table>
Comparison of co-occurrence contexts

Contexts range from general/implicit to specific/explicit:

<table>
<thead>
<tr>
<th>Textual / large span</th>
<th>from same topic domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small span</td>
<td>Collocations</td>
</tr>
</tbody>
</table>

features are
## Comparison of co-occurrence contexts

Contexts range from general/implicit to specific/explicit:

<table>
<thead>
<tr>
<th>Features are</th>
<th>Textual / Large span</th>
<th>Small Span</th>
<th>Syntactic (Single Relation)</th>
<th>Collocations</th>
<th>Attributes (Focus on Aspect)</th>
</tr>
</thead>
<tbody>
<tr>
<td>From same topic domain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Features are</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Comparison of co-occurrence contexts

Contexts range from general/implicit to specific/explicit:

<table>
<thead>
<tr>
<th>Textual / large span</th>
<th>from same topic domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small span</td>
<td>collocations</td>
</tr>
<tr>
<td>Syntactic</td>
<td>attributes</td>
</tr>
<tr>
<td>(single relation)</td>
<td>(focus on aspect)</td>
</tr>
<tr>
<td>Knowledge pattern</td>
<td>properties</td>
</tr>
</tbody>
</table>
Structured vs. unstructured context

- In **unstructured** models, context specification acts as a **filter**
  - determines whether context token counts as co-occurrence
  - e.g. must be linked by any direct syntactic dependency relation
Structured vs. unstructured context

▶ In **unstructured** models, context specification acts as a **filter**
  ▶ determines whether context token counts as co-occurrence
  ▶ e.g. must be linked by any direct syntactic dependency relation

▶ In **structured** models, feature terms are **subtyped**
  ▶ depending on their position in the context
  ▶ e.g. left **vs.** right context, type of syntactic relation, etc.
Structured vs. unstructured surface context

A dog bites a man. The man’s dog bites a dog. A dog bites a man.

**unstructured**

<table>
<thead>
<tr>
<th></th>
<th>bite</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
</tr>
<tr>
<td>man</td>
<td>3</td>
</tr>
</tbody>
</table>

Structured: more sensitive to semantic distinctions

Data are less sparse (L/R context aggregated)
Structured vs. unstructured surface context

A dog bites a man. The man’s dog bites a dog. A dog bites a man.

**unstructured**

<table>
<thead>
<tr>
<th></th>
<th>bite</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
</tr>
<tr>
<td>man</td>
<td>3</td>
</tr>
</tbody>
</table>

A dog bites a man. The man’s dog bites a dog. A dog bites a man.

**structured**

<table>
<thead>
<tr>
<th></th>
<th>bite-L</th>
<th>bite-R</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>man</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Structured vs. unstructured surface context

A dog bites a man. The man’s dog bites a dog. A dog bites a man.

**unstructured**

<table>
<thead>
<tr>
<th></th>
<th>bite</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
</tr>
<tr>
<td>man</td>
<td>3</td>
</tr>
</tbody>
</table>

- data are less sparse (L/R context aggregated)

A dog bites a man. The man’s dog bites a dog. A dog bites a man.

**structured**

<table>
<thead>
<tr>
<th></th>
<th>bite-L</th>
<th>bite-R</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>man</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

- more sensitive to semantic distinctions
Structured vs. unstructured dependency context

A dog bites a man. The man’s dog bites a dog. A dog bites a man.

**unstructured**

<table>
<thead>
<tr>
<th></th>
<th>bite</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
</tr>
<tr>
<td>man</td>
<td>2</td>
</tr>
</tbody>
</table>

- data are less sparse (all syntactic relations aggregated)

A dog bites a man. The man’s dog bites a dog. A dog bites a man.

**structured**

<table>
<thead>
<tr>
<th></th>
<th>bite-subj</th>
<th>bite-obj</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>man</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

- more sensitive to semantic distinctions
Building a distributional model

pre-processed corpus with linguistic annotation

define target terms
context units or aggregates

define target & feature terms
type & size of co-occurrence

term-context matrix

term-term matrix

feature scaling
goodmetric analysis

similarity/distance measure + normalization
dimensionality reduction

M
Marginal and expected frequencies

- Matrix of observed co-occurrence frequencies not sufficient

<table>
<thead>
<tr>
<th>target</th>
<th>feature</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>small</td>
<td>855</td>
</tr>
<tr>
<td>dog</td>
<td>domesticated</td>
<td>29</td>
</tr>
</tbody>
</table>

- Notation
  - $O = \text{observed co-occurrence frequency}$
Marginal and expected frequencies

- Matrix of observed co-occurrence frequencies not sufficient

<table>
<thead>
<tr>
<th>target</th>
<th>feature</th>
<th>( O )</th>
<th>( R )</th>
<th>( C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>small</td>
<td>855</td>
<td>33,338</td>
<td>490,580</td>
</tr>
<tr>
<td>dog</td>
<td>domesticated</td>
<td>29</td>
<td>33,338</td>
<td>918</td>
</tr>
</tbody>
</table>

- Notation
  - \( O \) = observed co-occurrence frequency
  - \( R \) = overall frequency of target term = row marginal frequency
  - \( C \) = overall frequency of feature = column marginal frequency
  - \( N \) = sample size \( \approx \) size of corpus
Marginal and expected frequencies

- **Matrix of observed** co-occurrence frequencies not sufficient

<table>
<thead>
<tr>
<th>target</th>
<th>feature</th>
<th>$O$</th>
<th>$R$</th>
<th>$C$</th>
<th>$E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>small</td>
<td>855</td>
<td>33,338</td>
<td>490,580</td>
<td>134.34</td>
</tr>
<tr>
<td>dog</td>
<td>domesticated</td>
<td>29</td>
<td>33,338</td>
<td>918</td>
<td>0.25</td>
</tr>
</tbody>
</table>

- **Notation**
  - $O = \text{observed co-occurrence frequency}$
  - $R = \text{overall frequency of target term} = \text{row marginal frequency}$
  - $C = \text{overall frequency of feature} = \text{column marginal frequency}$
  - $N = \text{sample size} \approx \text{size of corpus}$

- **Expected** co-occurrence **frequency** (cf. Evert 2008)

$$E = \frac{R \cdot C}{N} \leftrightarrow O$$
Obtaining marginal frequencies (Evert 2008)

- Term-document matrix
  - \( R = \) frequency of target term in corpus
  - \( C = \) size of document (\# tokens)
  - \( N = \) corpus size
Obtaining marginal frequencies (Evert 2008)

- Term-document matrix
  - $R =$ frequency of target term in corpus
  - $C =$ size of document (number of tokens)
  - $N =$ corpus size

- Syntactic co-occurrence
  - Number of dependency instances in which target/feature participates
  - $N =$ total number of dependency instances
  - $N, R, C$ can be computed from full co-occurrence matrix $M$
Obtaining marginal frequencies (Evert 2008)

- Term-document matrix
  - $R = \text{frequency of target term in corpus}$
  - $C = \text{size of document (\# tokens)}$
  - $N = \text{corpus size}$

- Syntactic co-occurrence
  - $\# \text{ of dependency instances in which target/feature participates}$
  - $N = \text{total number of dependency instances}$
  - $N, R, C$ can be computed from full co-occurrence matrix $M$

- Textual co-occurrence
  - $R, C, O$ are “document” frequencies, i.e. number of context units in which target, feature or combination occurs
  - $N = \text{total \# of context units}$
Obtaining marginal frequencies (Evert 2008)

