Text mining

3. Векторная модель

Дмитрий Ильвовский, Екатерина Черняк

dilvovsky@hse.ru, echernyak@hse.ru

Национальный Исследовательский Университет – Высшая Школа Экономики
НУЛ Интеллектуальных систем и структурного анализа

February 2, 2017

Векторная модель

Векторная модель (Vector Space Model, VSM)

Алгебраическая модель представления документов (или любых других объектов) векторами слов.

- Векторная модель коллекции документов: документ – это вектор, состоящий из частот слов, в нем встречающихся
  - Информационный поиск
- Векторная модель слов (иначе дистрибутивная семантика (distributional semantics)): слово – это вектор, состоящий из частот слов, встречающихся в его окрестности
  - Математическая модель семантики интересна и сама по себе, и как вспомогательное средство

Что такое информационный поиск?

Основные понятия

- **Документ** – любой контент, чаще всего – текст
- **Запрос** – небольшой текст, написанный пользователем для выражения его информационных нужд
- **Релевантность** – некая функция, показывающая, насколько данный документ соответствует запросу (то есть, насколько документ удовлетворяет информационные нужды пользователя)

Основные задачи

- **Индексация** – обработка и хранение текстов в хранилище, создание индекса
- **Поиск** – поиск в хранилище текста, наиболее релевантного запросу

Релевантность документов запросу

Релевантность – это сложно:

- Зависит от намерений пользователя
- Зависит от места, времени, используемого устройства
- Зависит от других документов, найденных по этому запросу

Очень большое упрощение:

- D – множество документов
- Q – множество запросов
- Теоретико-множественный подход: $R : D \times Q \to \{0, 1\}$ – бинарная функция релевантности
- Алгебраический подход: $R : D \times Q \to [0, 1]$ – задает ранжирование документов
- Вероятностный подход: $P(R|d, q)$ – когда-нибудь в следующий раз :)

Формальная модель

Чтобы построить модель **представления** текста, нужно определить из чего текст состоит: например, из слов или их производных (термов). Термами могут быть исходные слова, леммы, стемы, буквенные n-граммы и т.д..

на	дворе	трава	траве	дрова	не	руби	двора
3	1	1	2	2	1	1	1
на	двор	трава	дрова	не	рубить		
3	2	3	2	1	1		
на	двор	трав	дров	не	руб		
3	2	3	2	1	1		

Обозначения:

- T – множество термов, всего термов – $|T| = N$
- $d = (w_1, w_2, ..., w_N)$ – **векторное представление текста**
- Вес

$$w_i = \begin{cases} 0, & \text{если } t_i \notin d \\ > 0, & \text{иначе} \end{cases}$$

Индекс

- **Прямой индекс** – список слов (и их позиций) в документе
- **Инвертированный индекс** – список документов (и позиций в них), в (на) которых встречается слово

	d_1	d_2	...	d_M
t_1	w_{11}	w_{12}	...	w_{1M}
t_2	w_{21}	w_{22}		...
...	...			
t_N	w_{N1}	w_{N2}		w_{NM}

Поиск по запросу

Считаем нерелевантными документы, не содержащие ни одного терма из запроса.

Пусть $q = t_i$ – запрос состоит из одного терма.

- Найти все документы d_j, такие что $w_{ij} > 0$
- Отсортировать их по убыванию w_{ij}
- Вернуть пользователю отсортированный список “релевантных” запросу документов

Пусть $q = t_1, \ldots t_k$ – запрос состоит из нескольких термов.

- **Дизъюнктивный поиск**: вернуть список документов, содержащих хотя бы один терм из запроса
- **Коньюнктивный поиск**: вернуть список документов, содержащих все термы из запроса

Булева модель поиск

Простая модель, основанная на булевой алгебре. Каждый запрос – это выражение на языке булевой алгебры с тремя операторами AND, OR, NOT. Пример: $q =$ “java” AND “компилятор” AND (“unix” OR “linux”).
Релевантность в булевой модели поиска: документ релевантен запросу, если он удовлетворяет соответствующему булевому выражению.

	d_1	d_2	d_3	d_4	d_5
a	**1**	**1**	**1**	**0**	**1**
b	**0**	**1**	**0**	**1**	**1**
c	**0**	**0**	**1**	**0**	**1**

$q = a$ AND (b OR (NOT c))

Использование инвертированного индекса индекса для булевой модели поиска

$q = a$ AND (b OR (NOT c))
$a \to d_1, d_2, d_3, d_5$
$b \to d_2, d_4, d_5$
$c \to d_3, d_5$

	d_1	d_2	d_3	d_4	d_5
a	**1**	**1**	1	0	**1**
b	**0**	**1**	0	1	**1**
c	**0**	**0**	1	0	**1**
NOT c	1	1	0	1	0
b OR (NOT c)	1	1	0	1	1
a AND (b OR (NOT c))	1	1	0	0	1

