Distributional Semantic Models
Part 2: The parameters of a DSM

Stefan Evert¹
with Alessandro Lenci², Marco Baroni³ and Gabriella Lapesa⁴

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

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

Copyright © 2009–2017 Evert, Lenci, Baroni & Lapesa | Licensed under CC-by-sa version 3.0

Outline

DSM parameters
A taxonomy of DSM parameters
Examples

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

DSM parameters
General definition of DSMs
Mathematical notation:
▶ k × n co-occurrence matrix M ∈ R^{k×n} (example: 7×6)
▶ k rows = target terms
▶ n columns = features or dimensions

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, …
Outline

DSM parameters
A taxonomy of DSM parameters

Examples

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

Overview of DSM parameters

pre-processed corpus with linguistic annotation

define target & feature terms

define target terms

type & size of co-occurrence

classification tokens or types

type & size of co-occurrence

Feature scaling

similarity/distance measure + normalization

geometric analysis

probabilistic analysis

embedding learned by neural network

dimensionality reduction

Term-context matrix

Term-context matrix records frequency of term in each individual context (e.g. sentence, document, Web page, encyclopaedia article)

\[
F = \begin{bmatrix}
\cdot \ f_1 \cdot \\
\cdot \ f_2 \cdot \\
\vdots \\
\cdot \ f_k \cdot \\
\end{bmatrix}
\]

\[
\begin{array}{c|c|c|c|c|c|c}
\text{context} & \text{cat} & \text{dog} & \text{animal} & \text{time} & \text{reason} & \text{cause} \\
\hline
\text{Felis} & 10 & 10 & 7 & - & - & - \\
\text{Pet} & - & 10 & 4 & 11 & - & - \\
\text{Feed} & 2 & 15 & 10 & 2 & - & - \\
\text{Block} & - & - & - & 2 & 1 & - \\
\text{Philosophy} & - & - & - & - & 1 & 4 \\
\text{Kant} & - & - & - & 2 & 1 & 2 \\
\text{Back pain} & - & - & - & 1 & - & 1 \\
\end{array}
\]

\[
> \text{TC} \leftarrow \text{DSM} \_\text{TermContext}
\]

> head(TC, Inf) # extract full co-oc matrix from DSM object

Term-term matrix

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

\[
M = \begin{bmatrix}
\cdot \ m_1 \cdot \\
\cdot \ m_2 \cdot \\
\vdots \\
\cdot \ m_k \cdot \\
\end{bmatrix}
\]

\[
\begin{array}{c|c|c|c|c|c|c|c}
\text{context} & \text{cat} & \text{dog} & \text{animal} & \text{time} & \text{reason} & \text{cause} & \text{effect} \\
\hline
\text{breed} & 83 & 17 & 7 & 37 & - & 1 & - \\
\text{kill} & 361 & 13 & 30 & 60 & 1 & 2 & 4 \\
\text{kill} & 42 & 10 & 109 & 134 & 13 & 5 & 5 \\
\text{cause} & 19 & 9 & 29 & 117 & 81 & 34 & 109 \\
\text{time} & 1 & - & 14 & 68 & 6 & 47 & 1 \\
\text{important} & - & 1 & 4 & 55 & 34 & 55 & 17 \\
\text{important} & - & 1 & 6 & 60 & 35 & 17 & 1 \\
\end{array}
\]

\[
> \text{TT} \leftarrow \text{DSM} \_\text{TermTerm}
\]

> head(TT, Inf)
Term-term matrix

Some footnotes:
- Often target terms ≠ feature terms
  - e.g. nouns described by co-occurrences with verbs as features
  - identical sets of target & feature terms ⇒ symmetric matrix
- Different types of co-occurrence (Evert 2008)
  - surface context (word or character window)
  - textual context (non-overlapping segments)
  - syntactic context (dependency relation)
- Can be seen as smoothing of term-context matrix
  - average over similar contexts (with same context terms)
  - data sparseness reduced, except for small windows
  - we will take a closer look at the relation between term-context and term-term models in part 5 of this tutorial