- Surface co-occurrence
  - it is quite tricky to obtain fully consistent counts
  - at least correct $E$ for span size $k$ ($\equiv$ # tokens in span)$^1$

$$E = k \cdot \frac{R \cdot C}{N}$$

with $R, C = $ individual corpus frequencies and $N = $ corpus size

- can also be implemented by pre-multiplying $R' = k \cdot R$
- approach used for all pre-compiled surface DSMs in the course

alternatively, compute marginals and sample size by summing over full co-occurrence matrix ($\Rightarrow E$ as above, but inflated $N$)

$^1$NB: shifted PPMI (Levy & Goldberg 2014) corresponds to a post-hoc application of the span size adjustment. It performs worse than PPMI, but paper suggests they already approximate correct $E$ by summing over matrix $M$. 
Marginal frequencies in *wordspace*

DSM objects in *wordspace* (class `dsm`) include marginal frequencies as well as counts of nonzero cells for rows and columns.

```r
> TT$rows
term  f  nnzero
1  cat 22007  5
2  dog  50807  7
3 animal 77053  7
4  time 1156693  7
5 reason  95047  6
6  cause  54739  5
7  effect 133102  6
> TT$cols
...
> TT$globals$N
[1] 199902178
> TT$M  # the full co-occurrence matrix
```
Building a distributional model

- Define target terms
- Define target & feature terms
- Context units or aggregates
- Type & size of co-occurrence
- Pre-processed corpus with linguistic annotation
- Term-context matrix
- Term-term matrix
- Feature scaling
- Similarity/distance measure + normalization
- Dimensionality reduction
- Geometric analysis
- Probabilistic analysis
- Embedding learned by neural network

© Evert/Lapesa/Lenci/Baroni (CC-by-sa)
Feature scaling

- **M** is often dominated by few very large entries (highly skewed frequency distribution due to **Zipf’s law**)

- Logarithmic scaling: 
  \[ O' = \log(O + 1) \]
  (cf. Weber-Fechner law for human perception)

- Statistical association measures (Evert 2004, 2008) take frequency of target term and feature into account
  - usually based on comparison of observed and expected co-occurrence frequency
  - measures differ in how they balance **O** and **E**
Feature scaling

- M is often dominated by few very large entries (→ highly skewed frequency distribution due to **Zipf’s law**)

- Logarithmic scaling: $O' = \log(O + 1)$
  (cf. Weber-Fechner law for human perception)
Feature scaling

- **M** is often dominated by few very large entries (highly skewed frequency distribution due to **Zipf’s law**)

- Logarithmic scaling: $O' = \log(O + 1)$
  (cf. Weber-Fechner law for human perception)

- Statistical **association measures** (Evert 2004, 2008) take frequency of target term and feature into account
  - usually based on comparison of observed and expected co-occurrence frequency
  - measures differ in how they balance $O$ and $E$
Simple association measures

<table>
<thead>
<tr>
<th>target</th>
<th>feature</th>
<th>$O$</th>
<th>$E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>small</td>
<td>855</td>
<td>134.34</td>
</tr>
<tr>
<td>dog</td>
<td>domesticated</td>
<td>29</td>
<td>0.25</td>
</tr>
<tr>
<td>dog</td>
<td>sgjkj</td>
<td>1</td>
<td>0.00027</td>
</tr>
</tbody>
</table>
Simple association measures

- pointwise Mutual Information (MI)

\[
\text{MI} = \log_2 \frac{O}{E}
\]

<table>
<thead>
<tr>
<th>target</th>
<th>feature</th>
<th>$O$</th>
<th>$E$</th>
<th>MI</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>small</td>
<td>855</td>
<td>134.34</td>
<td>2.67</td>
</tr>
<tr>
<td>dog</td>
<td>domesticated</td>
<td>29</td>
<td>0.25</td>
<td>6.85</td>
</tr>
<tr>
<td>dog</td>
<td>sgjkk</td>
<td>1</td>
<td>0.00027</td>
<td>11.85</td>
</tr>
</tbody>
</table>
Simple association measures

- **pointwise Mutual Information (MI)**
  \[ \text{MI} = \log_2 \frac{O}{E} \]

- **local MI**
  \[ \text{local-MI} = O \cdot \text{MI} = O \cdot \log_2 \frac{O}{E} \]

<table>
<thead>
<tr>
<th>target</th>
<th>feature</th>
<th>( O )</th>
<th>( E )</th>
<th>MI</th>
<th>local-MI</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>small</td>
<td>855</td>
<td>134.34</td>
<td>2.67</td>
<td>2282.88</td>
</tr>
<tr>
<td>dog</td>
<td>domesticated</td>
<td>29</td>
<td>0.25</td>
<td>6.85</td>
<td>198.76</td>
</tr>
<tr>
<td>dog</td>
<td>sgjkj</td>
<td>1</td>
<td>0.00027</td>
<td>11.85</td>
<td>11.85</td>
</tr>
</tbody>
</table>
Simple association measures

- **Pointwise Mutual Information** (MI)
  \[ \text{MI} = \log_2 \frac{O}{E} \]

- **Local MI**
  \[ \text{local-MI} = O \cdot \text{MI} = O \cdot \log_2 \frac{O}{E} \]

- **t-score**
  \[ t = \frac{O - E}{\sqrt{O}} \]

<table>
<thead>
<tr>
<th>target</th>
<th>feature</th>
<th>$O$</th>
<th>$E$</th>
<th>MI</th>
<th>local-MI</th>
<th>t-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>small</td>
<td>855</td>
<td>134.34</td>
<td>2.67</td>
<td>2282.88</td>
<td>24.64</td>
</tr>
<tr>
<td>dog</td>
<td>domesticated</td>
<td>29</td>
<td>0.25</td>
<td>6.85</td>
<td>198.76</td>
<td>5.34</td>
</tr>
<tr>
<td>dog</td>
<td>sgjkJ</td>
<td>1</td>
<td>0.00027</td>
<td>11.85</td>
<td>11.85</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Other association measures

- simple log-likelihood (≈ local-MI)

\[ G^2 = \pm 2 \cdot \left( O \cdot \log_2 \frac{O}{E} - (O - E) \right) \]

with positive sign for \( O > E \) and negative sign for \( O < E \)
Other association measures

- simple log-likelihood ($\approx \text{local-MI}$)

$$G^2 = \pm 2 \cdot \left( O \cdot \log_2 \frac{O}{E} - (O - E) \right)$$

with positive sign for $O > E$ and negative sign for $O < E$

- Dice coefficient

$$\text{Dice} = \frac{2O}{R + C}$$
Other association measures

- **simple log-likelihood** (≈ local-MI)

\[ G^2 = \pm 2 \cdot \left( O \cdot \log_2 \frac{O}{E} - (O - E) \right) \]

with positive sign for \( O > E \) and negative sign for \( O < E \)

- **Dice coefficient**

\[ \text{Dice} = \frac{2O}{R + C} \]

- Many other association measures (AMs) available, often based on full contingency tables (see Evert 2008)
  - [http://www.collocations.de/](http://www.collocations.de/)
  - [http://sigil.r-forge.r-project.org/](http://sigil.r-forge.r-project.org/)
## Applying association scores in wordspace

```r
> options(digits=3)  # print fractional values with limited precision
> dsm.score(TT, score="MI", sparse=FALSE, matrix=TRUE)
```

