Distributional word representations

Christopher Potts

CS 244U: Natural language understanding
April 7

Related materials

full bibliograpy for these slides

- Core reading: (Turney and Pantel 2010)
- Assignment: http://www.stanford.edu/class/cs224u/hw/hw1/
- Code/data:
  - http://stanford.edu/class/cs224u/hw/hw1/cs224u-hw1.zip
    or
  - /afs/ir/class/cs224u/hwcode/hw1/
  - (Expanded with a word × document matrix.)

A corpus in matrix form

Upper left corner of a matrix derived from the training portion of this IMDB data release: http://ai.stanford.edu/~amaas/data/sentiment/.

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
against	0	0	0	1	0	0	3	2	3	0
age	0	0	0	1	0	3	1	0	4	0
agent	0	0	0	0	0	0	0	0	0	0
ages	0	0	0	0	0	2	0	0	0	0
ago	0	0	0	2	0	0	0	0	3	0
agree	0	1	0	0	0	0	0	0	0	0
ahead	0	0	0	1	0	0	0	0	0	0
ain't	0	0	0	0	0	0	0	0	0	0
air	0	0	0	0	0	0	0	0	0	0
aka	0	0	0	1	0	0	0	0	0	0

Guiding hypotheses (Turney and Pantel 2010:153)

Statistical semantics hypothesis: Statistical patterns of human word usage can be used to figure out what people mean (Weaver, 1955; Furnas et al., 1983). – If units of text have similar vectors in a text frequency matrix,[13] then they tend to have similar meanings. (We take this to be a general hypothesis that subsumes the four more specific hypotheses that follow.)

Bag of words hypothesis: The frequencies of words in a document tend to indicate the relevance of the document to a query (Salton et al., 1975). – If documents and pseudo-documents (queries) have similar column vectors in a term–document matrix, then they tend to have similar meanings.

Distributional hypothesis: Words that occur in similar contexts tend to have similar meanings (Harris, 1954; Firth, 1957; Deerwester et al., 1990). – If words have similar row vectors in a word–context matrix, then they tend to have similar meanings.

Extended distributional hypothesis: Patterns that co-occur with similar pairs tend to have similar meanings (Lin & Pantel, 2001). – If patterns have similar column vectors in a pair–pattern matrix, then they tend to express similar semantic relations.

Latent relation hypothesis: Pairs of words that co-occur in similar patterns tend to have similar semantic relations (Turney et al., 2003). – If word pairs have similar row vectors in a pair–pattern matrix, then they tend to have similar semantic relations.

Overview: great power, a great many design choices

Matrix type		Weighting		Dimensionality reduction		Vector comparison
word × document		probabilities		LSA		Euclidean
word × word		length normalization		PLSA		Cosine
word × search proximity	×	TF-IDF	×	LDA	×	Dice
adj. × modified noun		PMI		PCA		Jaccard
word × dependency rel.		Positive PMI		IS		KL
verb × arguments		PPMI with discounting		DCA		KL with skew
⋮		⋮		⋮		⋮

(Nearly the full cross-product to explore; only a handful of the combinations are ruled out mathematically, and the literature contains relatively little guidance.)

Overview: great power, a great many design choices

tokenization
annotation
tagging
parsing
feature selection
⋮ cluster texts by date/author/discourse context/...
⇓ ⇙

Matrix type		Weighting		Dimensionality reduction		Vector comparison
word × document		probabilities		LSA		Euclidean
word × word		length normalization		PLSA		Cosine
word × search proximity	×	TF-IDF	×	LDA	×	Dice
adj. × modified noun		PMI		PCA		Jaccard
word × dependency rel.		Positive PMI		IS		KL
verb × arguments		PPMI with discounting		DCA		KL with skew
⋮		⋮		⋮		⋮

(Nearly the full cross-product to explore; only a handful of the combinations are ruled out mathematically, and the literature contains relatively little guidance.)

General questions for vector-space modelers

- How do the rows (words, phrase-types, . . .) relate to each other?
- How do the columns (contexts, documents, . . .) relate to each other?
- For a given group of documents *D*, which words epitomize *D*?
- For a given a group of words *W*, which documents epitomize *W* (IR)?

Goals of semantics (from class meeting 2)

1. Word meanings
2. Connotations
3. Compositionality
4. Syntactic ambiguities
5. Semantic ambiguities
6. Entailment and monotonicity
7. Question answering

Other resources for word meanings

1. WordNet (Miller 1995; Fellbaum 1998)
   - http://wordnet.princeton.edu
   - http://compprag.christopherpotts.net/wordnet.html
   - http://www.stanford.edu/class/cs224u/slides/2013/cs224u-2013-lec02.pdf
2. GlobalWordNet: http://www.globalwordnet.org
3. Harvard General Inquirer (Stone et al. 1966)
   - http://wjh.harvard.edu/~inquirer/spreadsheet_guide.htm
   - http://wjh.harvard.edu/~inquirer/homecat.htm
4. FrameNet (Fillmore and Baker 2001)
   - https://framenet.icsi.berkeley.edu/fndrupal/
5. . . .

Matrix designs

- I'm going to set aside pre-processing issues like tokenization — the best approach there will be tailored to your application.
- I'm going to assume that we would prefer not to do feature selection based on counts, stopword dictionaries, etc. — our VSMs should sort these things out for us!
- For more designs: Turney and Pantel 2010:§2.1–2.5, §6

Word × document

Upper left corner of a matrix derived from the training portion of this IMDB data release: `http://ai.stanford.edu/~amaas/data/sentiment/`.

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
against	0	0	0	1	0	0	3	2	3	0
age	0	0	0	1	0	3	1	0	4	0
agent	0	0	0	0	0	0	0	0	0	0
ages	0	0	0	0	0	2	0	0	0	0
ago	0	0	0	2	0	0	0	0	3	0
agree	0	1	0	0	0	0	0	0	0	0
ahead	0	0	0	1	0	0	0	0	0	0
ain't	0	0	0	0	0	0	0	0	0	0
air	0	0	0	0	0	0	0	0	0	0
aka	0	0	0	1	0	0	0	0	0	0

Word × word

Upper left corner of a matrix derived from the training portion of this IMDB data release: http://ai.stanford.edu/~amaas/data/sentiment/.

	against	age	agent	ages	ago	agree	ahead	ain.t	air	aka	al
against	2003	90	39	20	88	57	33	15	58	22	24
age	90	1492	14	39	71	38	12	4	18	4	39
agent	39	14	507	2	21	5	10	3	9	8	25
ages	20	39	2	290	32	5	4	3	6	1	6
ago	88	71	21	32	1164	37	25	11	34	11	38
agree	57	38	5	5	37	627	12	2	16	19	14
ahead	33	12	10	4	25	12	429	4	12	10	7
ain't	15	4	3	3	11	2	4	166	0	3	3
air	58	18	9	6	34	16	12	0	746	5	11
aka	22	4	8	1	11	19	10	3	5	261	9
al	24	39	25	6	38	14	7	3	11	9	861

Word × discourse context

Upper left corner of an interjection × dialog-act tag matrix derived from the Switchboard Dialog Act Corpus (Stolcke et al. 2000):
http://compprag.christopherpotts.net/swda-clustering.html

	%	+	^2	^g	^h	^q	aa
absolutely	0	2	0	0	0	0	95
actually	17	12	0	0	1	0	4
anyway	23	14	0	0	0	0	0
boy	5	3	1	0	5	2	1
bye	0	1	0	0	0	0	0
bye-bye	0	0	0	0	0	0	0
dear	0	0	0	0	1	0	0
definitely	0	2	0	0	0	0	56
exactly	2	6	1	0	0	0	294
gee	0	3	0	0	2	1	1
goodness	1	0	0	0	2	0	0

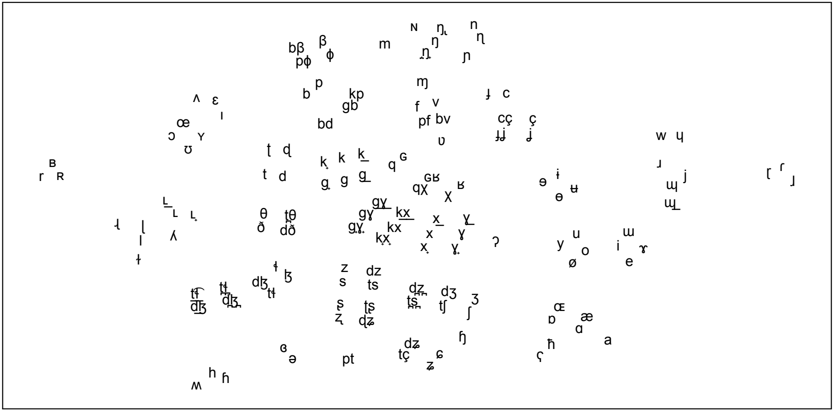
Phonological segment × feature values

Derived from http://www.linguistics.ucla.edu/people/hayes/120a/.
Dimensions: (141 × 28).

	syllabic	stress	long	consonantal	sonorant	continuant	delayed.release	approximant	tap	trill	...
ɒ	1	−1	−1	−1	1	1	0	1	−1	−1	
ɑ	1	−1	−1	−1	1	1	0	1	−1	−1	
ɶ	1	−1	−1	−1	1	1	0	1	−1	−1	
a	1	−1	−1	−1	1	1	0	1	−1	−1	
æ	1	−1	−1	−1	1	1	0	1	−1	−1	...
ʌ	1	−1	−1	−1	1	1	0	1	−1	−1	
ɔ	1	−1	−1	−1	1	1	0	1	−1	−1	
o	1	−1	−1	−1	1	1	0	1	−1	−1	
ɤ	1	−1	−1	−1	1	1	0	1	−1	−1	
ə	1	−1	−1	−1	1	1	0	1	−1	−1	
⋮						⋮					

Phonological segment × feature values

Derived from http://www.linguistics.ucla.edu/people/hayes/120a/.
Dimensions: (141 × 28).



