Topic modelling and dimensionality reduction

masha.shejanova@gmail.com

Тематическое моделирование

Что и зачем

Тема — “о чём документ” ≃ набор часто совместно встречающихся слов

Мы считаем, что тема употребление того или иного слова зависит от темы. А тема — от документа.

Зачем:

- поиск в электронных библиотеках
- трекинг новостных сюжетов
- “продвинутый” эмбеддинг документа

Базовое предположение

- каждый **документ** состоит из смеси некоторых **тем *(topics)***
- каждая **тема** состоит из набора **слов**

Иными словами, темы — это скрытые (латентные) переменные, которые управляют распределением слов в документе.

topic modeling vs. clustering

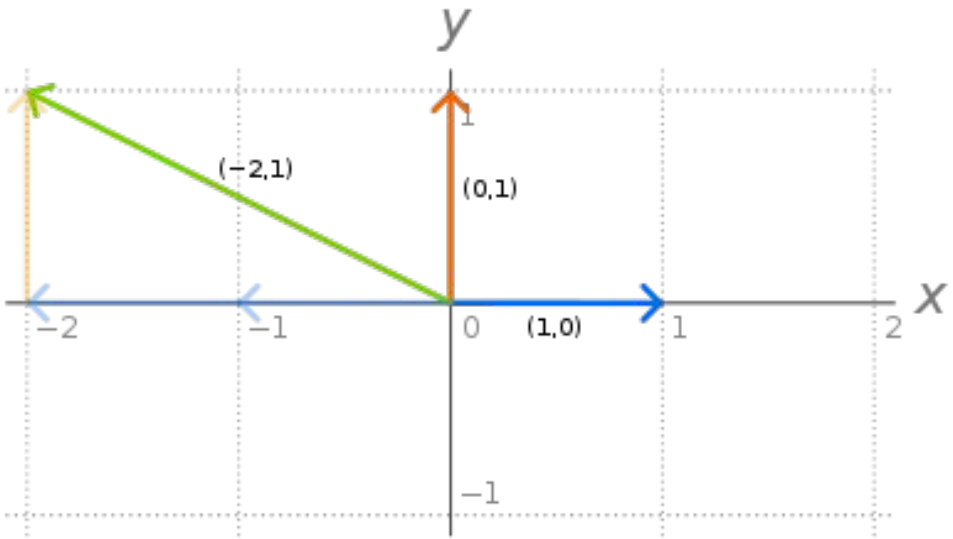
Что похожего: есть документы, раскидываем их по кучкам, заранее не знаем по каким.

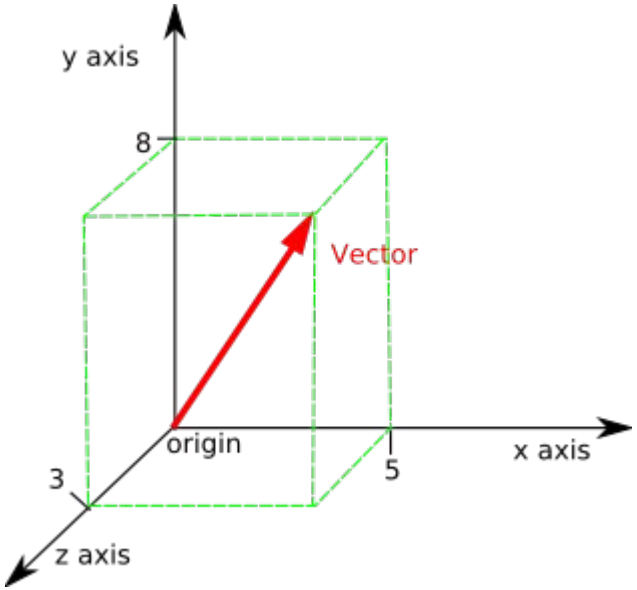
Что разного: у одного документа может быть высокая степень принадлежности больше, чем к одной теме.

PCA (метод главных компонент): идея

Базис линейного пространства

Стандартный базис:





Замена базиса

На самом деле, базисные вектора можно выбирать как угодно — главное чтобы можно было выразить через них все вектора пространства.

(И чтобы сами базисные вектора нельзя было выразить друг через друга).