Результат: d_1, d_2, d_5

Булева модель поиска: плюсы и минусы

Плюсы

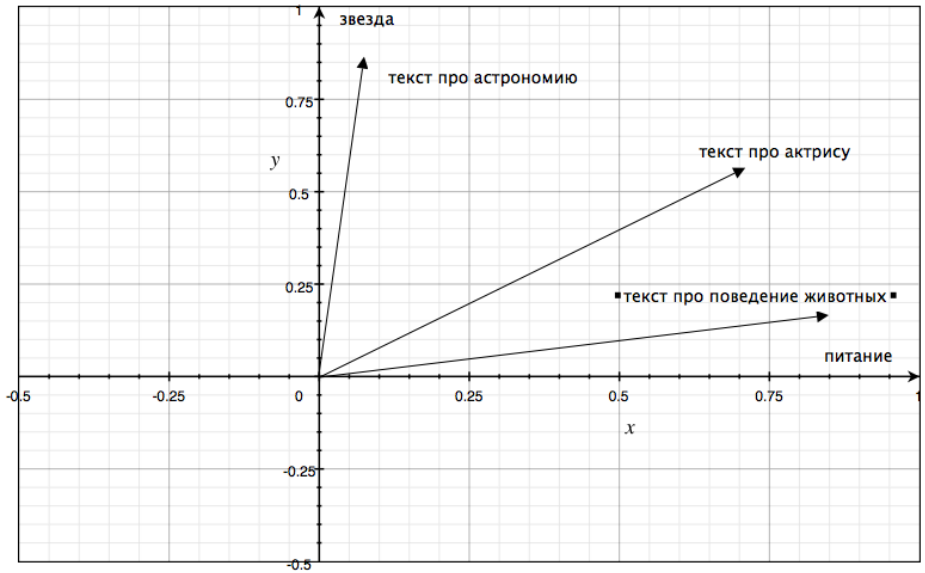
- Быстрый поиск: бинарные операции выполняются быстро
- Простой поиск: простое бинарное решение, документ либо релевантен, либо нет
- Легко добавить учет синонимов, учет времени и места запроса

Минусы

- Нет ранжирования документов по релевантности
- Настоящие пользователи поисковых систем не разговаривают на языке булевой алгебры

Векторная модель

- Каждый текст – это вектор в N-мерном пространстве термов (N – количество термов в проиндексированной коллекции текстов)
- Запрос – такой же вектор, как и любой текст
- Релевантность документа запросу определяется по сходству документа и запроса, иначе говоря, по близости соответсвующих им векторов друг другу

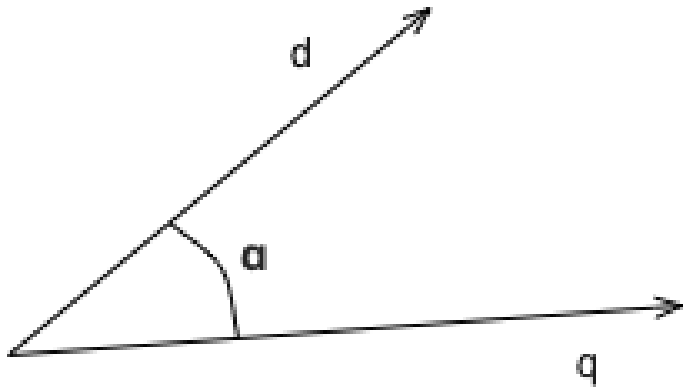


Векторная модель

$$d = (w_1^d, \ldots, w_N^d)$$

$$q = (w_1^q, \ldots, w_N^q)$$

$$\cos\alpha = \mathtt{sim}(q, d) = \frac{q \times d}{||q||||d||} = \frac{\sum_i w_i^q \times w_i^d}{\sqrt{(\sum_i w_i^q)^2}\sqrt{(\sum_i w_i^d)^2}}$$

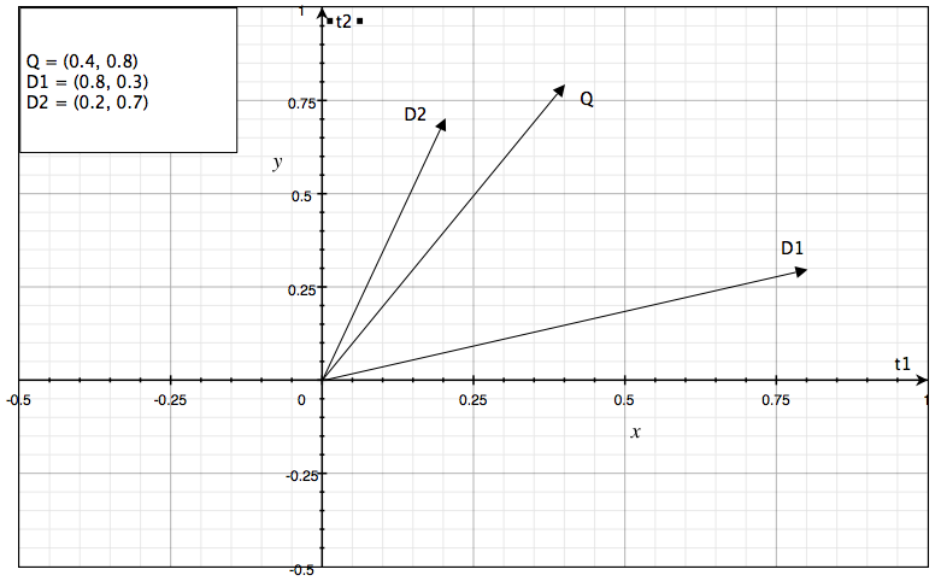


Векторная модель

$$\cos\alpha = \mathtt{sim}(q,d) = \frac{q \times d}{||q||||d||} = \frac{\sum_i w_i^q \times w_i^d}{\sqrt{(\sum_i w_i^q)^2}\sqrt{(\sum_i w_i^d)^2}}$$

$$\mathtt{sim}(q, d_2) = \frac{0.4 \times 0.2 + 0.8 \times 0.7}{\sqrt{0.4^2 + 0.8^2}\sqrt{0.2^2 + 0.7^2}} = 0.98$$

$$\mathtt{sim}(q, d_1) = \frac{0.4 \times 0.8 + 0.8 \times 0.3}{\sqrt{0.4^2 + 0.8^2}\sqrt{0.8^2 + 0.3^2}} = 0.74$$



Как определить веса w?