Other designs

- word × search query
- word × syntactic context
- pair × pattern (e.g., *mason* : *stone*, *cuts*)
- adj. × modified noun
- word × dependency rel.
- person × product
- word × person
- word × word × pattern
- verb × subject × object

$\vdots$

Loading the R code and data

Enter the Python shell while in the directory containing the code and data:

1.
```
# Load the code for this unit:
from vsm import *
```
2.
```
# The matrix is large and so might take a while to load:
m = Matrix('imdb-wordword.csv')
```

Vector distance measures

- All the definitions are in terms of *distance* measures. They can be turned into similarity measures by subtracting appropriate constants.
- Examples focus on row vectors; the definitions and assessments hold for column-wise comparisons as well.
- Further reading:
  - van Rijsbergen 1979:§3
  - Manning and Schütze 1999:§8.5
  - Lee 1999
  - Bullinaria and Levy 2007
  - Turney and Pantel 2010:§4.4–4.5

Euclidean distance

Definition (Euclidean distance)

Between vectors x and y of dimension n: $\sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$

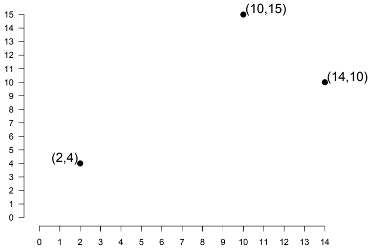
	d_x	d_y
A	2	4
B	10	15
C	14	10

Euclidean distance

Definition (Euclidean distance)

Between vectors x and y of dimension n: $\sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$

	d_x	d_y
A	2	4
B	10	15
C	14	10

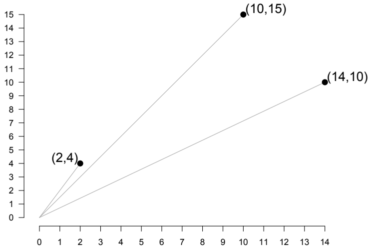


Euclidean distance

Definition (Euclidean distance)

Between vectors x and y of dimension n: $\sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$

	d_x	d_y
A	2	4
B	10	15
C	14	10

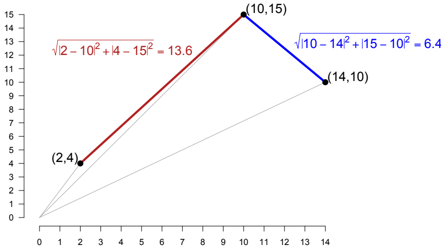


Euclidean distance

Definition (Euclidean distance)

Between vectors x and y of dimension n: $\sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$

	d_x	d_y
A	2	4
B	10	15
C	14	10



Euclidean distance

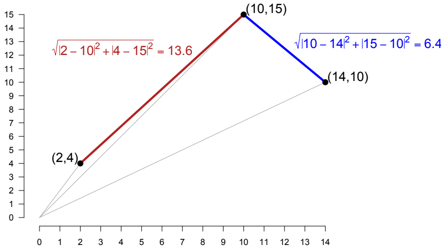
Definition (Euclidean distance)

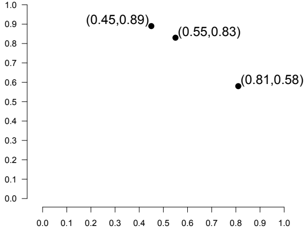
Between vectors x and y of dimension n: $\sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$

	d_x	d_y
A	2	4
B	10	15
C	14	10

L2 norm the rows $\Longrightarrow$

	d_x	d_y
A	0.45	0.89
B	0.55	0.83
C	0.81	0.58





Euclidean distance

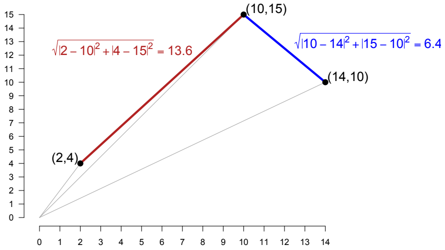
Definition (Euclidean distance)

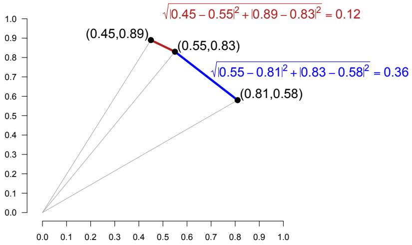
Between vectors x and y of dimension n: $\sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$

	d_x	d_y
A	2	4
B	10	15
C	14	10

L2 norm the rows $\Longrightarrow$

	d_x	d_y
A	0.45	0.89
B	0.55	0.83
C	0.81	0.58



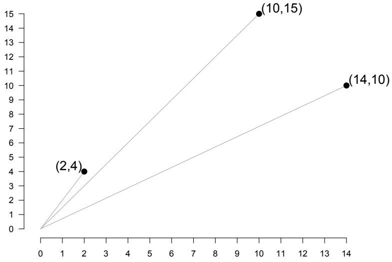


Cosine distance

Definition (Cosine distance)

Between vectors x and y of dimension n: $1 - \dfrac{\sum_{i=1}^{n} x_i \times y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \times \sqrt{\sum_{i=1}^{n} y_i^2}}$

	d_x	d_y
A	2	4
B	10	15
C	14	10

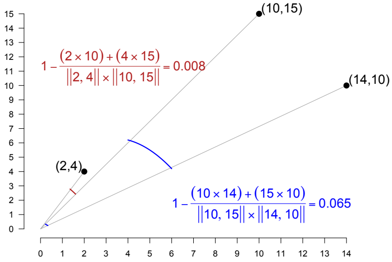


Cosine distance

Definition (Cosine distance)

Between vectors x and y of dimension n: $1 - \frac{\sum_{i=1}^{n} x_i \times y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \times \sqrt{\sum_{i=1}^{n} y_i^2}}$

	d_x	d_y
A	2	4
B	10	15
C	14	10



Cosine distance

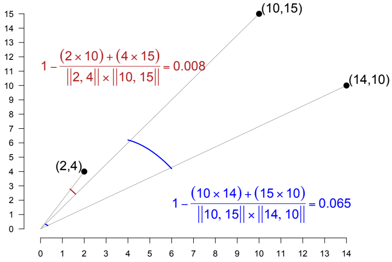
Definition (Cosine distance)

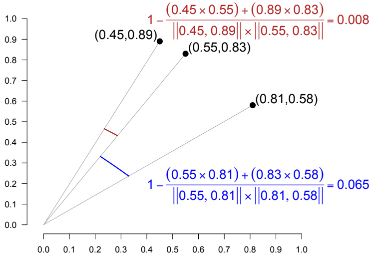
Between vectors x and y of dimension n: $1 - \frac{\sum_{i=1}^{n} x_i \times y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \times \sqrt{\sum_{i=1}^{n} y_i^2}}$

	d_x	d_y
A	2	4
B	10	15
C	14	10

L2 norm has no effect $\Longrightarrow$

	d_x	d_y
A	0.45	0.89
B	0.55	0.83
C	0.81	0.58





Dice and Jaccard distances

Definition (Dice distance; Dice 1945)

Between vectors x and y of dimension n:

$$1 - \frac{2 \times \sum_{i=1}^{n} \min(x_i, y_i)}{\sum_{i=1}^{n} x_i + y_i}$$

Alternatively, define a mapping S_n from vectors to sets such that $S_n(v) = \{v_i > n\}$ for $n \geqslant 0$, and use $1 - \frac{2 \times |S_n(x) \cap S_n(y)|}{|S_n(x)| + |S_n(y)|}$

Definition (Jaccard distance)

Between vectors x and y of dimension n:

$$\frac{\sum_{i=1}^{n} \min(x_i, y_i)}{\sum_{i=1}^{n} \max(x_i, y_i)}$$

Alternatively, with S_n from above, use $\frac{|S_n(x) \cap S_n(y)|}{|S_n(x) \cup S_n(y)|}$

- Jaccard and Dice give different numerical values, with Jaccard penalizing large numerical differences more, but the two deliver identical rankings (van Rijsbergen 1979:§3; Lee 1999).
- Cosine distance penalizes large numerical differences less than both (Manning and Schütze 1999:299).

KL divergence

Definition (KL divergence)

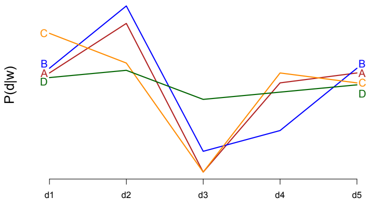
Between probability distributions p and q: $$D(p \parallel q) = \sum_{i=1}^{n} p_i \log\left(\frac{p_i}{q_i}\right)$$

p is the reference distribution. Before calculation, map all 0s to ϵ.

	d_1	d_2	d_3	d_4	d_5
A	10	15	0	9	10
B	5	8	1	2	5
C	14	11	0	10	9
D	13	14	10	11	12

Rows to prob. dists. $\Rightarrow$

	d_1	d_2	d_3	d_4	d_5
A	0.23	0.34	0.00	0.20	0.23
B	0.24	0.38	0.05	0.10	0.24
C	0.32	0.25	0.00	0.23	0.20
D	0.22	0.23	0.17	0.18	0.20



Word	KL distance from A	Rank
A	0.00	1
C	0.03	2
B	0.10	3
D	0.19	4

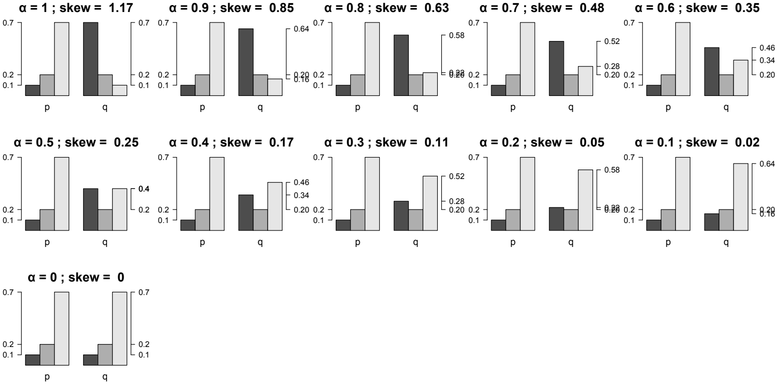
KL divergence with skew

Definition (α skew; Lee 1999)

Between probability distributions p and q:

$$\text{Skew}_{\alpha}(p, q) = D(p \parallel \alpha q + (1-\alpha)p)$$

$$p = [0.1, 0.2, 0.7] \qquad q = [0.7, 0.2, 0.1] \qquad D(p \parallel q) = 1.17$$



Relationships and generalizations

1. Euclidean, Jaccard, and Dice with raw count vectors will tend to favor raw frequency over distributional patterns.
2. Euclidean with L2-normed vectors is equivalent to cosine w.r.t. ranking (Manning and Schütze 1999:301).
3. Jaccard and Dice are equivalent w.r.t. ranking.
4. Both L2-norms and probability distributions can obscure differences in the amount/strength of evidence, which can in turn have an effect on the reliability of cosine, normed-euclidean, and KL divergence. These shortcomings might be addressed through weighting schemes.
5. Skew is KL but with a preliminary step that gives special credence to the reference distribution.

Other vector distance measures

For vectors x and y of dimension n

Let $X = S_n(x)$ and $Y = S_n(y)$, where $S_n(v) = \{v_i > n\}$ for $n \geqslant 0$.

- Matching coefficient (counts): $\sum_{i=1}^{n} \min(x_i, y_i)$
- Matching coefficient (binary): $\left|X \cap Y\right|$
- Overlap (counts): $\frac{\sum_{i=1}^{n} \min(x_i, y_i)}{\min\left(\sum_{i=1}^{n} x_i\ ,\ \sum_{i=1}^{n} y_i\right)}$
- Overlap (binary): $\frac{\left|X \cap Y\right|}{\min\left(|X|\ ,\ |Y|\right)}$
- Manhattan distance: $\sum_{i=1}^{n} |x_i - y_y|$

For probability distributions p and q

- Symmetric KL: $D(p \parallel q) + D(q \parallel p)$
- Jensen-Shannon: $\frac{1}{2}D(p \parallel \frac{p+q}{2}) + \frac{1}{2}D(q \parallel \frac{p+q}{2})$

Distance calculations

```
import numpy as np
# Define the vectors we used before:
a = np.array([2, 4])
b = np.array([10, 15])
c = np.array([14, 10])
# Get their Euclidean distances:
euclidean_distance(a, b)
euclidean_distance(a, c)
euclidean_distance(b, c)
# Compare with cosine:
cosine_distance(a,b)
```

Lexical neighbors experiments

```
# Sort by closeness to 'joy'; n=None returns the whole vocab
1 joy = neighbors(m,'joy',distfunc=euclidean_distance,n=None)
2 [x[0] for x in joy[:  5]]
3 ['joy', 'emotionally', 'ordinary', 'delivered', 'attitude']
4 [x[0] for x in joy[-5:  ]]
5 ['to', 'of', 'a', 'and', 'the']
```

Exploration

1. We saw that euclidean distance favors raw frequencies. Find words in the matrix that help make this point: a pair that are semantically unrelated but close according to `euclidean_distance`, and a pair that are semantically related by far apart according to `euclidean_distance`.
2. To what extent does using `cosine_distance` address the problem you uncovered in the previous exercise.