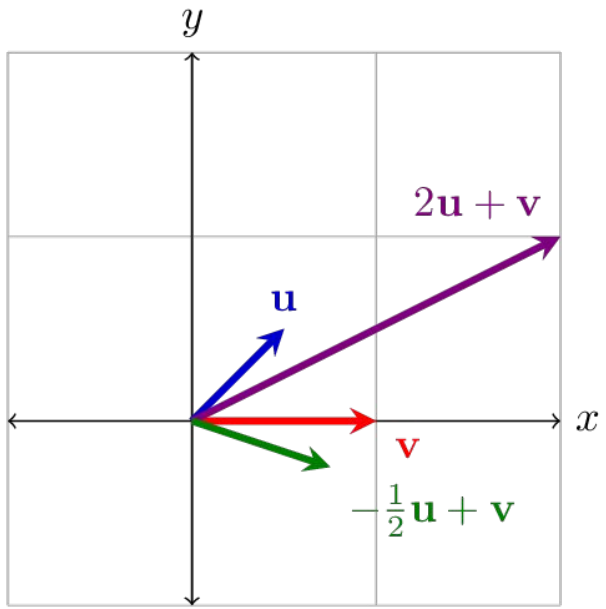
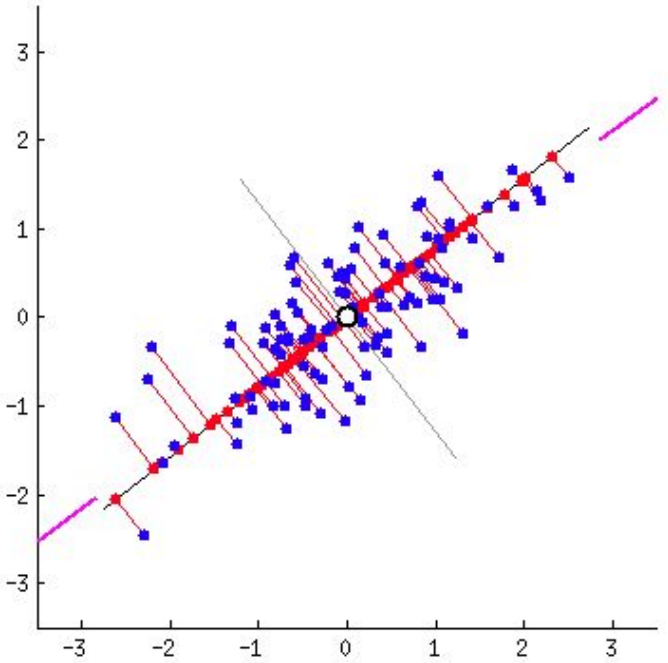


Figure 1: Vector combinations.

PCA

Найдём такой базис, чтобы как можно лучше выразить как можно больше значений за счёт фиксированного количества базисных векторов.

Сделаем проекцию всех данных на эти вектора.



SVD (сингулярное разложение): реализация

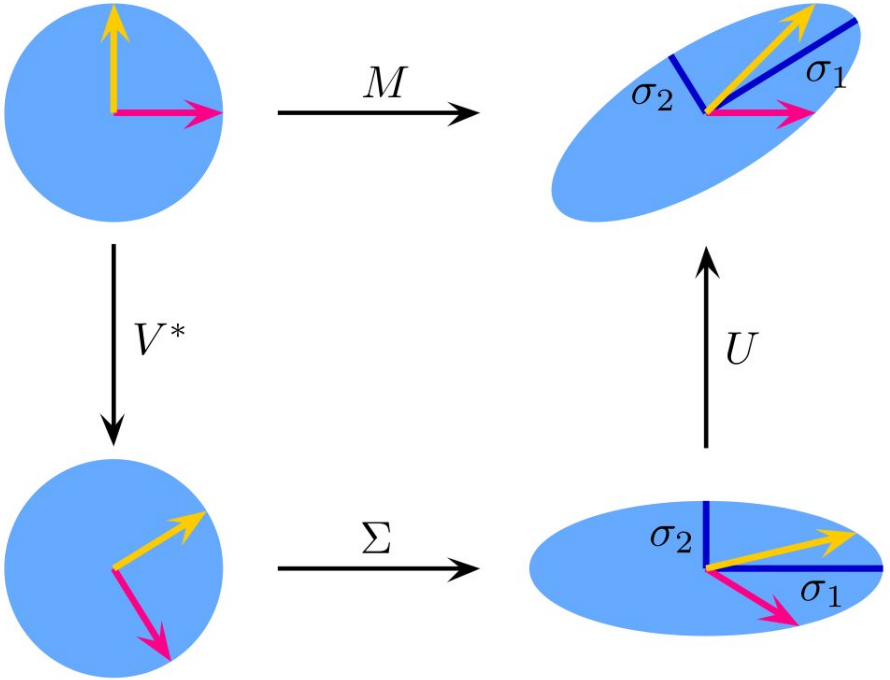
SVD

Любую матрицу M можно разложить на произведение трёх матриц: M = U * Σ * V^*

U, V^* — матрицы поворота

Σ — матрица растяжения

У Σ числа стоят только на главной диагонали, причём они убывают. Это сингулярные значения, т.е. корни из собственных значений.