Схема взвешивания *tfidf*

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D) = f_{t,d} \times \log \frac{|D|}{n_t + 1},$$

где $|D|$ – количество документов, n_t – количество документов, содержащих терм t

Векторная модель: плюсы и минусы

Плюсы:

- Использование весов позволяет улучшить поиск
- Возможен дизъюнктивный поиск
- Косинус – приятная и легко интерпретируемая с точки зрения математики функция
- Множество экспериментов показало, что косинусная функция – хорошая мера релевантности

Минусы

- Между термами бывают сильные семантические связи, поэтому термы никак не могут соответствовать ортогональным векторам
- *IDF* долго вычисляется

Что дальше? Необходим учет полисемии, синонимов и около-синонимов, аббревиатур, различных написаний одного и того же слова.

Снижение размерности в векторной модели

Введем обозначения

- M – количество термов
- $N = |D|$ – количество документов
- C – матрица терм-документ размера $M \times N$, $\mathtt{rank}(C) \leq \min M, N$

И вспомним кое-что из линейной алгебры:
Пусть A квадратная матрица размера $M \times M$. В уравнении $Ax = \lambda x$ – x – собственный вектор, λ – собственные значения, которые находятся с помощью характеристического уравнения $(A - \lambda I_M)x = 0$. Число собственных значений совпадает с рангом матрицы – количеством линейно независимых строк (или столбцов).

Теорема о сингулярном разложении матрицы

Пусть r – это ранг матрицы C размера $M \times N$. Сингулярным разложением матрицы C назовем уравнение

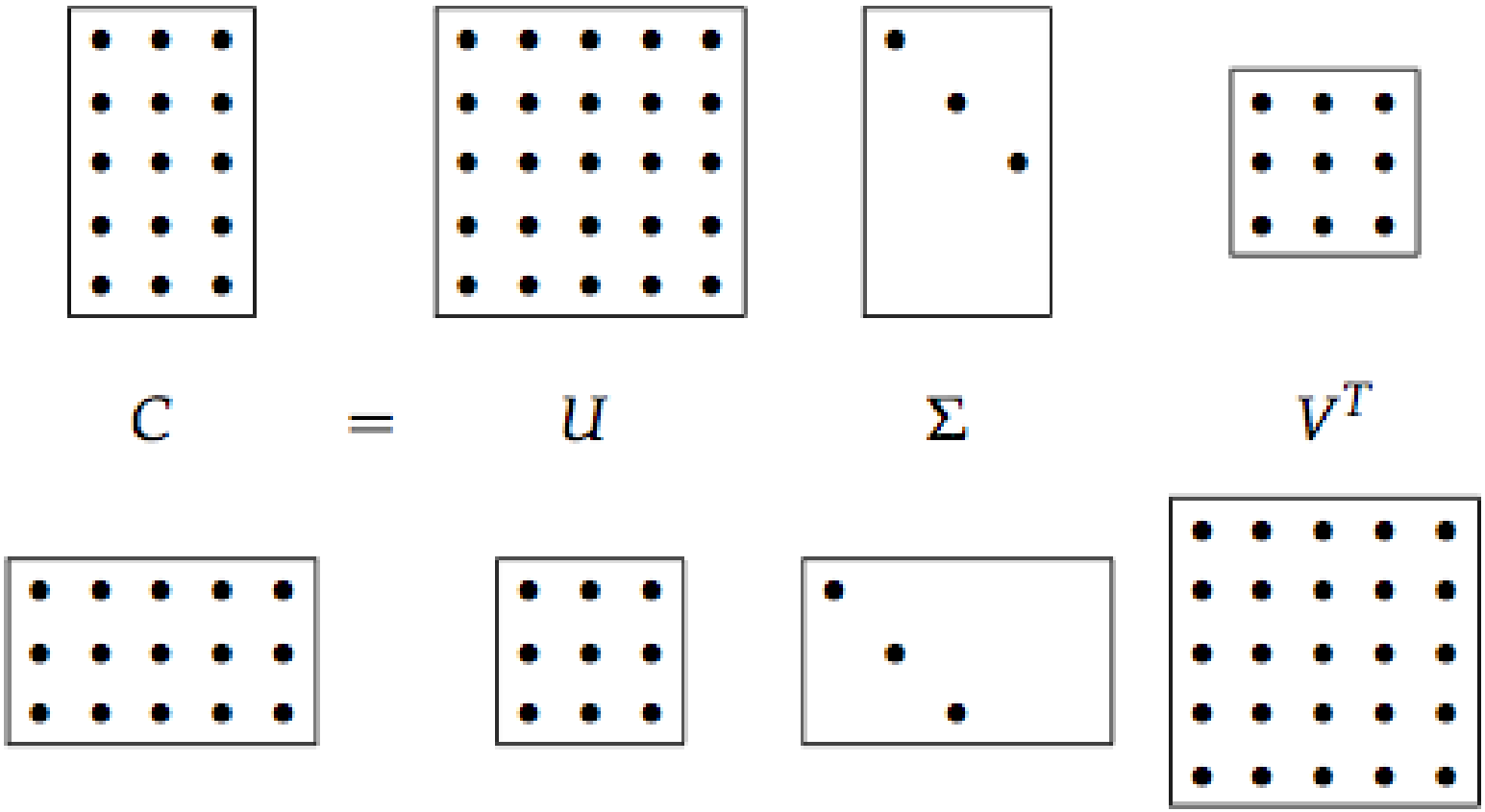
$$C = U\Sigma V^T,$$

где

- Собственные значения $\lambda_1, \ldots, \lambda_r$ матрицы CC^T совпадают с собственными значениями матрицы $C^T C$
- Пусть $\sigma_i = \sqrt{\lambda_i}$, $1 \leq i \leq r$, причем $\lambda_{i+1} \leq \lambda_i$. Тогда Σ – матрица размера $M \times N$, $\Sigma_{ii} = \sigma_i$, $1 \leq i \leq r$.