The semantic orientation method

1. Get your VSM into shape by weighting and/or dimensionality reduction.
2. Define two seed-sets S_1 and S_2 of words (they should be opposing in some way that is appropriate for your matrix).
3. For a given distance metric *dist* and word *w*:

$$\left(\sum_{w' \in S_1} dist(w, w')\right) - \left(\sum_{w' \in S_2} dist(w, w')\right)$$

The semantic orientation method

1. Get your VSM into shape by weighting and/or dimensionality reduction.
2. Define two seed-sets S_1 and S_2 of words (they should be opposing in some way that is appropriate for your matrix).
3. For a given distance metric *dist* and word *w*:

$$\left(\sum_{w' \in S_1} dist(w, w')\right) - \left(\sum_{w' \in S_2} dist(w, w')\right)$$

Turney and Littman's (2003:343) hypothesis

The ideas in SO-A can likely be extended to many other semantic aspects of words. The General Inquirer lexicon has 182 categories of word tags [Stone et al. 1966] and this paper has only used two of them, so there is no shortage of future work.

Using our code

```
# Load the source code:
neg = ['bad', 'nasty', 'poor', 'negative',
       'unfortunate', 'wrong', 'inferior']
pos = ['good', 'nice', 'excellent', 'positive',
       'fortunate', 'correct', 'superior']
so = semantic_orientation(m, seeds1=neg, seeds2=pos)
# Most negative:
[x[0] for x in so[: 5]]
['1/10', 'suck', 'gonna', 'crap', 'renting']
# Most positive:
[x[0] for x in so[-5: ]]
['breathtaking', 'titanic', 'victoria', 'powell', 'columbo']
```

Does your preferred matrix design (HW 1, question 3) do better?

Pos/neg semantic orientation results (top and bottom 15)

My preferred design:

Neighbor	Score
bad	−1.22
worst	−1.13
awful	−1.10
waste	−1.02
terrible	−1.02
worse	−1.00
horrible	−0.95
crap	−0.95
wrong	−0.95
stupid	−0.93
avoid	−0.90
pointless	−0.89
even	−0.89
garbage	−0.88
pathetic	−0.88

Neighbor	Score
excellent	1.17
nice	0.93
great	0.89
superior	0.83
well	0.76
very	0.74
perfect	0.71
role	0.67
performance	0.67
always	0.66
correct	0.66
good	0.65
fantastic	0.65
job	0.65
superb	0.64

Weighting and normalization

- This section focusses on methods for adjusting the counts in a matrix to better capture the underlying reationships.
- The examples are given in terms of word $\times$ document matrices, focussing on row-wise comparisons in places.
- The methods can also be applied column-wise, and to other kinds of matrices, though some (design, weighting) combos are better than others, as we will see.
- Further reading:
  - Manning and Schütze 1999:§15.2
  - Bullinaria and Levy 2007
  - Turney and Pantel 2010:§4.2

Goal of reweighting and related questions

- The goal of reweighting is to amplify the important, trustworthy, an unusual, while deemphasizing the mundane and the quirky.
- Absent a defined objective function, this will remain fuzzy.
- The intuition behind moving away from raw counts is that frequency is a poor proxy for the above values.
- So we should ask of each weighting scheme:
  - How does it compare to the raw count values?
  - How does it compare to the word frequencies?
  - What overall distribution of values does it deliver?

Relative frequencies

	d_1	d_2	d_3	d_4	d_5
A	10	15	0	9	10
B	5	8	1	2	5
C	14	11	0	10	9
D	13	14	10	11	12

Rows to $P(d|w)$ $\Rightarrow$

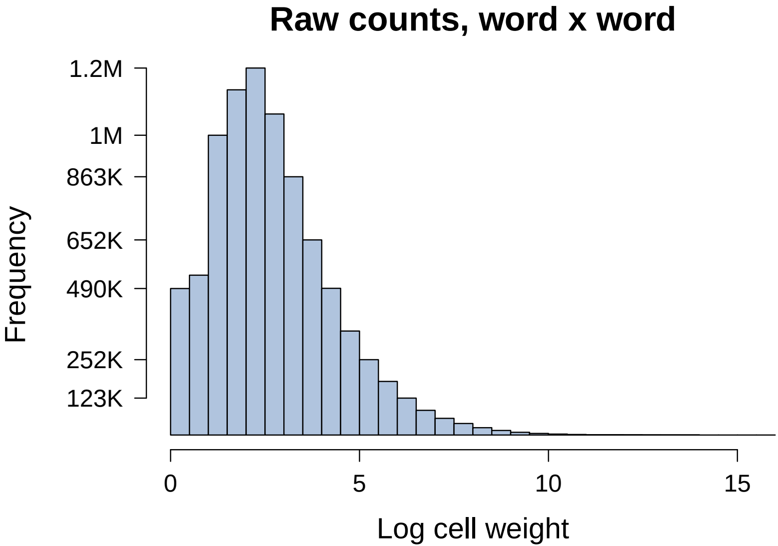
	d_1	d_2	d_3	d_4	d_5
A	0.23	0.34	0.00	0.20	0.23
B	0.24	0.38	0.05	0.10	0.24
C	0.32	0.25	0.00	0.23	0.20
D	0.22	0.23	0.17	0.18	0.20

Columns to $P(w|d)$ $\Downarrow$

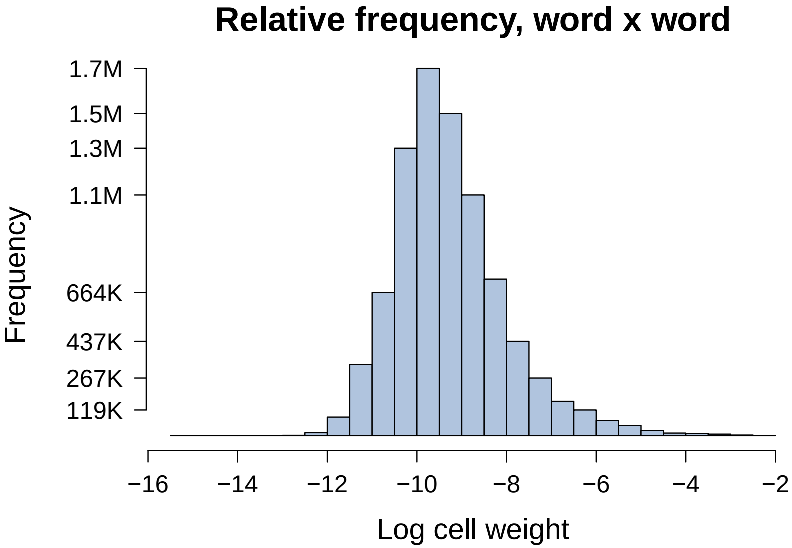
	d_1	d_2	d_3	d_4	d_5
A	0.24	0.31	0.00	0.28	0.28
B	0.12	0.17	0.09	0.06	0.14
C	0.33	0.23	0.00	0.31	0.25
D	0.31	0.29	0.91	0.34	0.33

Dangers of prob. values: exaggerated estimates for small counts; comparisons that ignore differences in magnitude