Definition of target and feature terms

- Choice of linguistic unit
  - words
  - bigrams, trigrams, …
  - multiword units, named entities, phrases, …
  - morphemes
  - word pairs (esp. analogy tasks)
- 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_{liver} vs. bank_{finance})
  - abstraction: POS tags (or bigrams) as feature terms
- Trade-off between deeper linguistic analysis and
  - need for language-specific resources
  - possible errors introduced at each stage of the analysis
Effects of linguistic annotation

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>arrivare</td>
<td>attendere</td>
</tr>
<tr>
<td>raggiugmento</td>
<td>arrivo (noun)</td>
</tr>
<tr>
<td>raggiunto</td>
<td>ricevere</td>
</tr>
<tr>
<td>trovar</td>
<td>accontentare</td>
</tr>
<tr>
<td>raggiunge</td>
<td>approdare</td>
</tr>
<tr>
<td>arrivasse</td>
<td>pervenire</td>
</tr>
<tr>
<td>arrivèrà</td>
<td>venire</td>
</tr>
<tr>
<td>concludere</td>
<td>piombare</td>
</tr>
</tbody>
</table>

[http://clic.cimec.unitn.it/infomap-query/](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) do not provide reliable distributional information (too much "noise")
- Frequency-based selection
  - minimum corpus frequency: $f \geq F_{\text{min}}$
  - or accept $n_w$ most frequent terms
  - sometimes also upper threshold: $F_{\text{min}} \leq f \leq F_{\text{max}}$
- Relevance-based selection
  - criterion from IR: document frequency $df$
  - terms with high $df$ are too general → uninformative
  - terms with very low $df$ may be too sparse to be useful
- Other criteria
  - POS-based filter: no function words, only verbs, ...

Overview of DSM parameters

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. [L3/R3 span, $k = 6$]

Parameters:
- span size (in words or characters)
- symmetric vs. one-sided span
- uniform or “triangular” (distance-based) weighting
- 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>▶ fox</td>
<td>▶ bitch</td>
</tr>
<tr>
<td>▶ pig</td>
<td>▶ terrier</td>
</tr>
<tr>
<td>▶ mongrel</td>
<td>▶ rottweiler</td>
</tr>
<tr>
<td>▶ animal</td>
<td>▶ canine</td>
</tr>
<tr>
<td>▶ sheep</td>
<td>▶ cat</td>
</tr>
<tr>
<td>▶ rabbit</td>
<td>▶ to bark</td>
</tr>
<tr>
<td>▶ pig</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:

▶ type of linguistic unit
  ▶ sentence
  ▶ paragraph
  ▶ turn in a conversation
  ▶ Web page

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 and Lapata 2007)
▶ direct vs. indirect dependency paths
  ▶ direct dependencies
  ▶ direct + indirect dependencies
▶ homogeneous data (e.g. only verb-object) vs. heterogeneous data (e.g. all children and parents of the verb)
▶ maximal length of dependency path

“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)
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 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.

<table>
<thead>
<tr>
<th></th>
<th>bite-l</th>
<th>bite-r</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>man</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Structured vs. unstructured dependency context

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

<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>

Comparison

- Unstructured context
  - data less sparse (e.g. man kills and kills man both map to the kill dimension of the vector $\mathbf{x}_{\text{man}}$)

- Structured context
  - more sensitive to semantic distinctions (kill-subj and kill-obj are rather different things!)
  - dependency relations provide a form of syntactic “typing” of the DSM dimensions (the “subject” dimensions, the “recipient” dimensions, etc.)
  - important to account for word-order and compositionality
Overview of DSM parameters

- **Context tokens vs. context types**
  - Features are usually context **tokens**, i.e. individual instances
    - document, Wikipedia article, Web page, ...
    - paragraph, sentence, tweet, ...
    - "co-occurrence" count = frequency of term in context token
  - Can also be generalised to context **types**, e.g.
    - type = cluster of near-duplicate documents
    - type = syntactic structure of sentence (ignoring content)
    - type = tweets from same author
    - frequency counts from all instances of type are aggregated
  - Context types may be anchored at individual tokens
    - n-gram of words (or POS tags) around target
    - subcategorisation pattern of target verb
    - overlaps with (generalisation of) syntactic co-occurrence