Σ – матрица сингулярных значений, U – матрица левых сингулярных векторов, V – матрица правых сингулярных векторов.

Сингулярное разложение матрицы

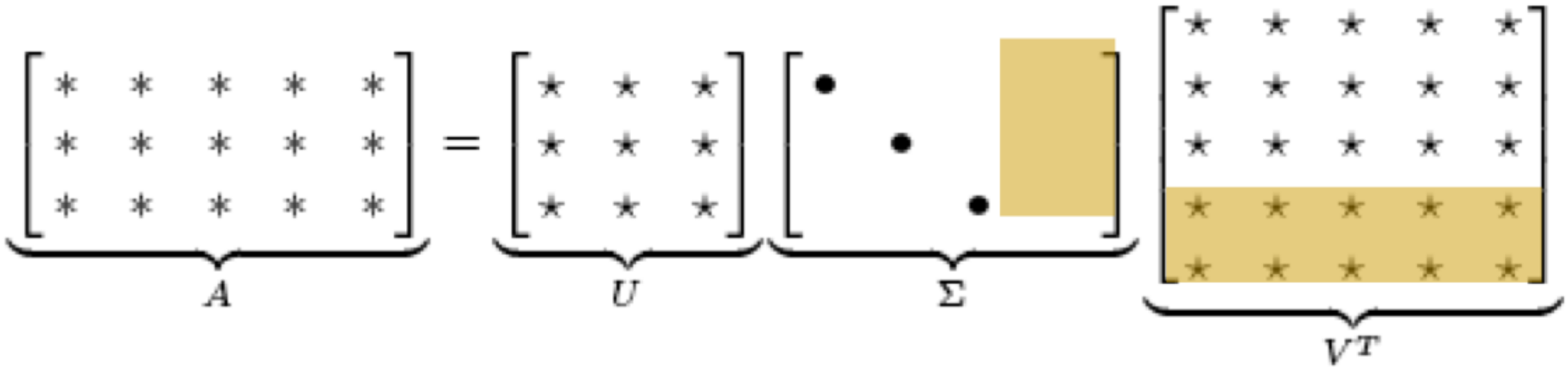


Сингулярное разложение матрицы

reduced SVD

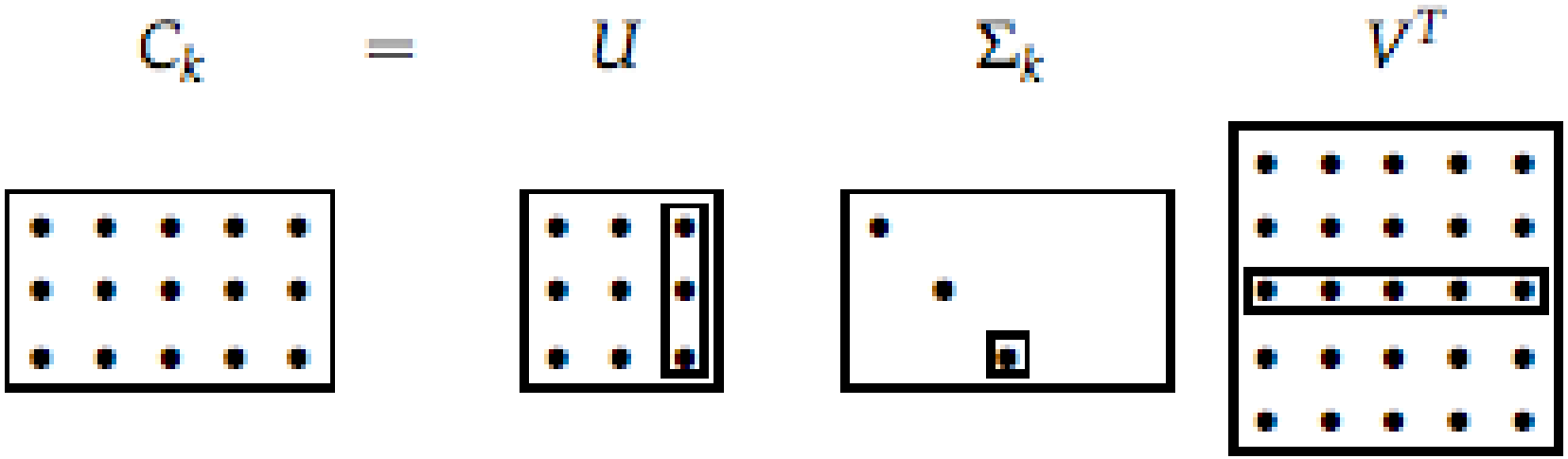
```
In[1]:  U, s, V = numpy.linalg.svd(C, full_matrices=False)
```