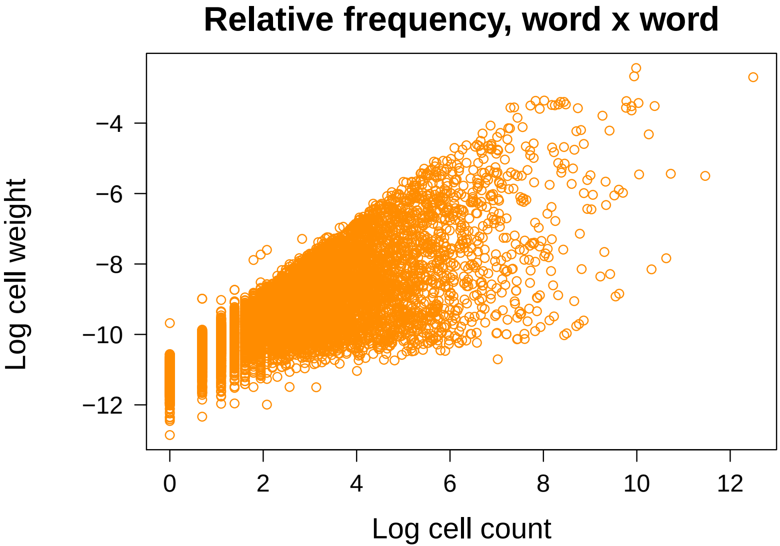
Relative frequencies compared to counts



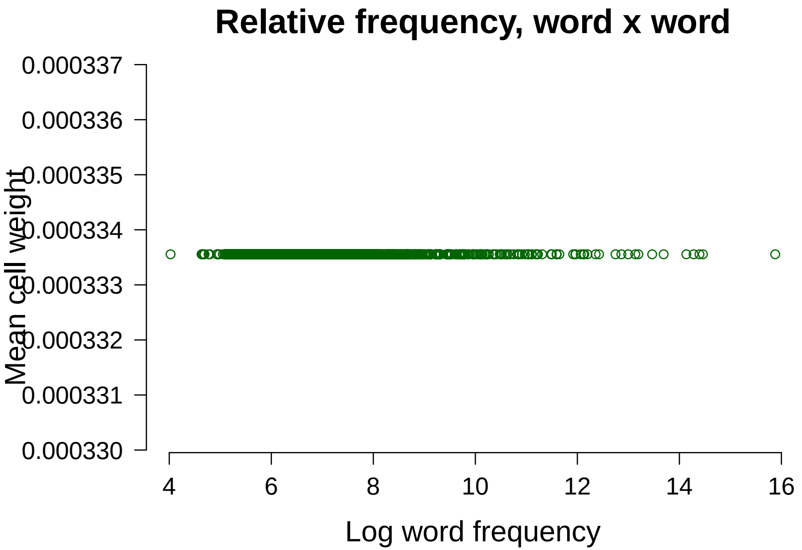
Relative frequencies compared to counts



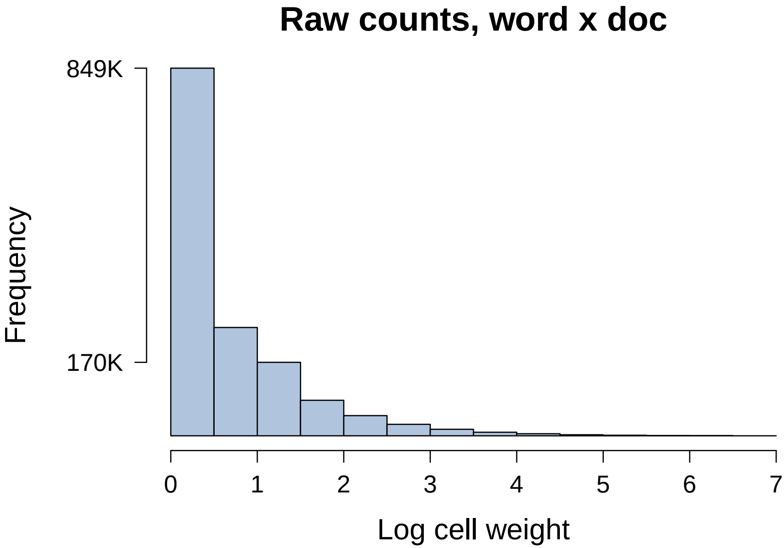
Relative frequencies compared to counts



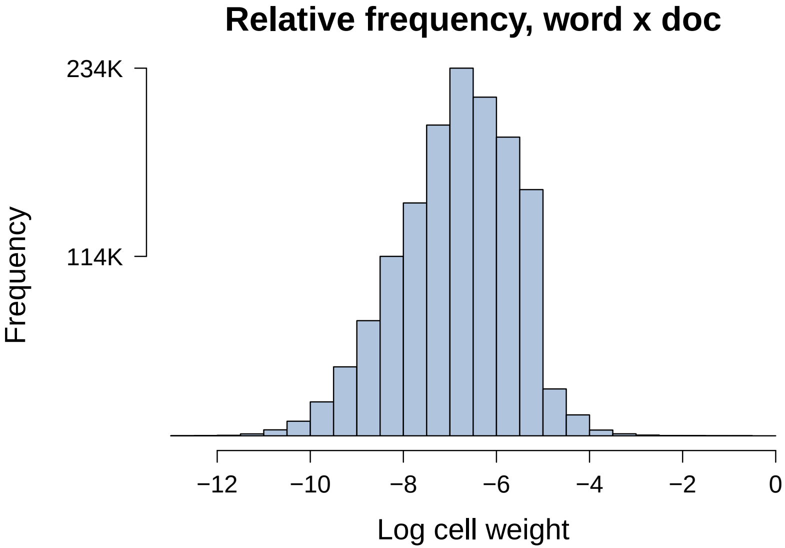
Relative frequencies compared to counts



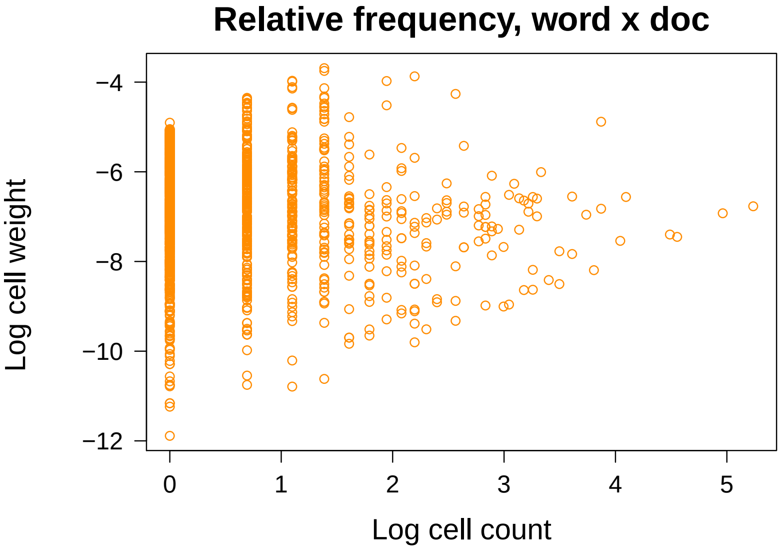
Relative frequencies compared to counts



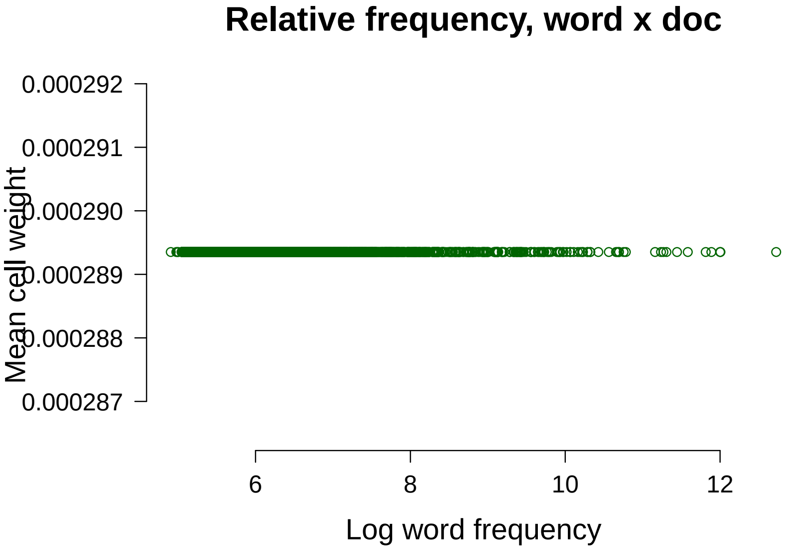
Relative frequencies compared to counts



Relative frequencies compared to counts



Relative frequencies compared to counts



Length (L2) normalization

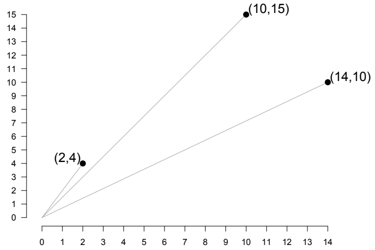
Definition (L2 normalization)

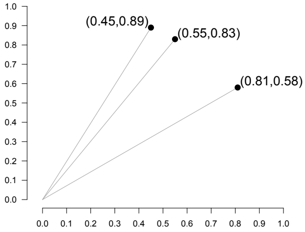
Given a vector x of dimension n, the normalization of x is a vector $\hat{x}$ also of dimension n obtained by dividing each element of x by $\sqrt{\sum_{i=1}^{n} x_i^2}$.

	d_x	d_y
A	2	4
B	10	15
C	14	10

L2 norm the rows $\Rightarrow$

	d_x	d_y
A	0.45	0.89
B	0.55	0.83
C	0.81	0.58





Term Frequency–Inverse Document Frequency (TF-IDF)

Definition (TF-IDF)

For a corpus of documents D:

- Term frequency (TF): $P(w|d)$
- Inverse document frequency (IDF): $\log\left(\frac{|D|}{|\{d \in D | w \in d\}|}\right)$ (assume $\log(0) = 0$)
- TF-IDF: TF $\times$ IDF

	d_1	d_2	d_3	d_4
A	10	10	10	10
B	10	10	10	0
C	10	10	0	0
D	0	0	0	1

$\Rightarrow$

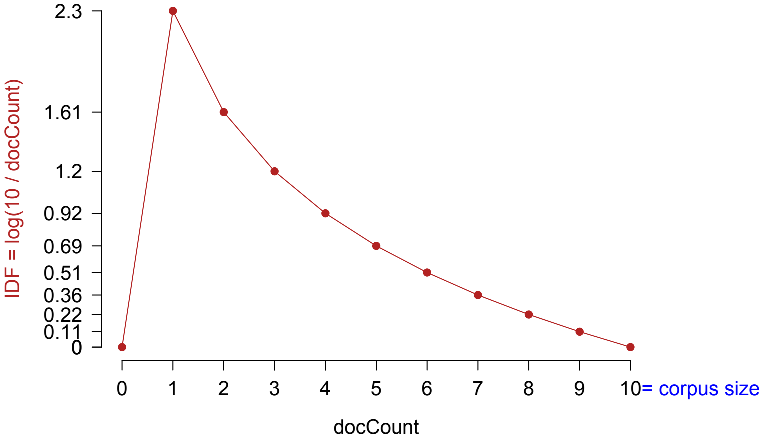
	IDF
A	0.00
B	0.29
C	0.69
D	1.39

$\Downarrow$

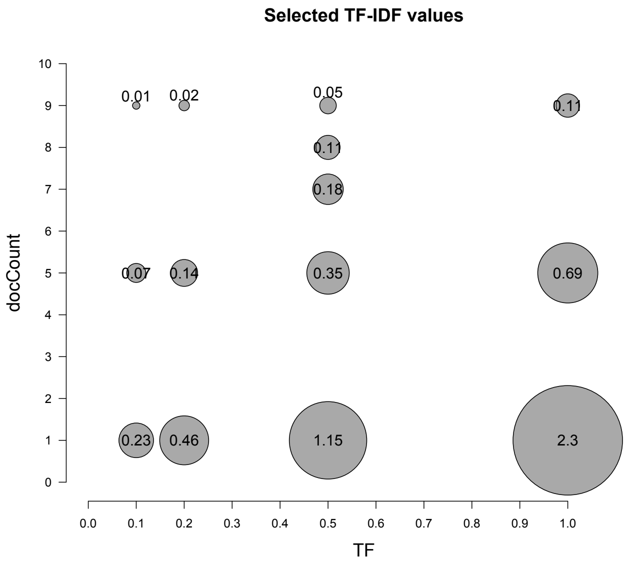
	TF			
	d_1	d_2	d_3	d_4
A	0.33	0.33	0.50	0.91
B	0.33	0.33	0.50	0.00
C	0.33	0.33	0.00	0.00
D	0.00	0.00	0.00	0.09

	TF-IDF			
	d_1	d_2	d_3	d_4
A	0.00	0.00	0.00	0.00
B	0.10	0.10	0.14	0.00
C	0.23	0.23	0.00	0.00
D	0.00	0.00	0.00	0.13

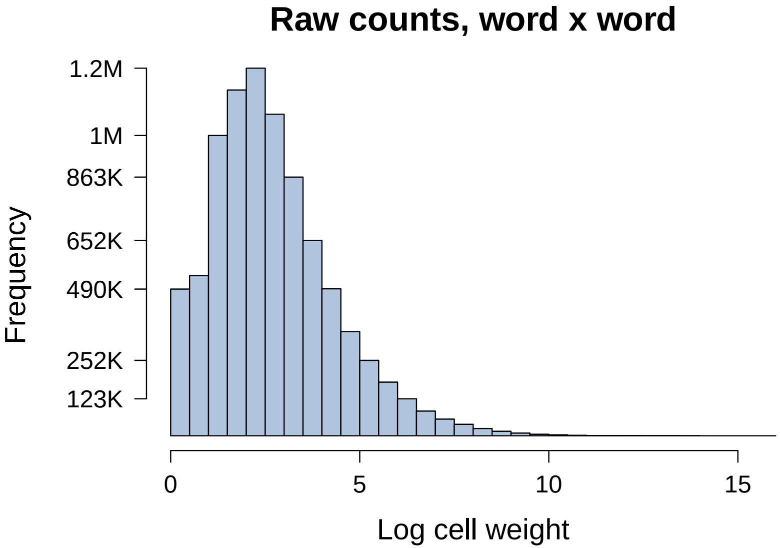
Term Frequency–Inverse Document Frequency (TF-IDF)



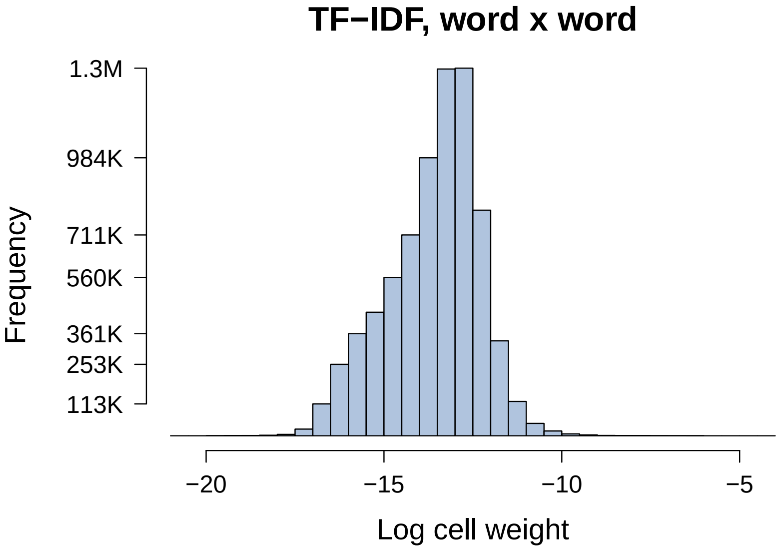
Term Frequency–Inverse Document Frequency (TF-IDF)