<table>
<thead>
<tr>
<th></th>
<th>breed</th>
<th>tail</th>
<th>feed</th>
<th>kill</th>
<th>important</th>
<th>explain</th>
<th>likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>6.21</td>
<td>4.568</td>
<td>3.129</td>
<td>2.801</td>
<td>-Inf</td>
<td>0.0182</td>
<td>-Inf</td>
</tr>
<tr>
<td>dog</td>
<td>7.78</td>
<td>3.081</td>
<td>3.922</td>
<td>2.323</td>
<td>-3.774</td>
<td>-1.1888</td>
<td>-0.4958</td>
</tr>
<tr>
<td>animal</td>
<td>3.50</td>
<td>2.132</td>
<td>4.747</td>
<td>2.832</td>
<td>-0.674</td>
<td>-0.4677</td>
<td>-0.0966</td>
</tr>
<tr>
<td>time</td>
<td>-1.65</td>
<td>-2.236</td>
<td>-0.729</td>
<td>-1.097</td>
<td>-1.728</td>
<td>-1.2382</td>
<td>0.6392</td>
</tr>
<tr>
<td>reason</td>
<td>-2.30</td>
<td>-Inf</td>
<td>-1.982</td>
<td>-0.388</td>
<td>1.472</td>
<td>4.0368</td>
<td>2.8860</td>
</tr>
<tr>
<td>cause</td>
<td>-Inf</td>
<td>-0.834</td>
<td>-Inf</td>
<td>-2.177</td>
<td>1.900</td>
<td>2.8329</td>
<td>4.0691</td>
</tr>
<tr>
<td>effect</td>
<td>-Inf</td>
<td>-2.116</td>
<td>-2.468</td>
<td>-2.459</td>
<td>0.791</td>
<td>1.6312</td>
<td>0.9221</td>
</tr>
</tbody>
</table>

- Cells with score $x$ = $-\infty$ are inconvenient.
- Distribution of scores may be even more skewed than co-occurrence frequencies themselves (esp. for $G^2$).
Applying association scores in wordspace

```r
> options(digits=3)  # print fractional values with limited precision
> dsm.score(TT, score="MI", sparse=FALSE, matrix=TRUE)

<table>
<thead>
<tr>
<th></th>
<th>breed</th>
<th>tail</th>
<th>feed</th>
<th>kill</th>
<th>important</th>
<th>explain</th>
<th>likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>6.21</td>
<td>4.568</td>
<td>3.129</td>
<td>2.801</td>
<td>-Inf</td>
<td>0.0182</td>
<td>-Inf</td>
</tr>
<tr>
<td>dog</td>
<td>7.78</td>
<td>3.081</td>
<td>3.922</td>
<td>2.323</td>
<td>-3.774</td>
<td>-1.1888</td>
<td>-0.4958</td>
</tr>
<tr>
<td>animal</td>
<td>3.50</td>
<td>2.132</td>
<td>4.747</td>
<td>2.832</td>
<td>-0.674</td>
<td>-0.4677</td>
<td>-0.0966</td>
</tr>
<tr>
<td>time</td>
<td>-1.65</td>
<td>-2.236</td>
<td>-0.729</td>
<td>-1.097</td>
<td>-1.728</td>
<td>-1.2382</td>
<td>0.6392</td>
</tr>
<tr>
<td>reason</td>
<td>-2.30</td>
<td>-Inf</td>
<td>-1.982</td>
<td>-0.388</td>
<td>1.472</td>
<td>4.0368</td>
<td>2.8860</td>
</tr>
<tr>
<td>cause</td>
<td>-Inf</td>
<td>-0.834</td>
<td>-Inf</td>
<td>-2.177</td>
<td>1.900</td>
<td>2.8329</td>
<td>4.0691</td>
</tr>
<tr>
<td>effect</td>
<td>-Inf</td>
<td>-2.116</td>
<td>-2.468</td>
<td>-2.459</td>
<td>0.791</td>
<td>1.6312</td>
<td>0.9221</td>
</tr>
</tbody>
</table>
```

☞ sparseness of matrix representation is lost (try with TC!)
☞ cells with score $x = -\infty$ are inconvenient
☞ distribution of scores may be even more skewed than co-occurrence frequencies themselves (esp. for $G^2$)
Sparse association measures

- Sparse association scores are cut off at zero, i.e.

\[ f(x) = \begin{cases} 
  x & x > 0 \\
  0 & x \leq 0 
\end{cases} \]

- Also known as “positive” scores
  - PPMI = positive pointwise MI (e.g. Bullinaria & Levy 2007)
  - wordspace computes sparse AMs by default ➔ "MI" = PPMI
Sparse association measures

- Sparse association scores are cut off at zero, i.e.

\[
f(x) = \begin{cases} 
  x & x > 0 \\
  0 & x \leq 0
\end{cases}
\]

- Also known as “positive” scores
  - \textsc{ppmi} = positive pointwise MI (e.g. Bullinaria & Levy 2007)
  - \textsc{wordspace} computes sparse AMs by default \( \Rightarrow \) "\textsc{mi}" = PPMI
- Preserves sparseness if \( x \leq 0 \) for all empty cells \( (O = 0) \)
  - Sparseness may even increase: cells with \( x < 0 \) become empty
Sparse association measures

- Sparse association scores are cut off at zero, i.e.

\[ f(x) = \begin{cases} 
  x & x > 0 \\
  0 & x \leq 0 
\end{cases} \]

- Also known as “positive” scores
  - PPMI = positive pointwise MI (e.g. Bullinaria & Levy 2007)
  - wordspace computes sparse AMs by default \( \Rightarrow "MI" = \text{PPMI} \)
- Preserves sparseness if \( x \leq 0 \) for all empty cells (\( O = 0 \))
  - sparseness may even increase: cells with \( x < 0 \) become empty
- Further thinning may be beneficial (Polajnar & Clark 2014)
  - apply shifted cutoff threshold \( x > \theta \) (Levy et al. 2015)
  - keep only \( k \) top-scoring features for each target
Score transformations

An additional scale transformation can be applied in order to de-skew association scores:
Score transformations

An additional scale transformation can be applied in order to de-skew association scores:

- signed logarithmic transformation
  \[ f(x) = \pm \log(|x| + 1) \]
Score transformations

An additional scale transformation can be applied in order to de-skew association scores:

- **signed logarithmic transformation**
  \[ f(x) = \pm \log(|x| + 1) \]

- **sigmoid transformation as soft binarization**
  \[ f(x) = \tanh x \]
Score transformations

An additional scale transformation can be applied in order to de-skew association scores:

- signed logarithmic transformation
  \[ f(x) = \pm \log(|x| + 1) \]

- sigmoid transformation as soft binarization
  \[ f(x) = \tanh x \]

- sparse AM as cutoff transformation (aka. ReLU)
Score transformations

An additional scale transformation can be applied in order to de-skew association scores:

- signed logarithmic transformation

\[ f(x) = \pm \log(|x| + 1) \]

- sigmoid transformation as soft binarization

\[ f(x) = \tanh x \]

- sparse AM as (shifted) cutoff transformation (aka. ReLU)
Association scores & transformations in wordspace

<table>
<thead>
<tr>
<th>Breed</th>
<th>Tail</th>
<th>Feed</th>
<th>Kill</th>
<th>Important</th>
<th>Explain</th>
<th>Likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>6.21</td>
<td>4.57</td>
<td>3.13</td>
<td>2.80</td>
<td>0.000</td>
<td>0.0182</td>
</tr>
<tr>
<td>Dog</td>
<td>7.78</td>
<td>3.08</td>
<td>3.92</td>
<td>2.32</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Animal</td>
<td>3.50</td>
<td>2.13</td>
<td>4.75</td>
<td>2.83</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Reason</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.472</td>
<td>4.0368</td>
</tr>
<tr>
<td>Cause</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.900</td>
<td>2.8329</td>
</tr>
<tr>
<td>Effect</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.791</td>
<td>1.6312</td>
</tr>
</tbody>
</table>

> dsm.score(TT, score="MI", matrix=TRUE) # PPMI

> dsm.score(TT, score="simple-ll", matrix=TRUE)

> dsm.score(TT, score="simple-ll", transf="log", matrix=T)

# logarithmic co-occurrence frequency

> dsm.score(TT, score="freq", transform="log", matrix=T)

# now try other parameter combinations

> ?dsm.score # read help page for available parameter settings
Building a distributional model

pre-processed corpus with linguistic annotation

define target terms
context units or aggregates

define target & feature terms
type & size of co-occurrence

M

feature scaling

general analysis

similarity/distance measure + normalization

dimensionality reduction

term-context matrix

term-term matrix

probabilistic analysis
embedding learned by neural network
Geometric distance = metric

- **Distance** between vectors $u, v \in \mathbb{R}^n \rightarrow$ (dis)similarity
  - $u = (u_1, \ldots, u_n)$
  - $v = (v_1, \ldots, v_n)$

![Diagram showing distance between vectors $u$ and $v$ with Euclidean and Manhattan distances marked.]