$C - M \times N, U - M \times M, s - M \times 1, V - M \times N$



Аппроксимация матрицы C матрицей меньшего ранга

1. Найти $C = U\Sigma V^T$
2. В матрице Σ обнулить $r - k$ наименьших значений и получить Σ_k
3. Искомая апроксимация: $C_k = U\Sigma_k V^T = \sum_{i=1}^{k} \sigma_i u_i v_i^T$

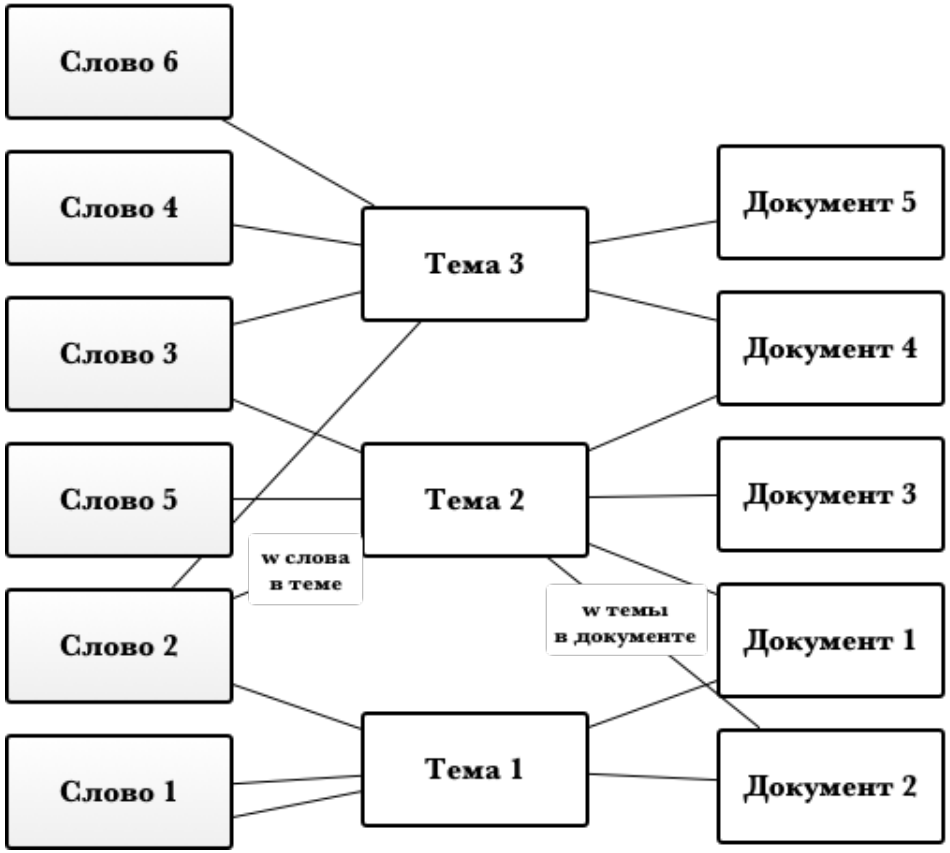


Латентно-семантический анализ

ЛСА помогает определить семантически близкие документы (посвященные одной теме), но непохожие в векторном пространстве (поскольку в них встречаются разные слова). Используем сингулярное разложение для создания нового векторного пространства, в котором у семантически похожие документы будут друг другу ближе.

- Близкие по значению термы формируют одно измерение (т.н. “тему”) в пространстве меньшей размерности
- За счет снижения размерности уменьшается и количество шума