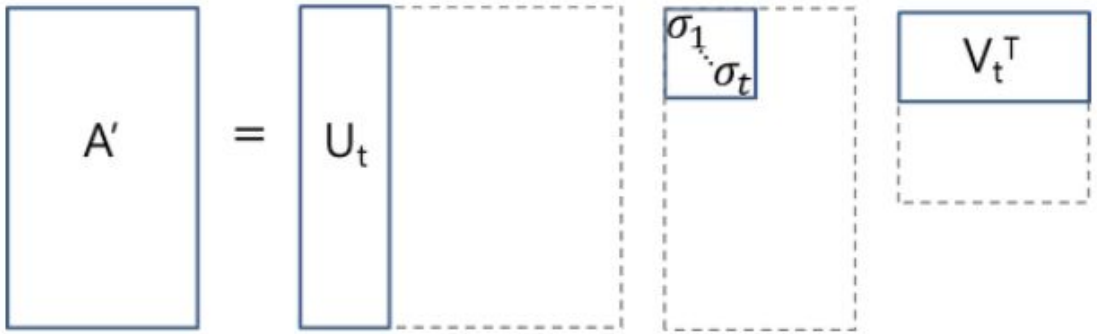


$$M = U \cdot \Sigma \cdot V^*$$

Truncated SVD

$$A \approx U_t S_t V_t^T$$

Intuitively, think of this as only keeping the t most significant dimensions in our transformed space.



(скрин из вот этой статьи)

Truncated SVD = LSA (latent semantic analysis) в тематическом моделировании

Что в итоге

- в средней матрице диагонали по убыванию выстроены компоненты — измерения “хорошего” базиса; чем выше, тем значимей компонента
- выбираем первые n (сколько хотим) компонент; их мы будем сохранять, а остальные выкинем
- в итоге значительно сократим объём используемой памяти
- и дополнительно получим разложение документов по этим компонентам, или, как говорят в тематическом моделировании, темам
- (NB: компоненты выстроены по убыванию для всего датасета, но каждый документ имеет свои пиковые компоненты)

Для текстов: матрица слово-документ

Для начала, считаем матрицу, сколько раз какое слово вошло в какой документ, например, с помощью CountVectorizer.

	котик	играть	авокадо	манго
документ 1	2	2	0	0
документ 2	1	3	0	0
документ 3	1	2	0	1
документ 4	0	1	1	2
документ 5	0	0	3	2

Truncated SVD на текстах

В таблице невооружённым глазом заметен паттерн: есть тексты про домашних животных (слова *котик*, *играть* и т.д.), а есть — про овощи и фрукты (слова *авокадо*, *манго* и т.д.). Эту матрицу мы раскладываем с помощью SVD.

В матрице Σ из произведения $U*\Sigma*V^*$ элементы будут соответствовать вот этим обобщённым концептам — темам.

Вместо векторов размером в словарь получаем вектора такого размера, какой захотим оставить.

	котик	играть	авокадо	манго
документ 1	2	2	0	0
документ 2	1	3	0	0
документ 3	1	2	0	1
документ 4	0	1	1	2
документ 5	0	0	3	2

Truncated SVD в sklearn

from sklearn.decomposition import TruncatedSVD

гиперпараметры:

- n_components — какого размера должны быть конечные векторы
- algorithm — randomized, arpack
- n_iter

Может применяться в связке с классификацией.

SVD на собачках

SVD можно применять и для других матриц — например, для картинок, ведь картинки — это просто матрицы из пикселей.

При большом количестве компонент разница незаметна.