- **Euclidean distance** $d_2(u, v) = 3.6$
- **"City block" Manhattan distance** $d_1(u, v) = 5$

© Evert/Lapesa/Lenci/Baroni (CC-by-sa) DSM Tutorial – Part 2
Geometric distance = metric

- **Distance** between vectors $u, v \in \mathbb{R}^n \rightarrow$ (dis)similarity
  - $u = (u_1, \ldots, u_n)$
  - $v = (v_1, \ldots, v_n)$
- **Euclidean** distance $d_2(u, v)$

$$d_2(u, v) := \sqrt{(u_1 - v_1)^2 + \cdots + (u_n - v_n)^2}$$
Geometric distance = metric

- **Distance** between vectors $u, v \in \mathbb{R}^n$ \(\rightarrow\) (dis)similarity
  - $u = (u_1, \ldots, u_n)$
  - $v = (v_1, \ldots, v_n)$
- **Euclidean** distance $d_2(u, v)$
- “City block” **Manhattan** distance $d_1(u, v)$

\[
d_1(u, v) := |u_1 - v_1| + \cdots + |u_n - v_n|
\]
Geometric distance = metric

- **Distance** between vectors $u, v \in \mathbb{R}^n \rightarrow (dis)\text{similarity}
  - $u = (u_1, \ldots, u_n)$
  - $v = (v_1, \ldots, v_n)$

- **Euclidean** distance $d_2(u, v)$
- “City block” **Manhattan** distance $d_1(u, v)$
- Both are special cases of the **Minkowski** $p$-distance $d_p(u, v)$ (for $p \in [1, \infty]$)

\[
d_p(u, v) := \left( |u_1 - v_1|^p + \cdots + |u_n - v_n|^p \right)^{1/p}
\]
Geometric distance = metric

» **Distance** between vectors $u, v \in \mathbb{R}^n \Rightarrow$ (dis)similarity
  
   - $u = (u_1, \ldots, u_n)$
   - $v = (v_1, \ldots, v_n)$

» **Euclidean** distance $d_2(u, v)$

» “City block” **Manhattan**

distance $d_1(u, v)$

» Both are special cases of the **Minkowski** $p$-distance $d_p(u, v)$
  (for $p \in [1, \infty]$)

\[
d_p(u, v) := (|u_1 - v_1|^p + \cdots + |u_n - v_n|^p)^{1/p}
\]

\[
d_\infty(u, v) = \max\{|u_1 - v_1|, \ldots, |u_n - v_n|\}
\]
Geometric distance = metric

- **Distance** between vectors \( \mathbf{u}, \mathbf{v} \in \mathbb{R}^n \) \( \rightarrow \) (dis)similarity
  - \( \mathbf{u} = (u_1, \ldots, u_n) \)
  - \( \mathbf{v} = (v_1, \ldots, v_n) \)

- **Hamming** distance \( d_0(\mathbf{u}, \mathbf{v}) \) not very useful for DSM

- Extension of the Minkowski
  - \( p \)-distance \( d_p(\mathbf{u}, \mathbf{v}) \) (for \( 0 \leq p \leq 1 \))

\[
d_p(\mathbf{u}, \mathbf{v}) := |u_1 - v_1|^p + \cdots + |u_n - v_n|^p
\]

\[
d_0(\mathbf{u}, \mathbf{v}) = \# \{ i \mid u_i \neq v_i \}\]
Computing distances

Preparation: store “scored” matrix in DSM object

```r
> TT <- dsm.score(TT, score="freq", transform="log")
```
Computing distances

Preparation: store “scored” matrix in DSM object

```
> TT <- dsm.score(TT, score="freq", transform="log")
```

Compute distances between individual term pairs ...

```
> pair.distances(c("cat","cause"), c("animal","effect"), TT, method="euclidean")

    cat/animal cause/effect
  4.16       1.53
```
Computing distances

Preparation: store “scored” matrix in DSM object

```
> TT <- dsm.score(TT, score="freq", transform="log")
```

Compute distances between individual term pairs ...

```
> pair.distances(c("cat","cause"), c("animal","effect"),
    TT, method="euclidean")

  cat/animal cause/effect
       4.16      1.53
```

... or full distance matrix.

```
> dist.matrix(TT, method="euclidean")
> dist.matrix(TT, method="minkowski", p=4)
```
Distance and vector length = norm

- Intuitively, distance $d(u, v)$ should correspond to length $\|u - v\|$ of displacement vector $u - v$
  - $d(u, v)$ is a metric
  - $\|u - v\|$ is a norm
  - $\|u\| = d(u, 0)$
Distance and vector length = norm

Intuitively, distance $d(u, v)$ should correspond to length $\|u - v\|$ of displacement vector $u - v$

- $d(u, v)$ is a metric
- $\|u - v\|$ is a norm
- $\|u\| = d(u, 0)$

Any norm-induced metric is translation-invariant
Distance and vector length = norm

- Intuitively, distance $d(u, v)$ should correspond to length $\|u - v\|$ of displacement vector $u - v$
  - $d(u, v)$ is a metric
  - $\|u - v\|$ is a norm
  - $\|u\| = d(u, 0)$
- Any norm-induced metric is translation-invariant
- **Minkowski** $p$-norm with $d_p(u, v) = \|u - v\|_p$

\[
\|u\|_p := (|u_1|^p + \cdots + |u_n|^p)^{1/p}
\] for $1 \leq p$

\[
\|u\|_p := |u_1|^p + \cdots + |u_n|^p
\] for $0 \leq p < 1$

\[
\|u\|_\infty = \max\{|u_1|, \ldots, |u_n|\}
\]
Normalisation of row vectors

- Part 1: geometric distances only meaningful for vectors of the same length $\|x\|$
Normalisation of row vectors

- Part 1: geometric distances only meaningful for vectors of the same length $\|\mathbf{x}\|$.

- Normalize by scalar division: $\mathbf{x}' = \mathbf{x} / \|\mathbf{x}\| = (\frac{x_1}{\|\mathbf{x}\|}, \frac{x_2}{\|\mathbf{x}\|}, \ldots)$ with $\|\mathbf{x}'\| = 1$.

- Norm must be compatible with distance measure!

Two dimensions of English V–Obj DSM

$\alpha = 54.3^\circ$
Normalisation of row vectors

- Part 1: geometric distances only meaningful for vectors of the same length $\|x\|$.
- Normalize by scalar division: $x' = x/\|x\| = (x_1/\|x\|, x_2/\|x\|, \ldots)$ with $\|x'\| = 1$.
- Norm must be compatible with distance measure!
- Special case: scale $x \geq 0$ to stochastic vector with $\|x\|_1 = |x_1| + \cdots + |x_n|$.
  → probabilistic interpretation.

