Матричные разложения и их применения в анализе данных

Руслан Хайдуров, Анастасия Иовлева

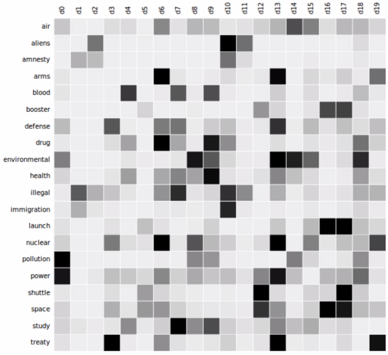
16 октября 2017

Содержание

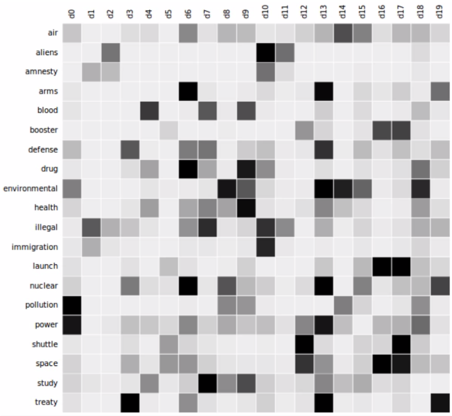
Latent semantic analysis

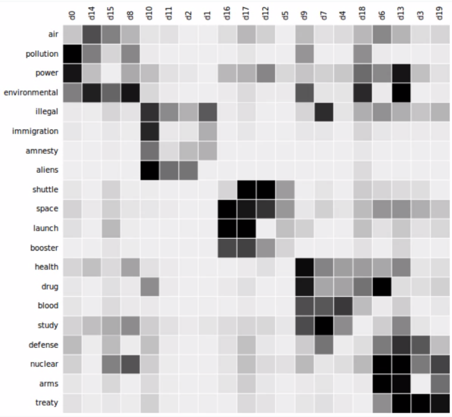
Выявление латентных зависимостей внутри множества документов

- сравнить документы
- сравнить слова
- отношения между словами (например, синонимы)



Пример


d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13 d14 d15 d16 d17 d18 d19
air
aliens
amnesty
arms
blood
booster
defense
drug
environmental
health
illegal
immigration
launch
nuclear
pollution
power
shuttle
space
study
treaty


d0 d14 d15 d8 d10 d11 d2 d1 d16 d17 d12 d5 d9 d7 d4 d18 d6 d13 d3 d19
air
pollution
power
environmental
illegal
immigration
amnesty
aliens
shuttle
space
launch
booster
health
drug
blood
study
defense
nuclear
arms
treaty

Матричные разложения

Используем SVD:

$$X \approx USV^T$$

$X \in \mathrm{Mat}(m, n)$ — исходная document-term матрица,
$U \in \mathrm{Mat}_{m \times t}, V \in \mathrm{Mat}_{n \times t}$ — ортогональные матрицы,
S — диагональная матрица.

Столбцы U и V соотносятся с темами текстов. Теперь мы можем, например, сравнить похожесть i-го и j-го слова, сравнив i-ую и j-ую строчки V.

Преимущества

- Уменьшение размерности
- Может использоваться без обучения (кластеризация)
- Кластеризует документы почти как человек
- Уменьшает количество синонимов, различает омнонимы
- Независимость от языка
- Устойчив к шуму (опечатки, ошибки типографии...)

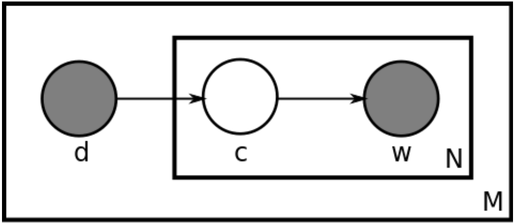
Недостатки

- Вероятностная модель метода не соответствует реальности (распределение Пуассона).
- Неясная интерпретация с точки зрения естественного языка

$$\{(\mathrm{car}), (\mathrm{truck}), (\mathrm{flower})\} \to \{(\mathrm{x} \cdot \mathrm{car} + \mathrm{y} \cdot \mathrm{truck}), (\mathrm{flower})\}$$
$$\{(\mathrm{car}), (\mathrm{bottle}), (\mathrm{flower})\} \to \{(\mathrm{x} \cdot \mathrm{car} + \mathrm{y} \cdot \mathrm{bottle}), (\mathrm{flower})\}$$