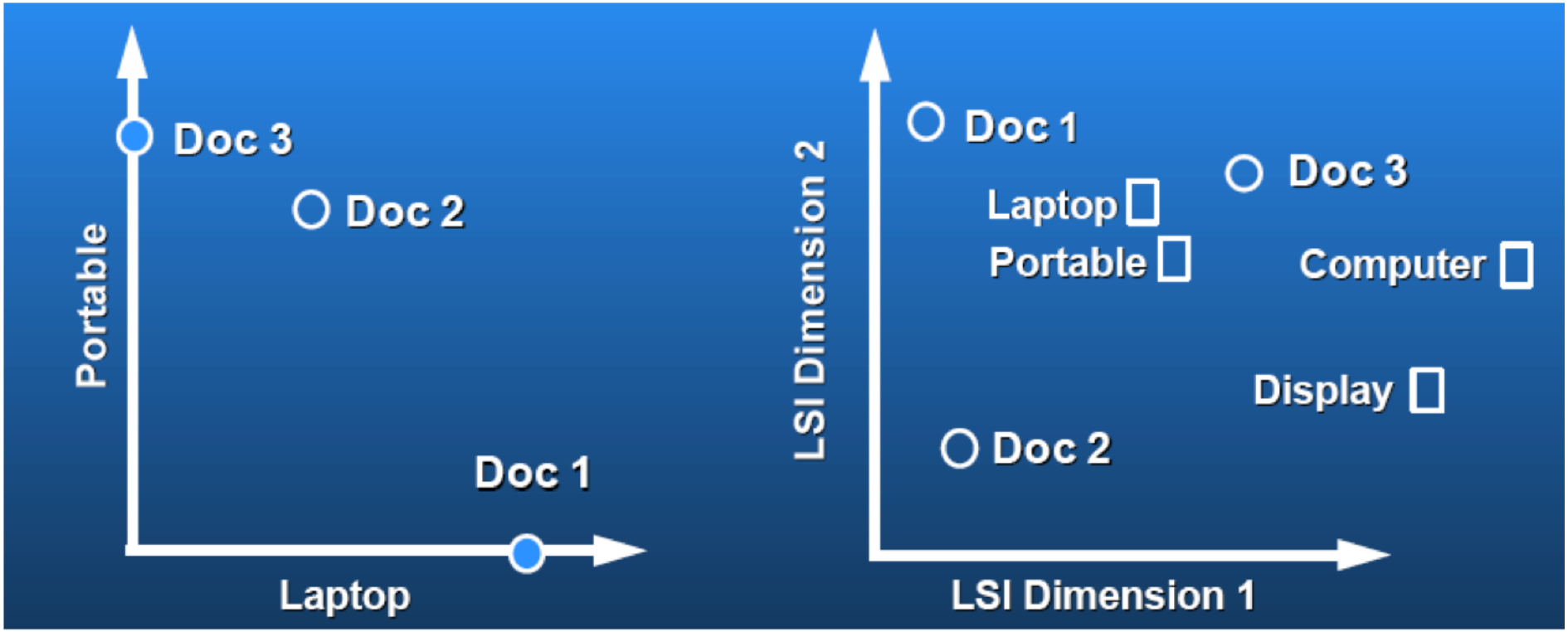
Латентно-семантический анализ



$$C_k = U\Sigma_k V^T$$

- k – число скрытых тем
- U_k – веса слов в темах
- V_k^T – веса тем в документах

Латентно-семантический анализ (Susan Dumais)



Латентно-семантический анализ

1. От матрицы терм-документ C переходим к ее апроксимации $C_k = U_k * \Sigma_k * V_k^T$
2. Преобразование вектора-запроса q: $q_k = \Sigma_k^{-1} U_k^T q$

Эмпирическая проверка

В своих работах Susan Dumais показала, что использование ЛСА в задаче поиска в среднем повышает качество результатов. Что это значит?
Проводится стандартная проверка:

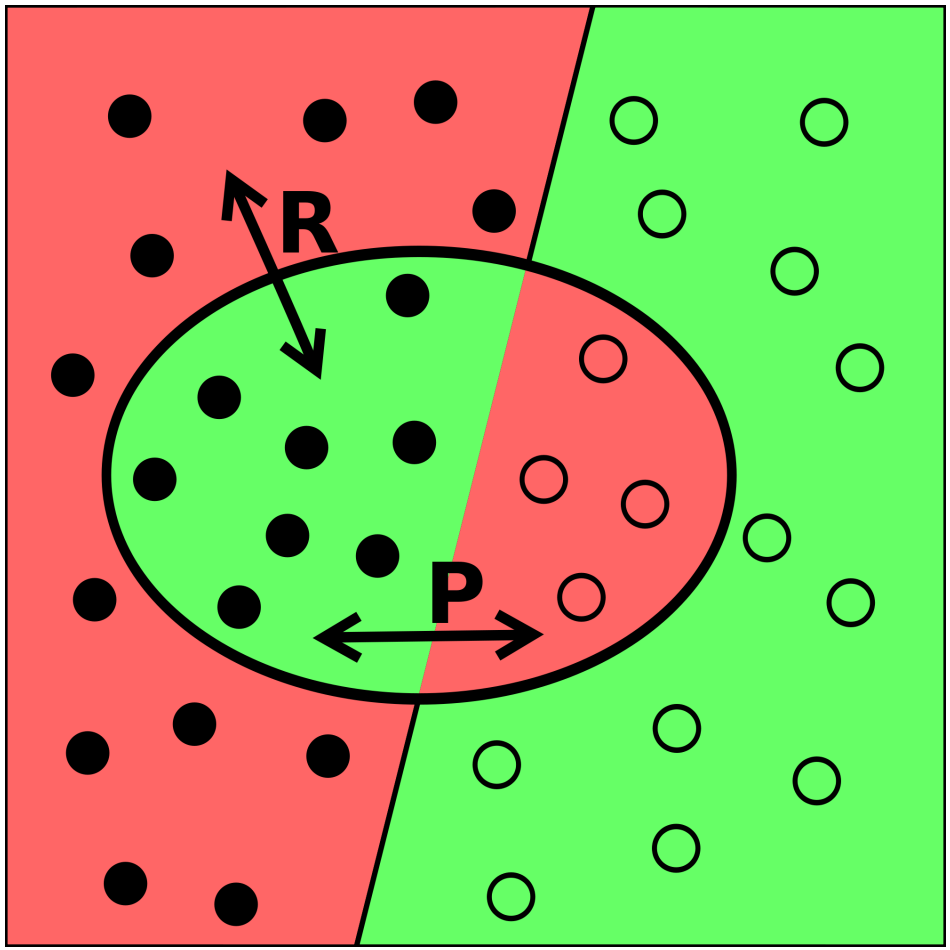
- Есть размеченные коллекции (запросы и релевантные им документы) TREC 1, 2, 3
- Есть baseline: $tf - idf$ и косинусная мера близости
- Есть меры качества: точность и полнота

Меры качества в информационном поиске

$$\text{точность} = \text{precision} = \mathrm{P} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

$$\text{полнота} = \text{recall} = \mathrm{R} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

$$\text{F-measure} = \frac{2\mathrm{PR}}{\mathrm{P} + \mathrm{R}}$$



Латентно-семантический анализ: другие приложения

- “soft clustering”: каждое измерение в пространстве меньшей размерности – это кластер, V_k^T – вектора принадлежности к кластерам
- Рекомендательные системы: вместо термов – объекты, вместо документов – пользователи
- Примитивная невероятностная модель скрытых тем
- Приложения в других областях (например, в компьютерном зрении.