Two dimensions of English V–Obj DSM

\[ \alpha = 54.3^\circ \]
Norms and normalization

```r
> rowNorms(TT$S, method = "euclidean")

            cat     dog  animal   time   reason   cause  effect

> TT <- dsm.score(TT, score = "freq", transform = "log",
                  normalize = TRUE, method = "euclidean")

> rowNorms(TT$S, method = "euclidean")  # all = 1 now

> dist.matrix(TT, method = "euclidean")

            cat     dog  animal   time   reason   cause  effect
cat 0.000000 0.22400 0.47335 0.78218 1.12135 1.23895 1.16087
dog 0.224000 0.00000 0.39819 0.69821 1.06544 1.17850 1.11381
animal 0.473350 0.39819 0.00000 0.42615 0.84083 0.96906 0.85970
time 0.782180 0.69821 0.42615 0.00000 0.47537 0.58463 0.50174
reason 1.121350 1.06544 0.84083 0.47537 0.00000 0.27692 0.19792
cause 1.238950 1.17850 0.96906 0.58463 0.27692 0.00000 0.22426
effect 1.160871 1.11381 0.85970 0.50174 0.19792 0.22426 0.00000
```
Distance measures for non-negative vectors

- Information theory: **Kullback-Leibler (KL) divergence** for stochastic vectors (non-negative $x \geq 0$ and $\|x\|_1 = 1$)

\[
D(u\|v) = \sum_{i=1}^{n} u_i \cdot \log_2 \frac{u_i}{v_i}
\]
Distance measures for non-negative vectors

- Information theory: **Kullback-Leibler (KL) divergence** for stochastic vectors (non-negative $x \geq 0$ and $\|x\|_1 = 1$)

\[
D(u\|v) = \sum_{i=1}^{n} u_i \cdot \log_2 \frac{u_i}{v_i}
\]

- Properties of KL divergence
  - most appropriate for a probabilistic interpretation of $M$
  - zeroes in $v$ without corresponding zeroes in $u$ are problematic
  - **not symmetric**, unlike geometric distance measures
  - alternatives: skew divergence, Jensen-Shannon divergence
Distance measures for non-negative vectors

- Information theory: **Kullback-Leibler (KL) divergence** for stochastic vectors (non-negative $x \geq 0$ and $\|x\|_1 = 1$)

$$D(u\|v) = \sum_{i=1}^{n} u_i \cdot \log_2 \frac{u_i}{v_i}$$

- Properties of KL divergence
  - most appropriate for a probabilistic interpretation of $M$
  - zeroes in $v$ without corresponding zeroes in $u$ are problematic
  - **not symmetric**, unlike geometric distance measures
  - alternatives: skew divergence, Jensen-Shannon divergence

- A symmetric distance metric (Endres & Schindelin 2003)

$$D_{uv} = D(u\|z) + D(v\|z) \quad \text{with} \quad z = \frac{u + v}{2}$$
Similarity measures

Angle $\alpha$ between vectors $u, v \in \mathbb{R}^n$ is given by

$$\cos \alpha = \frac{\sum_{i=1}^{n} u_i \cdot v_i}{\sqrt{\sum_{i=1}^{n} u_i^2} \cdot \sqrt{\sum_{i=1}^{n} v_i^2}} = \frac{u^T v}{\|u\|_2 \cdot \|v\|_2}$$

Two dimensions of English V–Obj DSM

$\alpha = 54.3^\circ$
Similarity measures

- Angle $\alpha$ between vectors $u, v \in \mathbb{R}^n$ is given by

$$\cos \alpha = \frac{\sum_{i=1}^{n} u_i \cdot v_i}{\sqrt{\sum_i u_i^2} \cdot \sqrt{\sum_i v_i^2}} = \frac{u^T v}{\|u\|_2 \cdot \|v\|_2}$$

- **Cosine** measure of similarity: $\cos \alpha$
  - $\cos \alpha = 1 \rightarrow$ collinear
  - $\cos \alpha = 0 \rightarrow$ orthogonal

- Corresponding metric: **angular distance** $\alpha$
Euclidean distance or cosine similarity?

\[ d_2 (u, v) = \|u - v\|_2 = \sqrt{\sum_i (u_i - v_i)^2} \]

\[ = \sqrt{\sum_i u_i^2 + \sum_i v_i^2 - 2 \sum_i u_i v_i} \]

\[ = \sqrt{\|u\|^2 + \|v\|^2 - 2 u^T v} \]

\[ = \sqrt{2 - 2 \cos \phi} \]

\[ d_2 (u, v) \] is a monotonically increasing function of \( \phi \)
Similarity measures for non-negative vectors

- Generalized **Jaccard coefficient** = shared features

\[
J(u, v) = \frac{\sum_{i=1}^{n} \min\{u_i, v_i\}}{\sum_{i=1}^{n} \max\{u_i, v_i\}}
\]

- \(1 - J(u, v)\) is a distance **metric** (Kosub 2016)
Similarity measures for non-negative vectors

- **Generalized Jaccard coefficient** = shared features
  \[
  J(u, v) = \frac{\sum_{i=1}^{n} \min\{u_i, v_i\}}{\sum_{i=1}^{n} \max\{u_i, v_i\}}
  \]

- \(1 - J(u, v)\) is a distance **metric** (Kosub 2016)

- An asymmetric measure of feature **overlap** (Clarke 2009)
  \[
  o(u, v) = \frac{\sum_{i=1}^{n} \min\{u_i, v_i\}}{\sum_{i=1}^{n} u_i}
  \]
Building a distributional model

- pre-processed corpus with linguistic annotation
  - define target terms
  - context units or aggregates
  - geometric analysis
  - feature scaling
  - similarity/distance measure + normalization
  - probabilistic analysis
- term-term matrix
- term-context matrix
  - define target & feature terms
  - type & size of co-occurrence
  - embedding learned by neural network
- dimensionality reduction
Dimensionality reduction = model compression

- Co-occurrence matrix $\mathbf{M}$ is often unmanageably large and can be extremely sparse
  - Google Web1T5: $1\text{M} \times 1\text{M}$ matrix with one trillion cells, of which less than 0.05% contain nonzero counts (Evert 2010)
- Compress matrix by reducing dimensionality (= rows)
Dimensionality reduction = model compression

- Co-occurrence matrix $\mathbf{M}$ is often unmanageably large and can be extremely sparse
  - Google Web1T5: $1\text{M} \times 1\text{M}$ matrix with one trillion cells, of which less than 0.05% contain nonzero counts (Evert 2010)
- Compress matrix by reducing dimensionality (= rows)

- Feature selection: columns with high frequency & variance
  - measured by entropy, chi-squared test, nonzero count, . . .
  - may select similar dimensions and discard valuable information
Dimensionality reduction = model compression

- Co-occurrence matrix $\mathbf{M}$ is often unmanageably large and can be extremely sparse
  - Google Web1T5: $1\text{M} \times 1\text{M}$ matrix with one trillion cells, of which less than 0.05% contain nonzero counts (Evert 2010)
- Compress matrix by reducing dimensionality (= rows)

- **Feature selection**: columns with high frequency & variance
  - measured by entropy, chi-squared test, nonzero count, ...
  - may select similar dimensions and discard valuable information

- **Projection** into (linear) subspace
  - principal component analysis (PCA)
  - independent component analysis (ICA)
  - random indexing (RI)
  - intuition: preserve distances between data points
Dimensionality reduction & latent dimensions

Landauer & Dumais (1997) claim that LSA dimensionality reduction (and related PCA technique) uncovers **latent dimensions** by exploiting correlations between features.