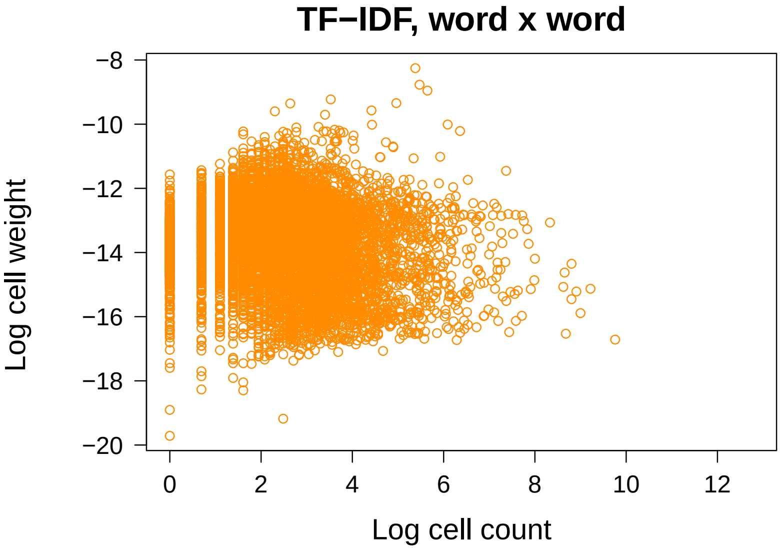
TF-IDF compared to counts



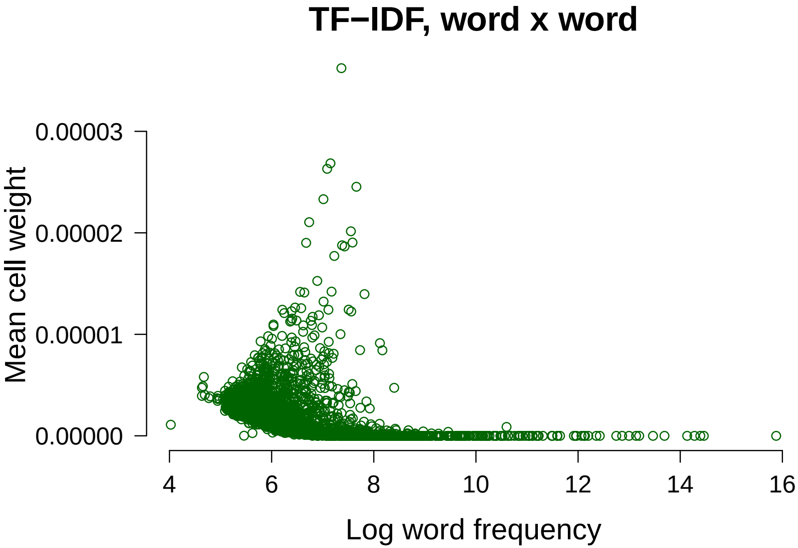
TF-IDF compared to counts



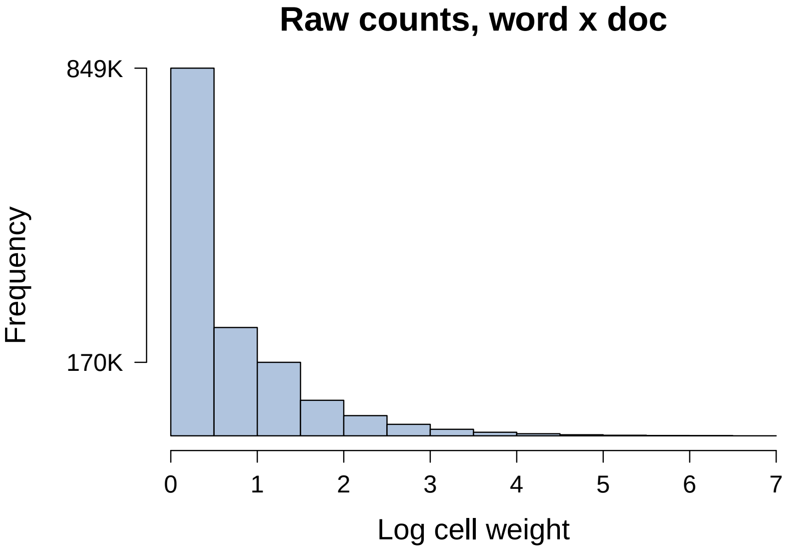
TF-IDF compared to counts



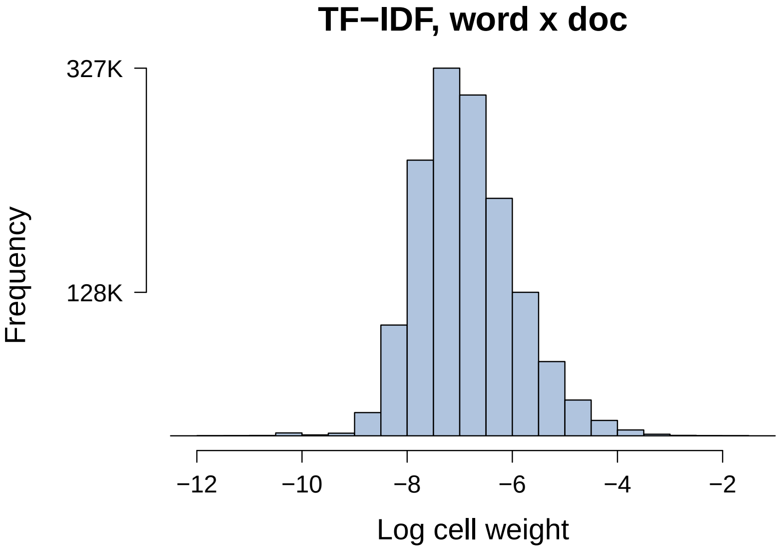
TF-IDF compared to counts



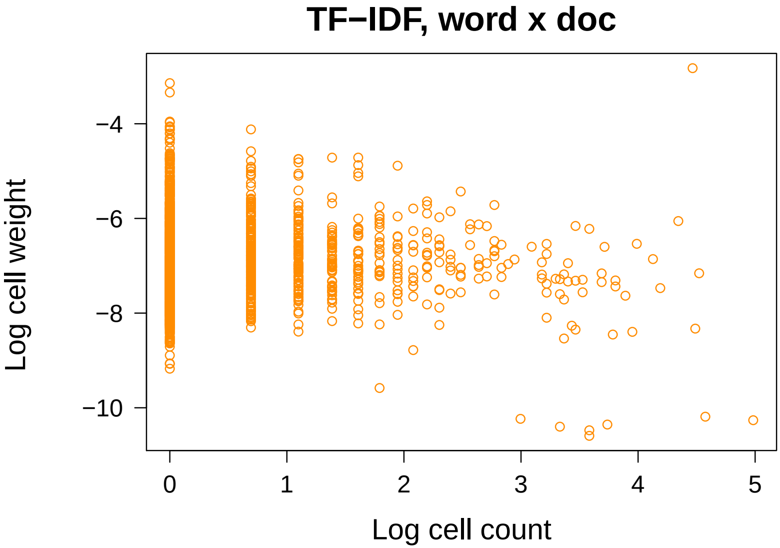
TF-IDF compared to counts



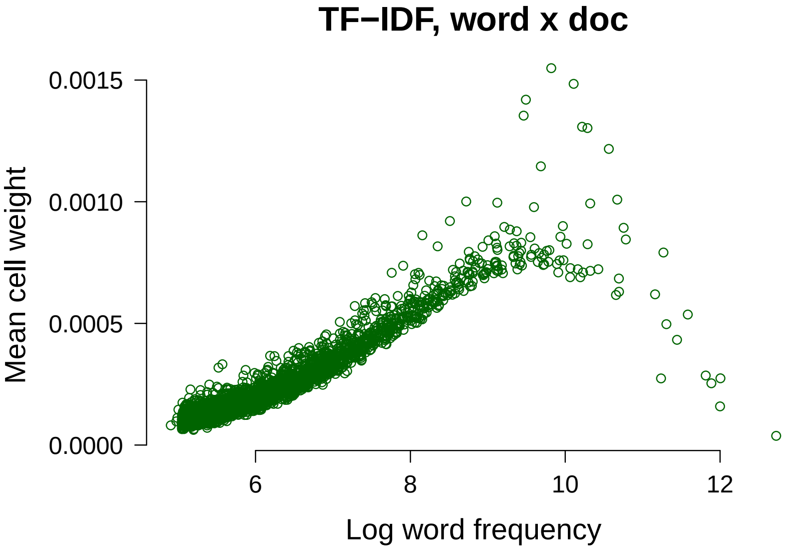
TF-IDF compared to counts



TF-IDF compared to counts



TF-IDF compared to counts



Pointwise Mutual Information (PMI)

Definition (PMI)

$$\log\left(\frac{P(w,d)}{P(w)P(d)}\right) \qquad \text{(assume } \log(0) = 0\text{)}$$

	d_1	d_2	d_3	d_4
A	10	10	10	10
B	10	10	10	0
C	10	10	0	0
D	0	0	0	1

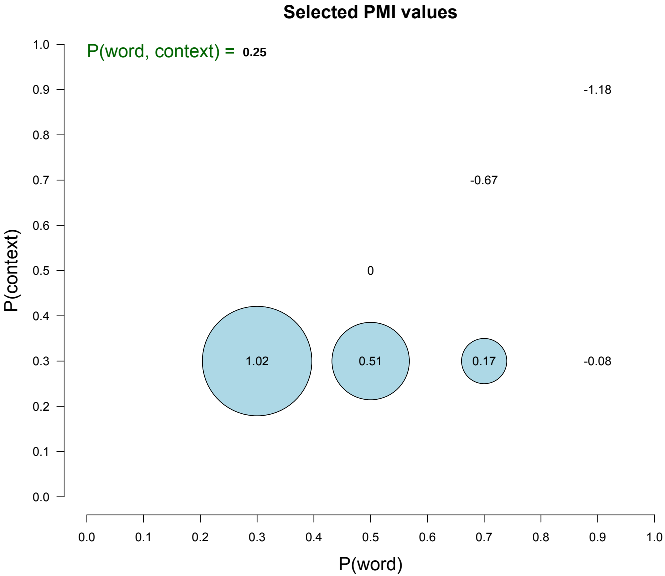
$\Rightarrow$

	$P(w,d)$				$P(w)$
A	0.11	0.11	0.11	0.11	0.44
B	0.11	0.11	0.11	0.00	0.33
C	0.11	0.11	0.00	0.00	0.22
D	0.00	0.00	0.00	0.01	0.01
$P(d)$	0.33	0.33	0.22	0.12	