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**
  - $O$ = 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

- **Expected co-occurrence frequency**
  \[
  E = \frac{R \cdot C}{N} \leftrightarrow O
  \]
Obtaining marginal frequencies

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

- **Syntactic co-occurrence**
  - # of dependency instances in which target/feature participates
  - $N = \text{total number of dependency instances}$
  - 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
  - $N = \text{total # of context units}$

Marginal frequencies in wordspace

DSM objects in wordspace 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
```

Geometric vs. probabilistic interpretation

- **Geometric interpretation**
  - row vectors as points or arrows in $n$-dimensional space
  - very intuitive, good for visualisation
  - use techniques from geometry and matrix algebra

- **Probabilistic interpretation**
  - co-occurrence matrix as observed sample statistic that is "explained" by a generative probabilistic model
  - e.g. probabilistic LSA (Hoffmann 1999), Latent Semantic Clustering (Rooth et al. 1999), Latent Dirichlet Allocation (Blei et al. 2003), etc.
  - explicitly accounts for random variation of frequency counts
  - recent work: neural word embeddings

NB: focus on geometric interpretation in this tutorial
Overview of DSM parameters

- pre-processed corpus with linguistic annotation
- define target terms
- define target & feature terms
- context tokens or types
- type & size of co-occurrence
- similarity/distance measure + normalization
- dimensionality reduction
- geometric analysis
- probabilistic analysis
- embedding learned by neural network

Feature scaling

- Logarithmic scaling: $O' = \log(O + 1)$
  (cf. Weber-Fechner law for human perception)
- Relevance weighting, e.g. tf.idf (information retrieval)
  \[
  tf.idf = tf \cdot \log(D/df) 
  \]
  - $tf = \text{co-occurrence frequency } O$
  - $df = \text{document frequency of feature (or nonzero count)}$
  - $D = \text{total number of documents (or row count of } M)$
- Statistical association measures: take frequency of target term and feature into account
  - often based on comparison of observed and expected co-occurrence frequency
  - measures differ in how they balance $O$ and $E$

Simple association measures

- pointwise Mutual Information (MI)
  \[
  MI = \log_2 \frac{O}{E} 
  \]
- local MI
  \[
  \text{local-MI} = O \cdot MI = O \cdot \log_2 \frac{O}{E} 
  \]
- 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>sgkj</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 ($\approx$ 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 simple association measures (AMs) available
- Further AMs computed from full contingency tables, see
  - Evert (2008)
  - http://www.collocations.de/
  - http://sigil.r-forge.r-project.org/
Applying association scores in \textsc{wordspace}

\begin{verbatim}
> options(digits=3) # print fractional values with limited precision
> dsm.score(TT, score="MI", sparse=FALSE, matrix=TRUE)

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

- sparseness of the matrix has been lost!
- cells with score $x = -\infty$ are inconvenient
- distribution of scores may be even more skewed than co-occurrence frequencies (esp. for local-MI)

\begin{verbatim}
> dsm.score(TT, score="simple-ll", matrix=TRUE)
> dsm.score(TT, score="simple-ll", transf="log", matrix=T)
> dsm.score(TT, score="freq", transform="log", matrix=T)
\end{verbatim}

Association scores & transformations in \textsc{wordspace}

\begin{verbatim}
> ?dsm.score # read help page for available parameter settings
\end{verbatim}
Scaling of column vectors

- In statistical analysis and machine learning, features are usually centred and scaled so that
  - mean $\mu = 0$
  - variance $\sigma^2 = 1$
- In DSM research, this step is less common for columns of $\mathbf{M}$
  - centring is a prerequisite for certain dimensionality reduction and data analysis techniques (esp. PCA)
  - but co-occurrence matrix no longer sparse!
  - scaling may give too much weight to rare features
- $\mathbf{M}$ cannot be row-normalised and column-scaled at the same time (result depends on ordering of the two steps)

Geometric distance = metric

- **Distance** between vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ (dis)similarity
  - $\mathbf{u} = (u_1, \ldots, u_n)$
  - $\mathbf{v} = (v_1, \ldots, v_n)$
  - **Euclidean** distance $d_2(\mathbf{u}, \mathbf{v})$
  - “City block” **Manhattan** distance $d_1(\mathbf{u}, \mathbf{v})$
- Both are special cases of the **Minkowski** $p$-distance $d_p(\mathbf{u}, \mathbf{v})$ (for $p \in [1, \infty]$)
  \[
  d_p(\mathbf{u}, \mathbf{v}) := (|u_1 - v_1|^p + \cdots + |u_n - v_n|^p)^{1/p}
  \]
  \[
  d_\infty(\mathbf{u}, \mathbf{v}) = \max\{|u_1 - v_1|, \ldots, |u_n - v_n|\}
  \]
Computing distances

**Preparation: store “scored” matrix in DSM object**

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

Compute distances between individual term pairs ...

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

... or full distance matrix.