- Example: term-term matrix
- V-Obj co-oc. extracted from BNC
  - targets = noun lemmas
  - features = verb lemmas
- feature scaling: association scores (SketchEngine log Dice)
- $k = 186$ nouns with $f_{\text{buy}} + f_{\text{sell}} \geq 25$
- $n = 2$ dimensions: $\text{buy}$ and $\text{sell}$

<table>
<thead>
<tr>
<th>noun</th>
<th>buy</th>
<th>sell</th>
</tr>
</thead>
<tbody>
<tr>
<td>antique</td>
<td>5.12</td>
<td>5.50</td>
</tr>
<tr>
<td>bread</td>
<td>5.96</td>
<td>3.99</td>
</tr>
<tr>
<td>computer</td>
<td>6.75</td>
<td>6.83</td>
</tr>
<tr>
<td>factory</td>
<td>4.95</td>
<td>4.72</td>
</tr>
<tr>
<td>group</td>
<td>4.93</td>
<td>4.28</td>
</tr>
<tr>
<td>jewellery</td>
<td>5.11</td>
<td>5.73</td>
</tr>
<tr>
<td>mill</td>
<td>5.14</td>
<td>5.41</td>
</tr>
<tr>
<td>people</td>
<td>3.00</td>
<td>4.26</td>
</tr>
<tr>
<td>record</td>
<td>6.81</td>
<td>6.68</td>
</tr>
<tr>
<td>souvenir</td>
<td>5.45</td>
<td>4.67</td>
</tr>
<tr>
<td>ticket</td>
<td>8.93</td>
<td>8.74</td>
</tr>
</tbody>
</table>
Dimensionality reduction & latent dimensions
Dimensionality reduction by PCA

variance = 3.35
Dimensionality reduction by PCA
Dimensionality reduction by PCA

variance = 0.98
Dimensionality reduction by PCA

variance = 2.14
Dimensionality reduction by PCA

variance = 2.37
Dimensionality reduction by PCA

variance = 2.43
Dimensionality reduction in practice

# SVD is the algorithm behind PCA dimensionality reduction
> TT2 <- dsm.projection(TT, n=2, method="svd")
> TT2

<table>
<thead>
<tr>
<th></th>
<th>svd1</th>
<th>svd2</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>-0.733</td>
<td>-0.6615</td>
</tr>
<tr>
<td>dog</td>
<td>-0.782</td>
<td>-0.6110</td>
</tr>
<tr>
<td>animal</td>
<td>-0.914</td>
<td>-0.3606</td>
</tr>
<tr>
<td>time</td>
<td>-0.993</td>
<td>0.0302</td>
</tr>
<tr>
<td>reason</td>
<td>-0.889</td>
<td>0.4339</td>
</tr>
<tr>
<td>cause</td>
<td>-0.817</td>
<td>0.5615</td>
</tr>
<tr>
<td>effect</td>
<td>-0.871</td>
<td>0.4794</td>
</tr>
</tbody>
</table>

> x <- TT2[, 1]  # first latent dimension
> y <- TT2[, 2]  # second latent dimension
> plot(x, y, pch=20, col="red",
       xlim=extendrange(x), ylim=extendrange(y))
> text(x, y, rownames(TT2), pos=3)
Dimensionality reduction as matrix factorization

- PCA is based on **singular value decomposition (SVD)**, which factorises any matrix \( M \) into

\[
M = U \Sigma V^T
\]

where \( U \) and \( V \) are orthogonal and \( \Sigma \) is a diagonal matrix of **singular values** \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m > 0 \)

\[
\begin{bmatrix}
  n \\
  k \\
  m \\
  \end{bmatrix}
= \begin{bmatrix}
  k \\
  m \\
  \end{bmatrix}
\cdot \begin{bmatrix}
  \sigma_1 \\
  m \\
  \Sigma \\
  \sigma_m \\
  \end{bmatrix}
\cdot \begin{bmatrix}
  m \\
  n \\
  V^T \\
  \end{bmatrix}
\]
Dimensionality reduction as matrix factorization

- Columns $a_i$ of $U$ and $b_i$ of $V$ (singular vectors) are orthogonal ($a_i^T a_j = 0$) and of unit length ($\|a_i\| = 1$)

- Key property: truncated SVD gives best least-squares approximation in $r$-dimensional subspace

\[
U_r \Sigma_r V_r^T = \begin{bmatrix}
\vdots & \vdots \\
\vdots & \vdots \\
\vdots & \vdots \\
\vdots & \vdots \\
\vdots & \vdots \\
U_r & \ldots & a_r
\end{bmatrix} \cdot \begin{bmatrix}
\sigma_1 \\
\vdots \\
\Sigma_r \\
\sigma_r \\
\vdots \\
\vdots
\end{bmatrix} \cdot \begin{bmatrix}
\ldots & \ldots & b_1 & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & b_r & \ldots & \ldots
\end{bmatrix}
\]
Dimensionality reduction as matrix factorization

- Truncated SVD as orthogonal projection

\[ \mathbf{MV}_r = \mathbf{U}_r \mathbf{\Sigma}_r = \begin{bmatrix} \vdots & \cdots & \vdots \\ \sigma_1 \mathbf{a}_1 & \cdots & \sigma_r \mathbf{a}_r \\ \vdots & \cdots & \vdots \end{bmatrix} \]

⇒ `method="svd" in dsm.projection()`

- \( \sigma_1^2 \geq \sigma_2^2 \geq \ldots \) = amount of distance information (i.e. variance of \( \mathbf{M} \)) captured by each latent dimension
Scaling latent dimensions

- Truncated SVD omits latent dimensions that capture relatively little distance information (here $r = 400$)
Scaling latent dimensions

- Truncated SVD omits latent dimensions that capture relatively little distance information (here $r = 400$)
- Skip first $k$ dimensions, e.g. $k = 50$ (Bullinaria & Levy 2012)
Scaling latent dimensions

- Truncated SVD omits latent dimensions that capture relatively little distance information (here $r = 400$)
- Skip first $k$ dimensions, e.g. $k = 50$ (Bullinaria & Levy 2012)
- Power-scaling of dimensions: $\sigma^P$ (Caron 2001)
  - Bullinaria & Levy (2012) report positive effect

![Power scaling P = 1/2](image)
Scaling latent dimensions

- Truncated SVD omits latent dimensions that capture relatively little distance information (here $r = 400$)
- Skip first $k$ dimensions, e.g. $k = 50$ (Bullinaria & Levy 2012)
- Power-scaling of dimensions: $\sigma^P$ (Caron 2001)
  - Bullinaria & Levy (2012) report positive effect
  - esp. with $P = 0$ to equalize dimensions (whitening)
Power-scaling in practice

```r
> TT2 <- dsm.projection(TT, n=2, method="svd", power=0)
> TT2

     svd1  svd2
cat   -0.322 -0.5110
dog   -0.343 -0.4721
animal -0.401 -0.2786
time   -0.436  0.0233
reason -0.390  0.3353
cause  -0.359  0.4338
effect -0.383  0.3704

# power-scaling can also be applied post-hoc
> sigma <- attr(TT2, "sigma")  # singular values
> scaleMargins(TT2, cols=sigma^0.5)  # P = 1/2
> scaleMargins(TT2, cols=sigma)  # unscaled (P = 1)
```
Other matrix factorization techniques

- **Non-negative matrix factorization (NMF)**
  - $U$ and $V$ are stochastic matrices ($a_i \geq 0$ and $\|a_i\|_1 = 1$)
  - cross-entropy instead of least-squares approximation
  - iterative algorithm with random initialisation for rank-$r$ approximation (≠ sequence of ordered components)
Other matrix factorization techniques

- **Non-negative matrix factorization (NMF)**
  - $U$ and $V$ are stochastic matrices ($a_i \geq 0$ and $\|a_i\|_1 = 1$)
  - cross-entropy instead of least-squares approximation
  - iterative algorithm with random initialisation for rank-$r$ approximation ($\neq$ sequence of ordered components)

- NMF of term-document matrix $\iff$ LDA topic model

$$U \Sigma V^T = \sigma_1 a_1 b_1^T + \sigma_2 a_2 b_2^T + \sigma_3 a_3 b_3^T + \ldots$$