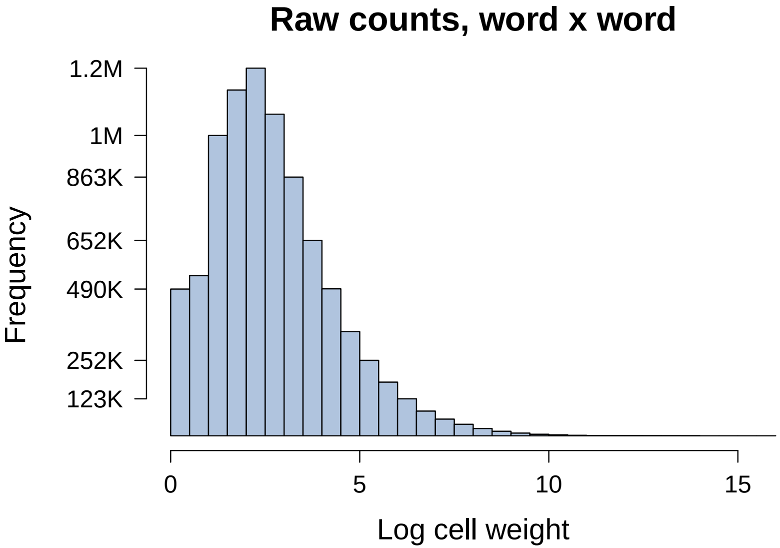
PMI
$\Downarrow$

	d_1	d_2	d_3	d_4
A	−0.28	−0.28	0.13	0.73
B	0.01	0.01	0.42	0.00
C	0.42	0.42	0.00	0.00
D	0.00	0.00	0.00	2.11

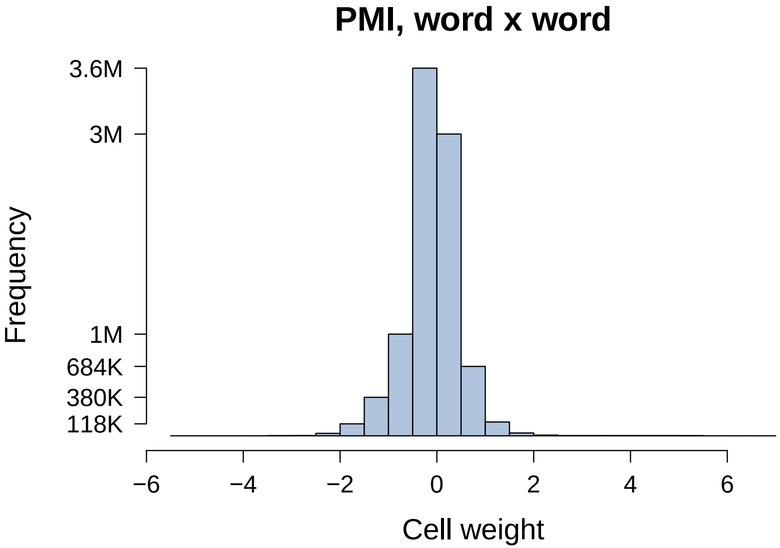
Pointwise Mutual Information (PMI)



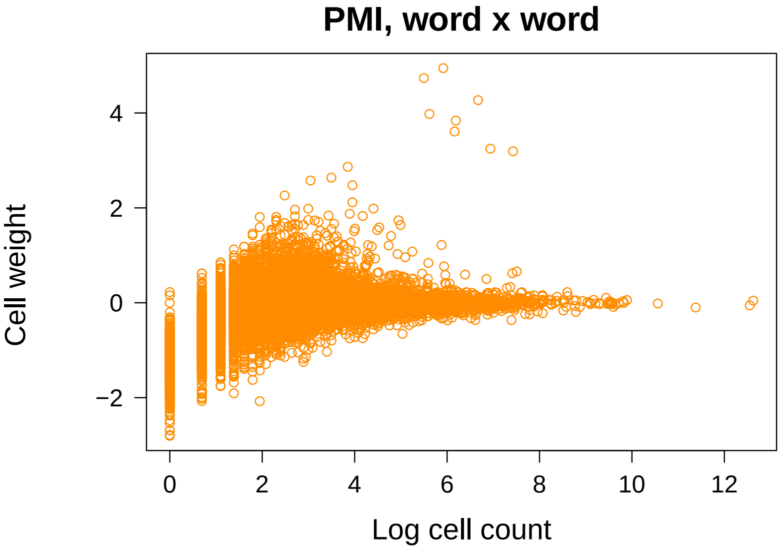
PMI compared to counts



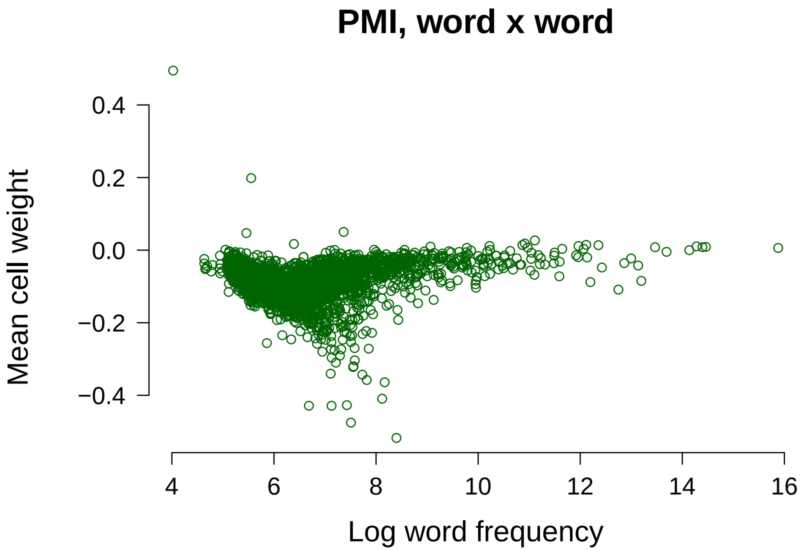
PMI compared to counts



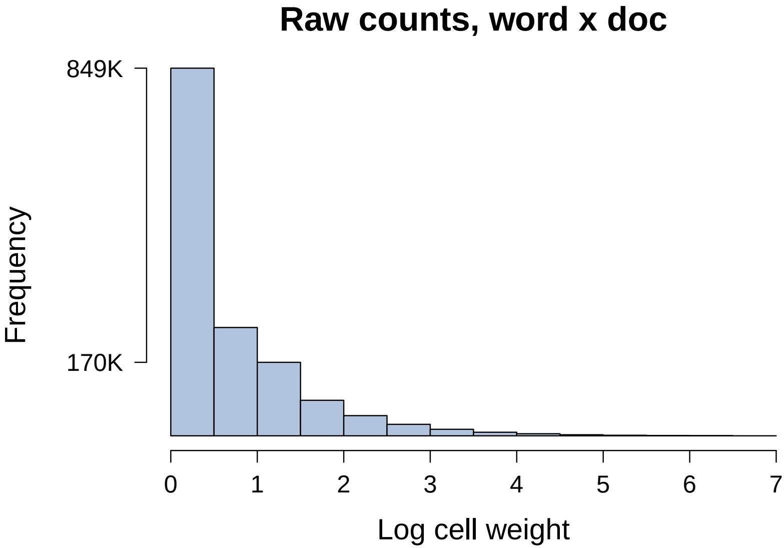
PMI compared to counts



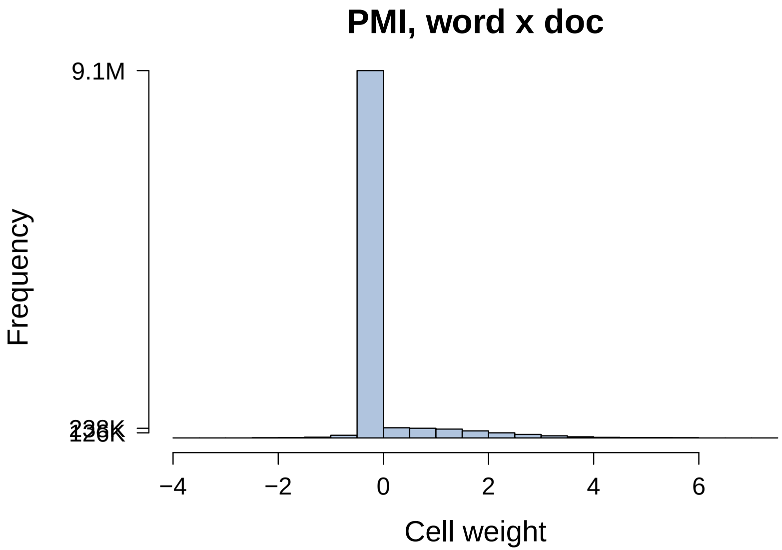
PMI compared to counts



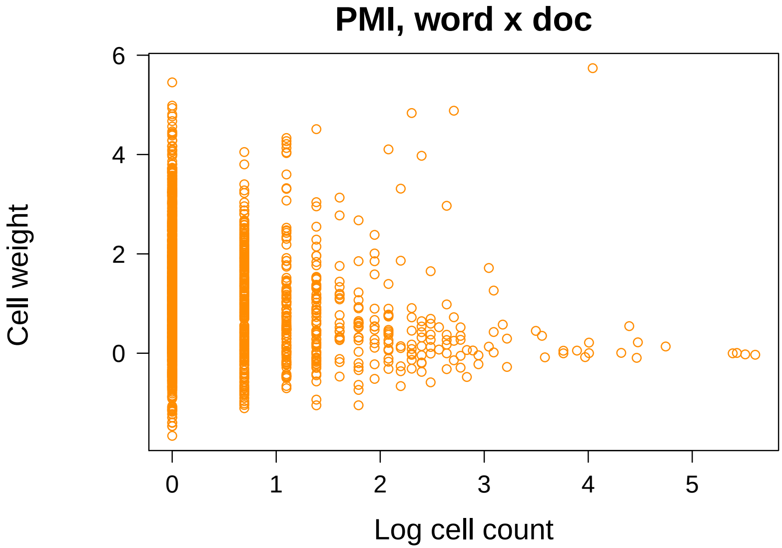
PMI compared to counts



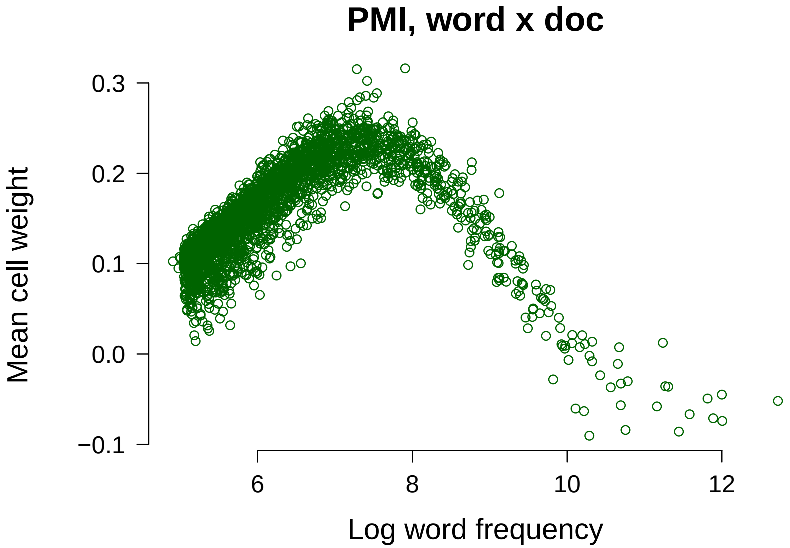
PMI compared to counts



PMI compared to counts



PMI compared to counts



PMI with Lapacian smoothing

Definition (Lapacian smoothing)

Add a constant amount to all the counts.

	d_1	d_2	d_3	d_4
A	10	10	10	10
B	10	10	10	0
C	10	10	0	0
D	0	0	0	1

PMI $\Rightarrow$

	d_1	d_2	d_3	d_4
A	−0.28	−0.28	0.13	0.73
B	0.01	0.01	0.42	0.00
C	0.42	0.42	0.00	0.00
D	0.00	0.00	0.00	2.11