(источник)

pLSA: probabilistic LSA

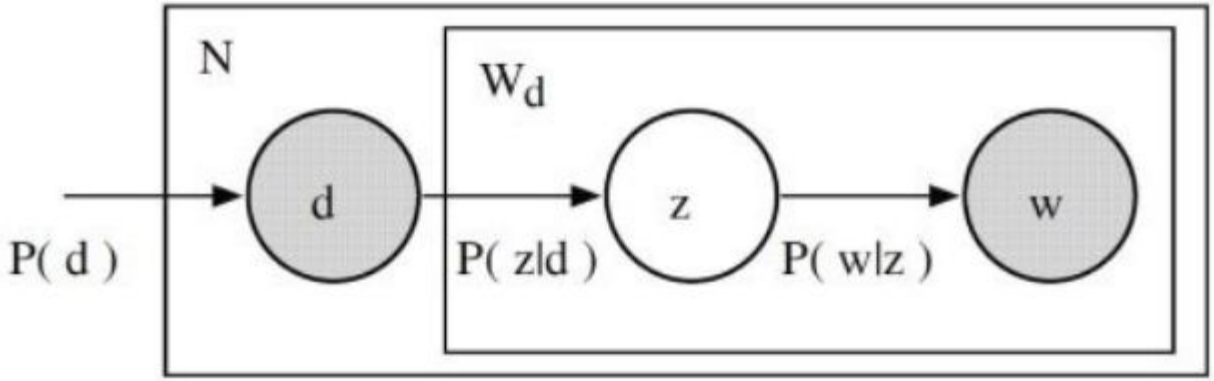
Probabilistic Latent Semantic Analysis

- использует вероятности
- генеративная

Два предположения:

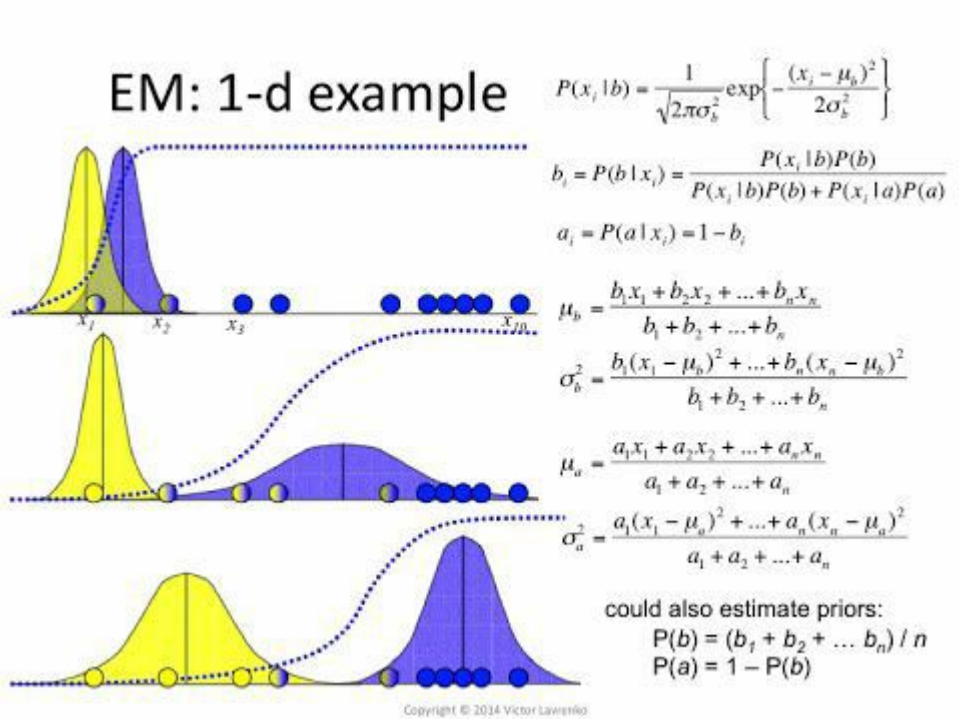
- **документ** состоит из смеси некоторых **тем** ➜ topic **z** is present in that document **d** with probability **P(z|d)**
- каждая **тема** состоит из набора **слов** ➜ given a topic **z**, word **w** is drawn from z with probability P(w|z)

Probabilistic Latent Semantic Analysis



$$P(D,W) = P(D)\sum_{Z} P(Z|D)P(W|Z)$$

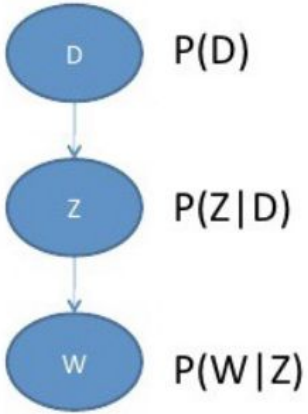
EM-алгоритм



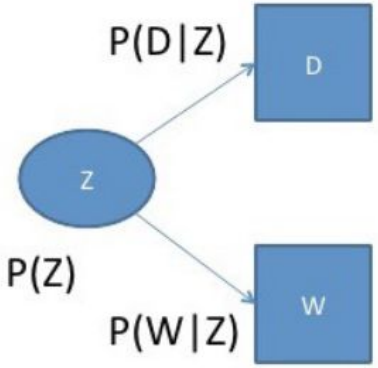
- видео с объяснением (картинка оттуда)
- картинка-схема шагов алгоритма