  - $a_i =$ probability distribution of words in $i$-th topic
  - $b_i =$ distribution of topic across documents
Other matrix factorization techniques

- **Non-negative matrix factorization (NMF)**
  - \( U \) and \( V \) are stochastic matrices (\( a_i \geq 0 \) and \( \|a_i\|_1 = 1 \))
  - cross-entropy instead of least-squares approximation
  - iterative algorithm with random initialisation for rank-\( r \) approximation (\( \neq \) sequence of ordered components)

- NMF of term-document matrix \( \iff \) LDA topic model

\[
U \Sigma V^T = \sigma_1 a_1 b_1^T + \sigma_2 a_2 b_2^T + \sigma_3 a_3 b_3^T + \ldots
\]

- \( a_i \) = probability distribution of words in \( i \)-th topic
- \( b_i \) = distribution of topic across documents

- Levy et al. (2015, 213) show that **word2vec** embeddings implicitly factorize a shifted PPMI matrix
  - sigmoid loss function, weighted towards high frequencies
  - similarly, **GloVe** (Pennington et al. 2014) factorizes matrix of conditional probabilities with a frequency-weighted least-squares approximation
Outline

DSM parameters
- A taxonomy of DSM parameters
- Context type & size
- Feature scaling
- Measuring distance
- Dimensionality reduction

Building a DSM
- Sparse matrices
- Example: a verb-object DSM

Appendix
- Examples
- Three famous examples
Scaling up to the real world

- So far, we have worked on minuscule toy models
- We want to scale up to real world data sets now
Scaling up to the real world

▶ So far, we have worked on minuscule toy models
☞ We want to scale up to real world data sets now

▶ Example 1: window-based DSM on BNC content words
  ▶ 83,926 lemma types with \( f \geq 10 \)
  ▶ term-term matrix with \( 83,926 \cdot 83,926 = 7 \) billion entries
  ▶ standard representation requires 56 GB of RAM (8-byte floats)
  ▶ only 22.1 million non-zero entries (\( \approx 0.32\% \))
Scaling up to the real world

- So far, we have worked on minuscule toy models
  - We want to scale up to real world data sets now

- Example 1: window-based DSM on BNC content words
  - 83,926 lemma types with $f \geq 10$
  - term-term matrix with $83,926 \times 83,926 = 7$ billion entries
  - standard representation requires 56 GB of RAM (8-byte floats)
  - only 22.1 million non-zero entries ($= 0.32\%$)

- Example 2: Google Web 1T 5-grams (1 trillion words)
  - more than 1 million word types with $f \geq 2500$
  - term-term matrix with 1 trillion entries requires 8 TB RAM
  - only 400 million non-zero entries ($= 0.04\%$)
### Sparse matrix representation

▶ Invented example of a **sparsely populated** DSM matrix

<table>
<thead>
<tr>
<th></th>
<th>eat</th>
<th>get</th>
<th>hear</th>
<th>kill</th>
<th>see</th>
<th>use</th>
</tr>
</thead>
<tbody>
<tr>
<td>boat</td>
<td>.</td>
<td>59</td>
<td>.</td>
<td>.</td>
<td>39</td>
<td>23</td>
</tr>
<tr>
<td>cat</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>26</td>
<td>58</td>
<td>.</td>
</tr>
<tr>
<td>cup</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>dog</td>
<td>33</td>
<td>.</td>
<td>42</td>
<td>.</td>
<td>83</td>
<td>.</td>
</tr>
<tr>
<td>knife</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>84</td>
</tr>
<tr>
<td>pig</td>
<td>9</td>
<td>.</td>
<td>.</td>
<td>27</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>
Sparse matrix representation

▶ Invented example of a **sparsely populated** DSM matrix

<table>
<thead>
<tr>
<th></th>
<th>eat</th>
<th>get</th>
<th>hear</th>
<th>kill</th>
<th>see</th>
<th>use</th>
</tr>
</thead>
<tbody>
<tr>
<td>boat</td>
<td>.</td>
<td>59</td>
<td>.</td>
<td>.</td>
<td>39</td>
<td>23</td>
</tr>
<tr>
<td>cat</td>
<td>.</td>
<td>.</td>
<td>26</td>
<td>58</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>cup</td>
<td>.</td>
<td>98</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>dog</td>
<td>33</td>
<td>.</td>
<td>42</td>
<td>.</td>
<td>83</td>
<td>.</td>
</tr>
<tr>
<td>knife</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>84</td>
</tr>
<tr>
<td>pig</td>
<td>9</td>
<td>.</td>
<td>27</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

▶ Store only non-zero entries in compact **sparse matrix format**

<table>
<thead>
<tr>
<th>row</th>
<th>col</th>
<th>value</th>
<th>row</th>
<th>col</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>59</td>
<td>4</td>
<td>1</td>
<td>33</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>39</td>
<td>4</td>
<td>3</td>
<td>42</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>23</td>
<td>4</td>
<td>5</td>
<td>83</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>26</td>
<td>5</td>
<td>6</td>
<td>84</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>58</td>
<td>6</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>98</td>
<td>6</td>
<td>4</td>
<td>27</td>
</tr>
</tbody>
</table>
Working with sparse matrices

- Compressed format: each row index (or column index) stored only once, followed by non-zero entries in this row (or column)
  - convention: column-major matrix (data stored by columns)

- Specialised algorithms for sparse matrix algebra
  - especially matrix multiplication, solving linear systems, etc.
  - take care to avoid operations that create a dense matrix!
Working with sparse matrices

- Compressed format: each row index (or column index) stored only once, followed by non-zero entries in this row (or column)
  - convention: *column-major* matrix (data stored by columns)

- Specialised algorithms for sparse matrix algebra
  - especially matrix multiplication, solving linear systems, etc.
  - take care to avoid operations that create a dense matrix!

- **R** implementation: Matrix package
  - essential for real-life distributional semantics
  - *wordspace* provides additional support for sparse matrices (vector distances, sparse SVD, …)

- Other software: Matlab, Octave, Python + SciPy
Outline

**DSM parameters**
- A taxonomy of DSM parameters
- Context type & size
- Feature scaling
- Measuring distance
- Dimensionality reduction

**Building a DSM**
- Sparse matrices
  - Example: a verb-object DSM

**Appendix**
- Examples
- Three famous examples
Triplet tables

- A sparse DSM matrix can be represented as a table of triplets (target, feature, co-occurrence frequency)
  - for syntactic co-occurrence and term-document matrices, marginals can be computed from a complete triplet table
  - for surface and textual co-occurrence, marginals have to be provided in separate files (see `?read.dsm.triplet`)

<table>
<thead>
<tr>
<th>noun</th>
<th>rel</th>
<th>verb</th>
<th>f</th>
<th>mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>subj</td>
<td>bite</td>
<td>3</td>
<td>spoken</td>
</tr>
<tr>
<td>dog</td>
<td>subj</td>
<td>bite</td>
<td>12</td>
<td>written</td>
</tr>
<tr>
<td>dog</td>
<td>obj</td>
<td>bite</td>
<td>4</td>
<td>written</td>
</tr>
<tr>
<td>dog</td>
<td>obj</td>
<td>stroke</td>
<td>3</td>
<td>written</td>
</tr>
</tbody>
</table>

- **DSM_VerbNounTriples_BNC** contains additional information
  - syntactic relation between noun and verb
  - written or spoken part of the British National Corpus
Constructing a DSM from a triplet table

- Additional information can be used for filtering (verb-object relation), or aggregate frequencies (spoken + written BNC)

```r
> tri <- subset(DSM_VerbNounTriples_BNC, rel == "obj")
```