$\Downarrow +4$

	d_1	d_2	d_3	d_4
A	14	14	14	14
B	14	14	14	4
C	14	14	4	4
D	4	4	4	5

PMI $\Rightarrow$

	d_1	d_2	d_3	d_4
A	−0.17	−0.17	−0.17	−0.17
B	0.03	0.03	0.03	−1.23
C	0.52	0.52	−0.74	−0.74
D	0.30	0.30	0.30	0.52

PMI with contextual discounting

Definition (Contextual rescaling)

For a matrix with *m* rows and *n* columns:

$$\text{newpmi}_{ij} = \text{pmi}_{ij} \times \frac{f_{ij}}{f_{ij}+1} \times \frac{\min(\sum_{k=1}^{m} f_{kj}, \sum_{k=1}^{n} f_{ik})}{\min(\sum_{k=1}^{m} f_{kj}, \sum_{k=1}^{n} f_{ik})+1}$$

Count matrix

	d_1	d_2	d_3	d_4
A	10	10	10	10
B	10	10	10	0
C	10	10	0	0
D	0	0	0	1

PMI

	d_1	d_2	d_3	d_4
A	−0.28	−0.28	0.13	0.73
B	0.01	0.01	0.42	0.00
C	0.42	0.42	0.00	0.00
D	0.00	0.00	0.00	2.11

$f_{ij}/(f_{ij}+1)$

	d_1	d_2	d_3	d_4
A	0.91	0.91	0.91	0.91
B	0.91	0.91	0.91	0.00
C	0.91	0.91	0.00	0.00
D	0.00	0.00	0.00	0.50

$\frac{\min(\sum_{k=1}^{m} f_{kj}, \sum_{k=1}^{n} f_{ik})}{\min(\sum_{k=1}^{m} f_{kj}, \sum_{k=1}^{n} f_{ik})+1}$

	d_1	d_2	d_3	d_4	Sum
A	$\frac{30}{30+1}$	$\frac{30}{30+1}$	$\frac{20}{20+1}$	$\frac{11}{11+1}$	40
B	$\frac{30}{30+1}$	$\frac{30}{30+1}$	$\frac{20}{20+1}$	$\frac{11}{11+1}$	30
C	$\frac{30}{30+1}$	$\frac{30}{30+1}$	$\frac{20}{20+1}$	$\frac{11}{11+1}$	20
D	$\frac{1}{1+1}$	$\frac{1}{1+1}$	$\frac{1}{1+1}$	$\frac{1}{1+1}$	1
Sum	30	30	20	11	

Discounted PMI

	d_1	d_2	d_3	d_4
A	−0.24	−0.24	0.11	0.61
B	0.01	0.01	0.36	0.00
C	0.36	0.36	0.00	0.00
D	0.00	0.00	0.00	0.53

PMI with contextual discounting

Definition (Contextual rescaling)

For a matrix with *m* rows and *n* columns:

$$\text{newpmi}_{ij} = \text{pmi}_{ij} \times \frac{f_{ij}}{f_{ij}+1} \times \frac{\min(\sum_{k=1}^{m} f_{kj}, \sum_{k=1}^{n} f_{ik})}{\min(\sum_{k=1}^{m} f_{kj}, \sum_{k=1}^{n} f_{ik})+1}$$

Count matrix

	d_1	d_2	d_3	d_4
A	10	10	10	10
B	10	10	10	0
C	10	10	0	0
D	0	0	0	1

PMI

	d_1	d_2	d_3	d_4
A	−0.28	−0.28	0.13	0.73
B	0.01	0.01	0.42	0.00
C	0.42	0.42	0.00	0.00
D	0.00	0.00	0.00	2.11

$f_{ij}/(f_{ij}+1)$

	d_1	d_2	d_3	d_4
A	0.91	0.91	0.91	0.91
B	0.91	0.91	0.91	0.00
C	0.91	0.91	0.00	0.00
D	0.00	0.00	0.00	0.50

$\frac{\min(\sum_{k=1}^{m} f_{kj}, \sum_{k=1}^{n} f_{ik})}{\min(\sum_{k=1}^{m} f_{kj}, \sum_{k=1}^{n} f_{ik})+1}$

	d_1	d_2	d_3	d_4	Sum
A	0.97	0.97	0.95	0.92	40
B	0.97	0.97	0.95	0.92	30
C	0.95	0.95	0.95	0.92	20
D	0.50	0.50	0.50	0.50	1
Sum	30	30	20	11	

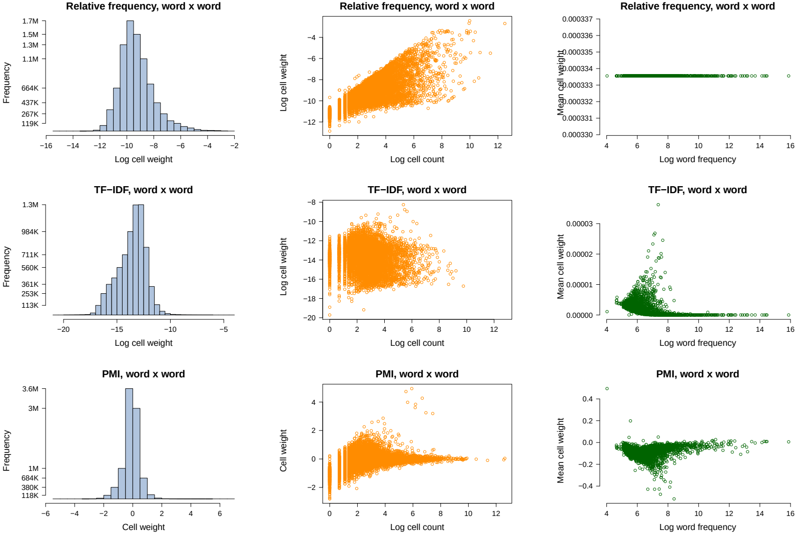
Discounted PMI

	d_1	d_2	d_3	d_4
A	−0.24	−0.24	0.11	0.61
B	0.01	0.01	0.36	0.00
C	0.36	0.36	0.00	0.00
D	0.00	0.00	0.00	0.53

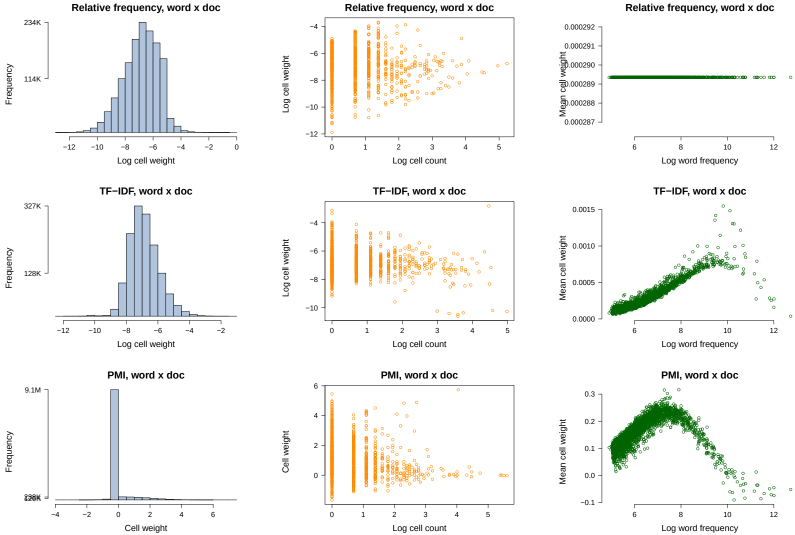
Other weighting/normalization schemes

- Expected values: $\text{expected}_{ij} = \sum_r \text{observed}_{ir} \times \left(\frac{\sum_k \text{observed}_{kj}}{\sum_{kr} \text{observed}_{kr}} \right)$
- t-test: $\frac{p(w,d) - p(w)p(d)}{\sqrt{p(w)p(d)}}$
- TF-IDF variants that seek to be sensitive to the empirical distribution of words (Church and Gale 1995; Manning and Schütze 1999:553; Baayen 2001)

Summary comparisons



Summary comparisons



Relationships and generalizations

- Many weighting schemes end up favoring rare events that may not be trustworthy. Discounting procedures seek to combat this.
- The magnitude of counts can be important; $[1, 10]$ and $[1000, 10000]$ might represent very different situations; creating probability distributions or length normalizing will obscure this.
- TF-IDF severely punishes words that appear in many documents — it behaves oddly for dense matrices, which can include word $\times$ word matrices

Using our code

```
# Reweight the imdb matrix using PPMI with discounting:
p = pmi(m, positive=True, discounting=True)
# Reweight the imdb matrix using TF-IDF:
t = tfidf(m)
# Use neighbors on 'outstanding' and 'good'
neighbors(p, 'outstanding')
neighbors(t, 'good', distfunc=euclidean_distance)
...
```

Exploration

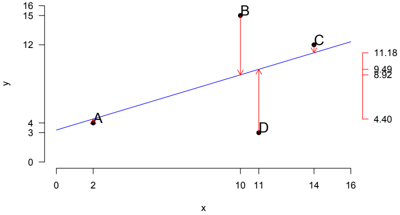
1. Which combination of weighting and distance seems to give the best results?
2. Do you notice systematic differences that we can understand in terms of the underlying properties of the matrix and/or the distance measure?
3. Load the word x document version of the imdb matrix:
   `d = Matrix('imdb-worddoc.csv')`
   and see how it interacts with the weighting and distance concepts.

Dimensionality reduction

- The goal of dimensionality reduction is eliminate rows/columns that are highly correlated while bringing similar things together and pushing dissimilar things apart.
- This section looks briefly at Latent Semantic Analysis (Deerwester et al. 1990), which seeks not only to find a reduced-sized matrix but also to capture similarities that come not just from direct co-occurrence, but also from second-order co-occurrence.
- Latent Semantic Analysis is an application of truncated singular value decomposition (SVD). SVD is a central matrix operation; 'truncation' here means looking only at submatrices of the full decomposition.
- For more:
  - Turney and Pantel 2010:§4.3
  - Manning and Schütze 1999:§15.4
  - Manning et al. 2009:§18

Latent Semantic Analysis (truncated singular value decomposition)

- For the 2d case, SVD is closely related to fitting a least-squares regression, where the idea is to find a line that minimizes the errors (equivalently, whose vector of errors is orthogonal to the fitted line):



- The least-squares regression reduces the matrix to a line.
- Trunctated SVD, as applied in LSA, is the process of reducing a rectangular $m \times n$ matrix to an $i \times n$ matrix where $i \ll m$ or a $m \times j$ matrix where $j \ll n$.
- In the reduced dimension matrices, once-correlated variables are orthogonal and only the dimensions of greatest variation remain.