```r
> dist.matrix(TT, method="euclidean")
> dist.matrix(TT, method="minkowski", p=4)
```

Normalisation of row vectors

- Geometric distances only meaningful for vectors of the same length \(|x|\)
- Normalize by scalar division: 
  \[ x' = x/|x| = (\frac{x_1}{|x|}, \frac{x_2}{|x|}, \ldots) \]
  with \(|x'| = 1 \)
- Norm must be compatible with distance measure!
- Special case: scale to relative frequencies with 
  \[ |x|_1 = |x_1| + \cdots + |x_n| \]
  probabilistic interpretation

```

\[ \|x\|_p := (|x_1|^p + \cdots + |x_n|^p)^{1/p} \]
```

Distance and vector length = norm

- Intuitively, distance \(d(u,v)\) should correspond to length \(|u - v|\) of displacement vector \(u - v\)
  \[ d(u,v) \text{ is a metric} \]
  \[ |u - v| \text{ is a norm} \]
  \[ |u| = d(u,0) \]

Such a metric is always translation-invariant

- \(d_p(u,v) = \|u - v\|_p\)
- **Minkowski \(p\)-norm** for \(p \in [1, \infty)\) (not \(p < 1\):
  \[ |u|_p := (|u_1|^p + \cdots + |u_n|^p)^{1/p} \]

```

Norms and normalization

```
> rowNorms(TT$S, method="euclidean")
  cat   dog   animal  time  reason  cause  effect
  6.90  8.96  8.82  10.29  8.13  6.86  6.52
```

```
> 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  0.000 0.224 0.473 0.782 1.121 1.239 1.161
  dog  0.224 0.000 0.398 0.698 1.065 1.179 1.113
  animal 0.473 0.398 0.000 0.426 0.841 0.971 0.860
  time 0.782 0.698 0.426 0.000 0.475 0.585 0.502
  reason 1.121 1.065 0.841 0.475 0.000 0.277 0.198
  cause 1.239 1.179 0.971 0.585 0.277 0.000 0.224
  effect 1.161 1.113 0.860 0.502 0.198 0.224 0.000
```

Two dimensions of English V−Obj DSM

- α = 54.3°
  - Geometric distance
  - origin
  - Geometric distances only meaningful for vectors of the same length \(|x|\)
  - Normalize by scalar division:
    \[ x' = x/|x| = (\frac{x_1}{|x|}, \frac{x_2}{|x|}, \ldots) \]
    with \(|x'| = 1 \)
  - Norm must be compatible with distance measure!
  - Special case: scale to relative frequencies with 
    \[ |x|_1 = |x_1| + \cdots + |x_n| \]
    probabilistic interpretation

\[ \|x\|_p := (|x_1|^p + \cdots + |x_n|^p)^{1/p} \]
Other distance measures

- Information theory: Kullback-Leibler (KL) divergence for probability vectors ($\mathbb{R}^n$, non-negative, $\|x\|_1 = 1$)

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

- Properties of KL divergence
  - most appropriate in 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 measure (Endres and 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

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

- cosine measure of similarity: $\cos \alpha$
  - $\cos \alpha = 1 \Rightarrow$ collinear
  - $\cos \alpha = 0 \Rightarrow$ orthogonal

- Corresponding metric: angular distance $\alpha$

Dimensionality reduction = model compression

- Co-occurrence matrix $M$ is often unmanageably large and can be extremely sparse
  - Google Web1T5: $1M \times 1M$ 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
  - joint selection of multiple features is useful but expensive