- Construct DSM object from triplet input
  - `raw.freq=TRUE` indicates raw co-occurrence frequencies (rather than a pre-weighted DSM)
  - constructor aggregates counts from duplicate entries
  - marginal frequencies are automatically computed

```r
> VObj <- dsm(target=tri$noun, feature=tri$verb, 
  score=tri$f, raw.freq=TRUE)
> VObj # inspect marginal frequencies (e.g. head(VObj$rows, 20))
```
Exploring the DSM

```r
> VObj <- dsm.score(VObj, score="MI", normalize=TRUE)

> nearest.neighbours(VObj, "dog")  # angular distance
   horse   cat  animal  rabbit  fish  guy
   73.9  75.9  76.2  77.0  77.2  78.5
   cichlid  kid  bee  creature
   78.6  79.0  79.1  79.5

> nearest.neighbours(VObj, "dog", method="manhattan")
# NB: we used an incompatible Euclidean normalization!

> VObj50 <- dsm.projection(VObj, n=50, method="svd")
> nearest.neighbours(VObj50, "dog")
```
Practice

- Code examples and further explanations: `hands_on_day2.R`
- How many different models can you build from `DSM_VerbNounTriples_BNC`?
  - apply different filters, scores, transformations and metrics
  ☞ explore nearest neighbours of selected words
- Build real-life DSMs from pre-compiled co-occurrence data
  - [http://wordspace.collocations.de/doku.php/course:material](http://wordspace.collocations.de/doku.php/course:material)
  - load pre-compiled matrix and apply different parameters
  ☞ compare nearest neighbours or semantic maps
- Learn how to import your own co-occurrence data
  ☞ `hands_on_day2_inputFormats.R`
  - download example data sets to subdirectory `data/`
- Explore matrix factorization techniques
  ☞ `hands_on_day2_matrix_factorization.R`
Outline

DSM parameters
  A taxonomy of DSM parameters
  Context type & size
  Feature scaling
  Measuring distance
  Dimensionality reduction

Building a DSM
  Sparse matrices
  Example: a verb-object DSM

Appendix
  Examples
  Three famous examples
Some well-known DSM examples

Latent Semantic Analysis (Landauer & Dumais 1997)

- term-context matrix with document context
- weighting: log term frequency and term entropy
- distance measure: cosine
- dimensionality reduction: SVD

Hyperspace Analogue to Language (Lund & Burgess 1996)

- term-term matrix with surface context
- structured (left/right) and distance-weighted frequency counts
- distance measure: Minkowski metric ($1 \leq p \leq 2$)
- dimensionality reduction: feature selection (high variance)
## Some well-known DSM examples

### Latent Semantic Analysis (Landauer & Dumais 1997)
- term-context matrix with document context
- weighting: log term frequency and term entropy
- distance measure: cosine
- dimensionality reduction: SVD

### Hyperspace Analogue to Language (Lund & Burgess 1996)
- term-term matrix with surface context
- structured (left/right) and distance-weighted frequency counts
- distance measure: Minkowski metric \( (1 \leq p \leq 2) \)
- dimensionality reduction: feature selection (high variance)
Some well-known DSM examples

Infomap NLP (Widdows 2004)

- term-term matrix with unstructured surface context
- weighting: none
- distance measure: cosine
- dimensionality reduction: SVD
### Some well-known DSM examples

#### Infomap NLP (Widdows 2004)
- term-term matrix with unstructured surface context
- weighting: none
- distance measure: cosine
- dimensionality reduction: SVD

#### Random Indexing (Karlgren & Sahlgren 2001)
- term-term matrix with unstructured surface context
- weighting: various methods
- distance measure: various methods
- dimensionality reduction: random indexing (RI)
Some well-known DSM examples

Dependency Vectors (Padó & Lapata 2007)

- term-term matrix with unstructured dependency context
- weighting: log-likelihood ratio
- distance measure: PPMI-weighted Dice (Lin 1998)
- dimensionality reduction: none
Some well-known DSM examples

**Dependency Vectors (Padó & Lapata 2007)**

- term-term matrix with unstructured dependency context
- weighting: log-likelihood ratio
- distance measure: PPMI-weighted Dice (Lin 1998)
- dimensionality reduction: none

**Distributional Memory (Baroni & Lenci 2010)**

- term-term matrix with structured and unstructured dependencies + knowledge patterns
- weighting: local-MI on type frequencies of link patterns
- distance measure: cosine
- dimensionality reduction: none
... and an unexpected application

Authorship attribution (Burrows 2002)

- Burrows’s Delta method is very popular in modern literary stylometry and authorship attribution (Evert et al. 2017)
- document-term matrix with word forms as features
- weighting: relative frequency of word form in document
- feature selection: 200–5,000 most frequent words (mfw)
- columns are standardized ($\mu = 0, \sigma^2 = 1$) $\Rightarrow$ z-scores
- clustering of documents based on various distance metrics (or nearest-neighbour classifier for known authors)
- dimensionality reduction: none
- main result: angle/cosine $\gg$ Manhattan $\gg$ Euclidean
Outline

DSM parameters
A taxonomy of DSM parameters
Context type & size
Feature scaling
Measuring distance
Dimensionality reduction

Building a DSM
Sparse matrices
Example: a verb-object DSM

Appendix
Examples
Three famous examples
Latent Semantic Analysis (Landauer & Dumais 1997)

- Corpus: 30,473 articles from Grolier’s Academic American Encyclopedia (4.6 million words in total)
  - articles were limited to first 2,000 characters
- Word-article frequency matrix for 60,768 words
  - row vector shows frequency of word in each article
- Logarithmic frequencies scaled by word entropy
- Reduced to 300 dim. by singular value decomposition (SVD)
  - borrowed from LSI (Dumais et al. 1988)
  - central claim: SVD reveals latent semantic features, not just a data reduction technique
- Evaluated on TOEFL synonym test (80 items)
  - LSA model achieved 64.4% correct answers
  - also simulation of learning rate based on TOEFL results

- Corpus: ≈ 60 million words of news messages
  - from the New York Times News Service
- Word-word co-occurrence matrix
  - 20,000 target words & 2,000 context words as features
  - row vector records how often each context word occurs close to the target word (co-occurrence)
  - co-occurrence window: left/right 50 words (Schütze 1998) or ≈ 1000 characters (Schütze 1992)
- Rows weighted by inverse document frequency (tf.idf)
- Context vector = centroid of word vectors (bag-of-words)
  - goal: determine “meaning” of a context
- Reduced to 100 SVD dimensions (mainly for efficiency)
- Evaluated on unsupervised word sense induction by clustering of context vectors (for an ambiguous word)
  - induced word senses improve information retrieval performance
HAL (Lund & Burgess 1996)

- HAL = Hyperspace Analogue to Language
- Corpus: 160 million words from newsgroup postings
- Word-word co-occurrence matrix
  - same 70,000 words used as targets and features
  - co-occurrence window of 1 – 10 words
- Separate counts for left and right co-occurrence
  - i.e. the context is *structured*
- In later work, co-occurrences are weighted by (inverse) distance (Li *et al.* 2000)
  - but no dimensionality reduction
- Applications include construction of semantic vocabulary maps by multidimensional scaling to 2 dimensions
HAL (Lund & Burgess 1996)

Figure 2. Multidimensional scaling of co-occurrence vectors.
Appendix

Three famous examples

References I


References II


Evert, Stefan (2010). Google Web 1T5 n-grams made easy (but not for the computer). In Proceedings of the 6th Web as Corpus Workshop (WAC-6), pages 32–40, Los Angeles, CA.

Evert, Stefan; Proisl, Thomas; Jannidis, Fotis; Reger, Isabella; Pielström, Steffen; Schöch, Christof; Vitt, Thorsten (2017). Understanding and explaining Delta measures for authorship attribution. Digital Scholarship in the Humanities, 22(suppl_2), ii4–ii16.
References III


References IV


References V