pLSA vs. SVD: другой способ разложить формулу

- Start with document



- Start with topic



$$P(D,W) = \sum_Z P(Z)P(D|Z)P(W|Z)$$

$$A \approx U_t S_t V_t^T$$

Что ещё бывает? (Более продвинутые вещи)

- LDA (a Bayesian version of pLSA, использует распределение дирихле)
- ARTM — LDA, но с регуляризацией
- bigARTM — ARTM с наворотами :) (но вообще, это библиотека, в которой есть все эти методы и больше!)

Снижение размерности

Что и зачем

В общем случае — у нас есть признаковое пространство на много-много измерений (например, мешок слов по корпусу, и каждое слово — признак). Мы хотим “сжать” их как-то так, чтобы потерять минимум информации.

Каждое новое “измерение” — элемент вектора — будут заключать в себе обобщённое представление нескольких элементов из большого вектора.

- убрать несущественные признаки
- тематическое моделирование
- визуализация

SVD

LSA == PCA == Truncated SVD

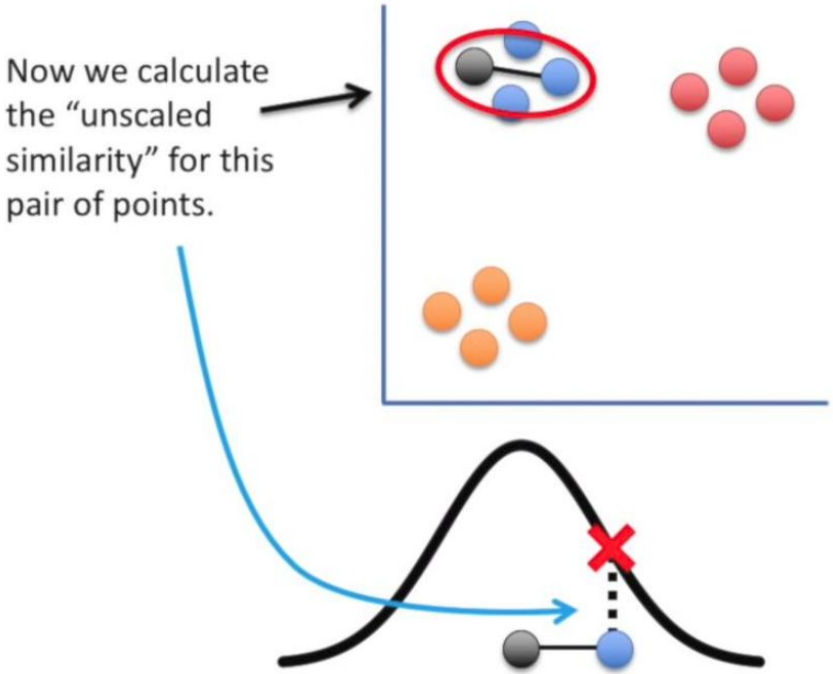
t-SNE

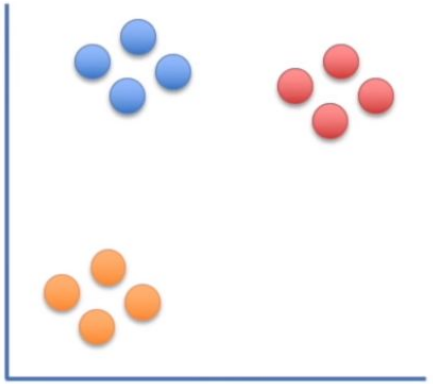
- используется для визуализации
- хорош только для перевода в очень маленькие размерности

Шаги:

- посчитать расстояние от каждой точки до каждой другой (используя формулу нормального (SNE) или T распределения (t-SNE))
- случайно породить соответствующие им точки в маленькой размерности
- решить задачу оптимизации: надо, чтобы распределения расстояний (реальных и в пространстве маленькой размерности) максимально совпадали

t-SNE





At each step, a point on the line is attracted to points it is near in the scatter plot, and repelled by points it is far from…