- 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 and 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 cooc’s extracted from BNC
  - targets = noun lemmas
  - features = verb lemmas
- feature scaling: association scores (modified log Dice coefficient)
- \( k = 111 \) nouns with \( f \geq 20 \)
  - (must have non-zero row vectors)
- \( n = 2 \) dimensions: buy and sell

<table>
<thead>
<tr>
<th>noun</th>
<th>buy</th>
<th>sell</th>
</tr>
</thead>
<tbody>
<tr>
<td>bond</td>
<td>0.28</td>
<td>0.77</td>
</tr>
<tr>
<td>cigarette</td>
<td>-0.52</td>
<td>0.44</td>
</tr>
<tr>
<td>dress</td>
<td>0.51</td>
<td>-1.30</td>
</tr>
<tr>
<td>freehold</td>
<td>-0.01</td>
<td>-0.08</td>
</tr>
<tr>
<td>land</td>
<td>1.13</td>
<td>1.54</td>
</tr>
<tr>
<td>number</td>
<td>-1.05</td>
<td>-1.02</td>
</tr>
<tr>
<td>pub</td>
<td>-0.08</td>
<td>-1.30</td>
</tr>
<tr>
<td>share</td>
<td>1.92</td>
<td>1.99</td>
</tr>
<tr>
<td>system</td>
<td>-1.63</td>
<td>-0.70</td>
</tr>
</tbody>
</table>

Motivating latent dimensions & subspace projection

- The latent property of being a commodity is “expressed” through associations with several verbs: sell, buy, acquire, ...
- Consequence: these DSM dimensions will be correlated

- Identify latent dimension by looking for strong correlations (or weaker correlations between large sets of features)
- Projection into subspace \( V \) of \( k < n \) latent dimensions as a “noise reduction” technique ➔ LSA
- Assumptions of this approach:
  - “latent” distances in \( V \) are semantically meaningful
  - other “residual” dimensions represent chance co-occurrence patterns, often particular to the corpus underlying the DSM

Centering the data set

- Uncentered data set
- Centered data set
- Variance of centered data
Centering the data set

- Uncentered data set
- Centered data set
- Variance of centered data

\[ \sigma^2 = \frac{1}{k-1} \sum_{i=1}^{k} \| x^{(i)} \|^2 \]

Projection and preserved variance: examples

- Uncentered data set
- Centered data set
- Variance of centered data

\[ \text{variance} = 0.36 \]

\[ \text{variance} = 0.72 \]

\[ \text{variance} = 1.26 \]
Projection and preserved variance: examples

```
> TT2 <- dsm.projection(TT, n=2, method="svd")
> text(TT2, rownames(TT2), pos=3)
```

Dimensionality reduction in practice

```
svd1  svd2
cat  -0.733 -0.6615
dog   -0.782 -0.6110
animal -0.914  0.3606
time  -0.993  0.0302
reason -0.889  0.4339
cause  -0.817  0.5615
effect  -0.871  0.4794
```

Orthogonal PCA dimensions

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

Outline

```
Building a DSM
Sparse matrices
Example: a verb-object DSM
```
### Some well-known DSM examples

#### Latent Semantic Analysis (Landauer and 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 and 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)

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

#### Random Indexing (Karlgren and Sahlgren 2001)
- term-term matrix with unstructured surface context
- weighting: various methods
- distance measure: various methods
- dimensionality reduction: random indexing (RI)

#### Dependency Vectors (Padó and 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 and 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
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>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>39</td>
<td>23</td>
</tr>
<tr>
<td>cat</td>
<td>.</td>
<td>.</td>
<td>26</td>
<td>58</td>
<td>.</td>
</tr>
<tr>
<td>cup</td>
<td>98</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>dog</td>
<td>33</td>
<td>42</td>
<td>83</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>knife</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>
</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!

- 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
- Examples

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

**Exploring the DSM**

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

```r
> nearest.neighbours(VObj, "dog")  # angular distance
```

```r
  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
```

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

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

**References I**


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.
References II


References III