Example: toy dialect difference (*gnarly* for LA; *wicked* for Boston)

	d1	d2	d3	d4	d5	d6
gnarly	1	0	1	0	0	0
wicked	0	1	0	1	0	0
awesome	1	1	1	1	0	0
lame	0	0	0	0	1	1
terrible	0	0	0	0	0	1

$\Downarrow\Uparrow$

Distance from *gnarly*
1. gnarly
2. awesome
3. terrible
4. wicked
5. lame

Example: toy dialect difference (*gnarly* for LA; *wicked* for Boston)

	d1	d2	d3	d4	d5	d6
gnarly	1	0	1	0	0	0
wicked	0	1	0	1	0	0
awesome	1	1	1	1	0	0
lame	0	0	0	0	1	1
terrible	0	0	0	0	0	1

$\Downarrow\Uparrow$

Distance from *gnarly*
1. gnarly
2. awesome
3. terrible
4. wicked
5. lame

T(erm)

gnarly	0.41	0.00	0.71	0.00	-0.58
wicked	0.41	0.00	-0.71	0.00	-0.58
awesome	0.82	-0.00	-0.00	-0.00	0.58
lame	0.00	0.85	0.00	-0.53	0.00
terrible	0.00	0.53	0.00	0.85	0.00

×

S(ingular values)

1	2.45	0.00	0.00	0.00	0.00
2	0.00	1.62	0.00	0.00	0.00
3	0.00	0.00	1.41	0.00	0.00
4	0.00	0.00	0.00	0.62	0.00
5	0.00	0.00	0.00	0.00	-0.00

×

D(ocument) ^{T}

d1	0.50	-0.00	0.50	0.00	-0.71
d2	0.50	0.00	-0.50	0.00	0.00
d3	0.50	-0.00	0.50	0.00	0.71
d4	0.50	-0.00	-0.50	-0.00	0.00
d5	-0.00	0.53	0.00	-0.85	0.00
d6	0.00	0.85	0.00	0.53	0.00

Example: toy dialect difference (*gnarly* for LA; *wicked* for Boston)

	d1	d2	d3	d4	d5	d6
gnarly	1	0	1	0	0	0
wicked	0	1	0	1	0	0
awesome	1	1	1	1	0	0
lame	0	0	0	0	1	1
terrible	0	0	0	0	0	1

⇓⇑

Distance from *gnarly*
1. gnarly
2. awesome
3. terrible
4. wicked
5. lame

T(erm)

gnarly	0.41	0.00	0.71	0.00	-0.58
wicked	0.41	0.00	-0.71	0.00	-0.58
awesome	0.82	-0.00	-0.00	-0.00	0.58
lame	0.00	0.85	0.00	-0.53	0.00
terrible	0.00	0.53	0.00	0.85	0.00

×

S(ingular values)

1	2.45	0.00	0.00	0.00	0.00
2	0.00	1.62	0.00	0.00	0.00
3	0.00	0.00	1.41	0.00	0.00
4	0.00	0.00	0.00	0.62	0.00
5	0.00	0.00	0.00	0.00	-0.00

×

D(ocument)^T

d1	0.50	-0.00	0.50	0.00	-0.71
d2	0.50	0.00	-0.50	0.00	0.00
d3	0.50	-0.00	0.50	0.00	0.71
d4	0.50	-0.00	-0.50	-0.00	0.00
d5	-0.00	0.53	0.00	-0.85	0.00
d6	0.00	0.85	0.00	0.53	0.00

gnarly	0.41	0.00
wicked	0.41	0.00
awesome	0.82	-0.00
lame	0.00	0.85
terrible	0.00	0.53

×

2.45	0.00
0.00	1.62

=

gnarly	1.00	0.00
wicked	1.00	0.00
awesome	2.00	0.00
lame	0.00	1.38
terrible	0.00	0.85

Distance from *gnarly*
1. gnarly
2. wicked
3. awesome
4. terrible
5. lame

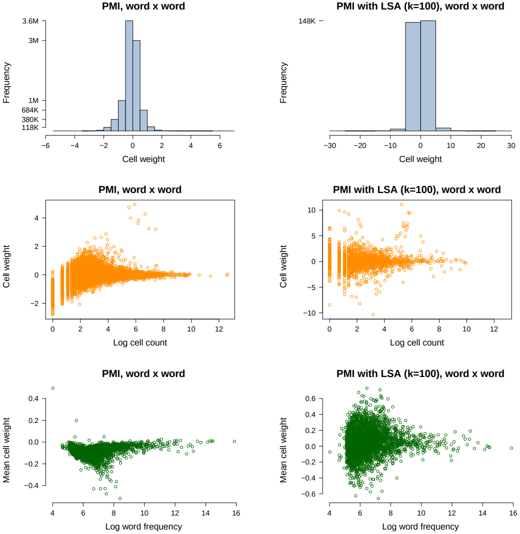
Using our code

```
# Run LSA on the PMI imdb matrix:
s = lsa(p, k=100)
# Use neighbors to see what's happening:
df = neighbors(s, 'happy')
df = neighbors(s, 'happy')
...
```

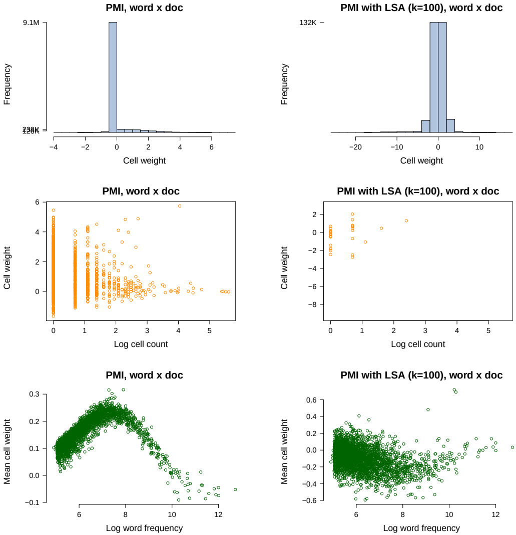
Exploration

1. How do the results compare with what you saw for these matrices before reduction?
2. What happens if you set `k=1` using `lsa`. What do the results look like then? What do you think this first (and now only) dimension is capturing?

Comparisons before and after LSA with k=100



Comparisons before and after LSA with k=100



Other dimensionality reduction techniques

- Probabilistic LSA (PLSA; Hofmann 1999)
- Latent Dirichlet Allocation (LDA; Blei et al. 2003; Steyvers and Griffiths 2007)
- t-Distributed Stochastic Neighbor Embedding (t-SNE; van der Maaten and Geoffrey 2008)
- For even more: Turney and Pantel 2010:160

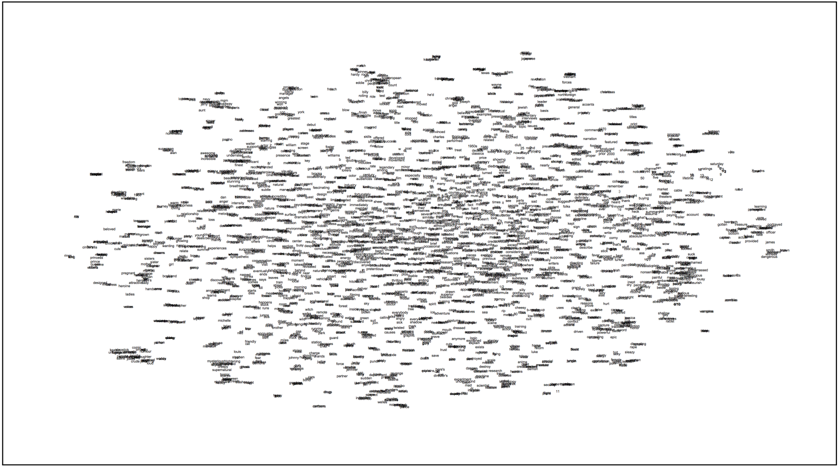
Tools for VSMs

- See Turney and Pantel 2010:§5 for lots of open-source projects
- Python NLTK's `text` and `cluster`: http://www.nltk.org/
- Python's `gensim package`: http://radimrehurek.com/gensim/
- R's `topicmodels` package (mostly for LDA)

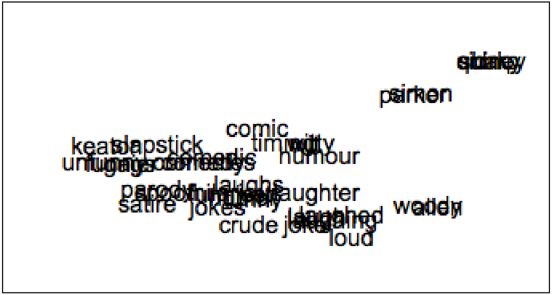
Tools for visualization

- t-SNE implementations for dimensionality reduction and 2d visualization: `http://homepage.tudelft.nl/19j49/t-SNE.html`
- Multiple maps t-SNE (van der Maaten and Hinton 2012)
- Gephi: `http://gephi.org/`

Visualization with t-SNE



Visualization with t-SNE



Visualization with t-SNE

Visualization with t-SNE

Looking ahead in the course

- VSMs and (semi-)supervised training (next meeting)
- VSMs and the goals of semantics (next meeting)
- VSMs and semantic composition (May 14)
- VSMs and sentiment analysis (May 14, 19)
- VSMs and relation extraction (see Turney and Pantel 2010:§2.3-2.4, §5.3)

References I

Baayen, R. Harald. 2001. *Word Frequency Distributions*. Dordrecht: Kluwer Academic Publishers.

Blei, David M.; Andrew Y. Ng; and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Bullinaria, John A. and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods* 39(3):510–526.

Church, Kenneth Ward and William Gale. 1995. Inverse dcument frequency (IDF): A measure of deviations from Poisson. In David Yarowsky and Kenneth Church, eds., *Proceedings of the Third ACL Workshop on Very Large Corpora*, 121–130. The Association for Computational Linguistics.

Deerwester, S.; S. T. Dumais; G. W. Furnas; T. K. Landauer; and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41(6):391–407. doi:\bibinfo{doi}{10.1002/(SICI)1097-4571(199009)41:6⟨391::AID-ASI1⟩3.0.CO;2-9}.

Dice, Lee R. 1945. Measures of the amount of ecologic association between species. *Ecology* 26(3):267–302.

Fellbaum, Christiane, ed. 1998. *WordNet: An Electronic Database*. Cambridge, MA: MIT Press.

Fillmore, Charles J. and Collin F. Baker. 2001. Frame semantics for text understanding. In *Proceedings of the WordNet and Other Lexical Resources Workshop*, 59–64. Pittsburgh, PA: Association for Computational Linguistics.

Hofmann, Thomas. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 50–57. New York: ACM. doi:\bibinfo{doi}{http://doi.acm.org/10.1145/312624.312649}. URL http://doi.acm.org/10.1145/312624.312649.

Lee, Lillian. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 25–32. College Park, Maryland, USA: ACL. doi:\bibinfo{doi}{10.3115/1034678.1034693}.

van der Maaten, Laurens and Hinton Geoffrey. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9:2579–2605.

van der Maaten, Laurens and Geoffrey E. Hinton. 2012. Visualizing non-metric similarities in multiple maps. *Machine Learning* 87(1):33–55.

Manning, Christopher D.; Prabhakar Raghavan; and Hinrich Schütze. 2009. *An Introduction to Information Retrieval*. Cambridge University Press.

Manning, Christopher D. and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Miller, George A. 1995. WordNet: A lexical database for English. *Communications of the ACM* 38(11):39–41.

van Rijsbergen, Cornelis Joost. 1979. *Information Retrieval*. London: Buttersworth.

Steyvers, Mark and Tom Griffiths. 2007. Probabilistic topic models. In Thomas K. Landauer; Danielle S. McNamara; Simon Dennis; and Walter Kintsch, eds., *Handbook of Latent Semantic Analysis*, 427–448. Hillsdale, NJ: Lawrence Erlbaum Associates.

Stolcke, Andreas; Klaus Ries; Noah Coccaro; Elizabeth Shriberg; Rebecca Bates; Daniel Jurafsky; Paul Taylor; Rachel Martin; Marie Meteer; and Carol Van Ess-Dykema. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics* 26(3):339–371.

Stone, Philip J; Dexter C Dunphry; Marshall S Smith; and Daniel M Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. Cambridge, MA: MIT Press.

Turney, Peter D. and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)* 21:315–346. doi:\bibinfo{doi}{http://doi.acm.org/10.1145/944012.944013}.

Turney, Peter D. and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* 37:141–188.