1. Векторная модель

2. Векторная модель в информационном поиске

3. Дистрибутивная семантика

Дистрибутивная семантика

Смысл слова [Ludwig Wittgenstein]

Die Bedeutung eines Wortes liegt in seinem Gebrauch.

Distributional hypothesis [J.R.Firth, 1957]

You shall know a word by the company it keeps!

Что такое “бардьюк”?

- Он подал ей бокан бардьюка.
- Бардьюк подают к блюдам из говядины.
- Ноги у него заплетались, а лицо горело от выпитого бардьюка.
- Виноград сорта бардьюк выращивают в Австралии.
- К простому ужину из хлеба и сыра я взял бутылку отличного местного бардьюка.
- Напитки были прекрасны: кроваво-красный бардьюк и легкое белое рейнское.

doc#50340	так Остро ощущает, когда он влюблен. Эта	компульсия	может существовать и при отсутствии влюбленности
doc#336729	психологии оракул гороскоп такое понятие —	компульсия	— гороскоп притяжение к оракуул, реализующееся
doc#878536	(обсессии); навязчивое поведение (	компульсия	); оппозиционное поведение;
doc#1000748	характерна другая сила тяги, так называемая «	компульсия	». - И что означает это слово в применении
doc#1000748	учителей очень образно описывал, что такое «	компульсия	». Это влечение, сравнимое с жизненно важными
doc#1369221	обеспечиваются равновесие, гомеостазис. Иногда	компульсия	лучше устраняется посредством ее «взрыва
doc#2553333	борьбы с ними. Навязчивое влечение (	компульсия	) — стремление, вопреки разуму, воле и чувствам
doc#3060833	есть субъективный компонент — влечение, или	компульсия	, и объективный — ритуал (вызванные влечением
doc#3575480	рука поднимается вверх и запускается ваша	компульсия	, само ощущение будет буквально утягивать
doc#3575480	направлении. Не то, чтобы у вас исчезла	компульсия	, просто у вас появляется компульсия быть
doc#3575480	исчезла компульсия, просто у вас появляется	компульсия	быть более таким, каким вы хотите быть.
doc#4796843	вещи, которые усиливает эту компульсию, и	компульсия	потеряет всю свою силу. Сама компульсия
doc#4796843	компульсия потеряет всю свою силу. Сама	компульсия	это только то, что лежит на поверхности
doc#4796843	их основе тревога или нечто подавленное.	Компульсия	является защитным механизмом против чувства
doc#4796843	жизни тревогу или депрессию (Чаще всего	компульсия	приводится в действие тревогой-беспокойством
doc#4796843	начните прорабатывать эти чувства. Скоро ваша	компульсия	, независимо от того что вы делаете, уйдет
doc#4796843	непомерную, но полезную службу, которую	компульсия	выполняет для вас. Поблагодарите ее вовлекая
doc#4796843	После того как я обработал эту тревогу,	компульсия	к курению никогда больше не возвращалась
doc#4900615	максимально успешного лечения заболеваний]	КОМПУЛЬСИЯ	compulsion [непреодолимое побуждение совершать
doc#5703937	родственными». Эти близнецы — навязчивость (	компульсия	) и торможение — знакомы каждому, кто испытывал

Дистрибутивная семантика

Векторная модель: матрицы терм-контекст или терм-терм

	$context_1$	$context_2$	...	$context_{\|C\|}$
$term_1$	f_{11}	f_{12}		$f_{1\|C\|}$
$term_2$	f_{21}	f_{22}		$f_{2\|C\|}$
...				
$term_{\|V\|}$	$f_{\|V\|1}$	$f_{\|V\|2}$		$f_{\|V\|\|C\|}$

Термы $|V|$ – обычно индивидуальные слова, чаще всего существительные

Контексты $|C|$:

- Поверхностные: слова или символы в пределах окна
- Текстуальные: тексты целиком или элементы текста (абзацы, предложения)
- Синтаксические: специфические синтаксические связи, чаще всего *subj-of, obj-of*

Веса в дистрибутивной модели

Веса f_{ij}

- бинарный вес: 1, если $term_i$ и $context_j$ встречаются рядом, 0, иначе
- частота: сколько раз $term_i$ и $context_j$ встречаются рядом
- **Positive Pointwise Mutual Information** [Niwa, Nitta, 1994]:

$$PMI(term_i, context_j) = \log_2 \frac{P(term_i, context_j)}{P(term_i)P(context_j)}$$

$$PPMI(term_i, context_j) = \begin{cases} PMI, & \text{if } PMI > 0. \\ 0, & \texttt{иначе}. \end{cases}$$

ММП оценки вероятностей

$$PPMI(term_i, context_j) = \max(\log_2 \frac{P(term_i, context_j)}{P(term_i)P(context_j)}, 0)$$

- $P(term_i, context_j) = \frac{f_{ij}}{\sum_{n=1}^{|V|}\sum_{m=1}^{|C|} f_{nm}}$
- $P(term_i) = \frac{\sum_{m=1}^{|C|} f_{im}}{\sum_{n=1}^{|V|}\sum_{m=1}^{|C|} f_{nm}}$
- $P(context_j) = \frac{\sum_{m=1}^{|V|} f_{nj}}{\sum_{n=1}^{|V|}\sum_{m=1}^{|C|} f_{nm}}$

(P)PMI придает больший вес редким событиям. Дополнительная корректировка на $\alpha = 0.75$:

$$P_\alpha(context_j) = \frac{(\sum_{m=1}^{|V|} f_{nj})^\alpha}{\sum_{m=1}^{|C|}(\sum_{m=1}^{|V|} f_{nm})^\alpha}$$

Меры близости

Для определения близости между термами

- косинусная мера близости:

$$cos(term_i, term_j) = \frac{t_i \times t_j}{||t_i||||t_j||} = \frac{\sum_k f_{ik} \times f_{jk}}{\sqrt{(\sum_k f_{ik})^2}\sqrt{(\sum_k f_{jk})^2}}$$

- мера Жаккара:

$$jc(term_i, term_j) = \frac{\sum_k min(f_{ik}, f_{jk})}{\sum_k max(f_{ik}, f_{jk})}$$

От разреженных векторов к эмебддингам (embedding)

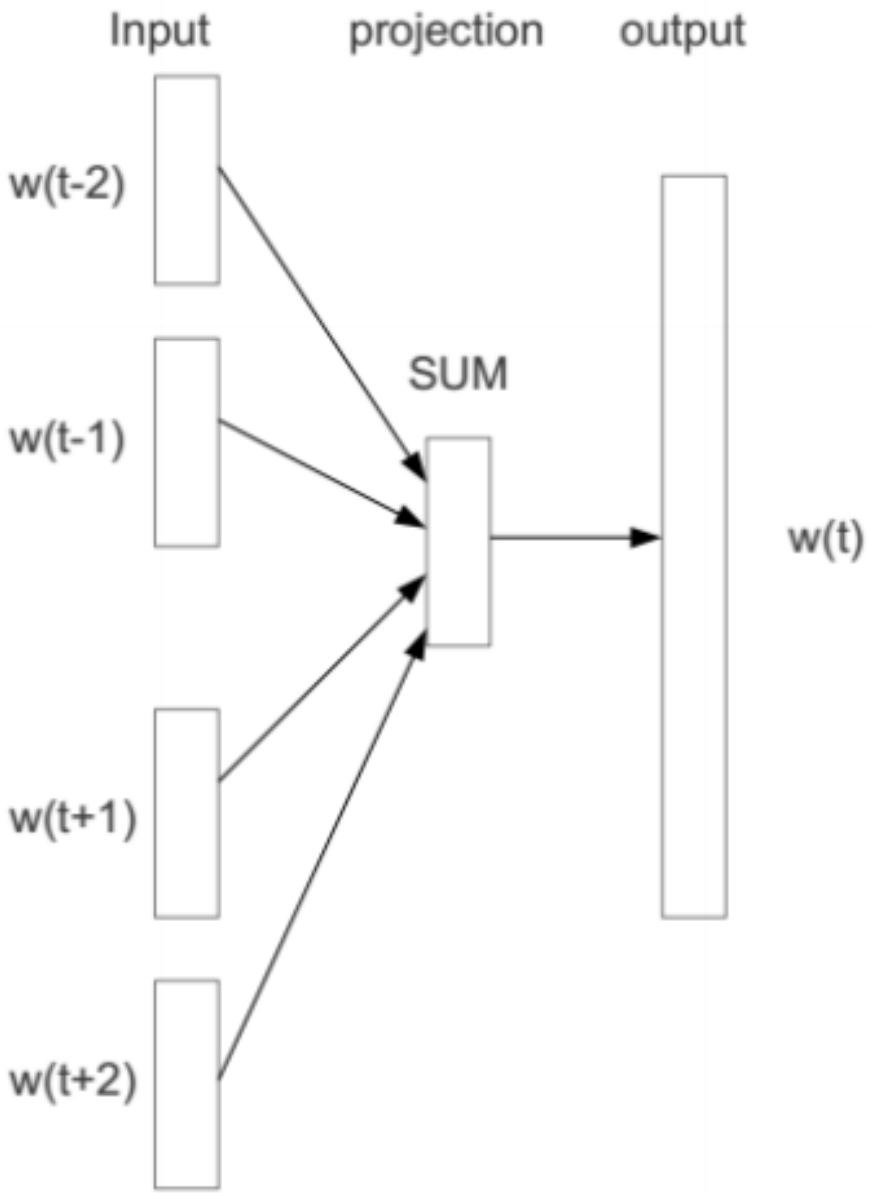
- Счетные модели:
  - Сингулярное разложение матрицы $P(PMI)$: $C_k = U\Sigma_k V^T$
    U_k – вектора слов
- Предективные модели:
  - skip-gram, CBOW aka word2vec [Mikolov et al., 2013]
  - dependency embeddings [Levi et al., 2015]
  - GloVe [Pennington et al., 2014]

Как оценить качество?

- WordSim-353 – коллекция из 353 пар слов, для каждой пары определено сходство по шкале от 0 до 10
- задания из теста TOEFL: выбрать одно подходящее слово из 4, выбрать лишнее слово
- Google 20K dataset: пары вопросы и ответы
- Словари синонимов и тезаурусы.

Continuous bag-of-words model (CBOW) [Mikolov, 2013]

- Входной слой: контекст слова (+, – 2 слова слева и справа)
- Слой проекции: линейный
- Выходной слой: вектор слова



skip-gram [Mikolov, 2013]

- Обратная задача: предсказание векторов контекста по данному слову
- Выходной слой: вектора слова
- Если объем обучающего множества и количество эпох достаточно велики, то вектора, полученные для одного и того же слова с помощью CBOW и skip-gram архитектуры похожи
- Негативное сэмплирование: не все отрицательные контексты важны