- Не отличает разные значения одного слова
- Зависит от SVD

Probabilistic latent semantic analysis

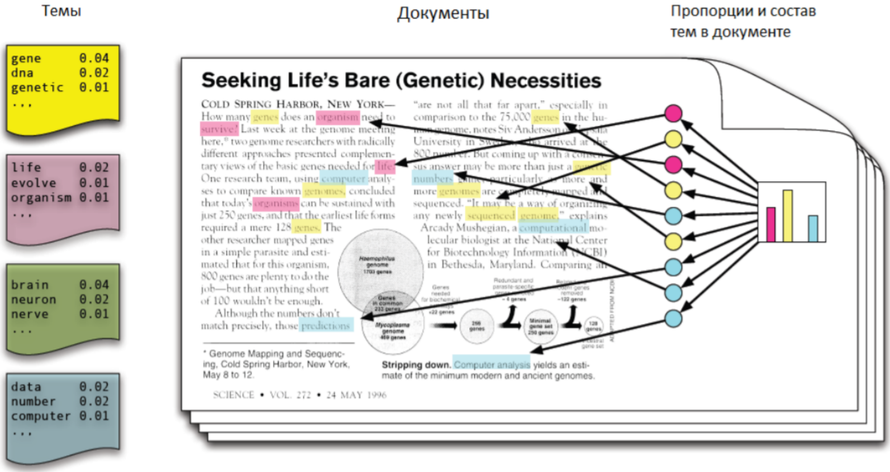


Использует мультиномиальное распределение:

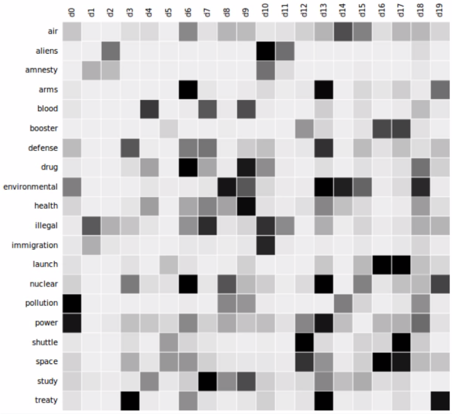
$$P(w \mid d) = P(d) \sum_{c} P(c \mid d) P(w \mid c).$$

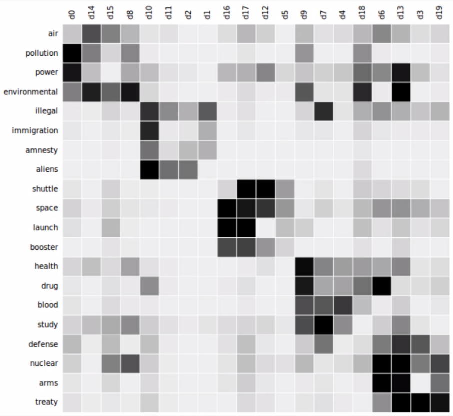
Обучается с помощью EM-алгоритма

Тематическое моделирование



Тематическое моделирование





Зачем?

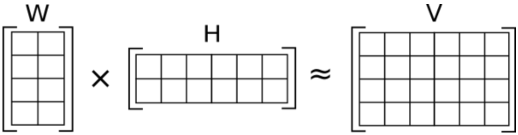
- Тематический поиск
- Ранжировать по сходству с заданным фрагментов
- Классификация текстов, правила каталогизации новых документов
- Суммаризация и аннотация коллекций документов
- Как темы изменялись со временем (если есть время создания документа)
- Определить тематику авторов, журналов, конференций и т.д. (если они известны)
- Разделить документ на тематически однородные фрагменты

Где используется?

- Анализ коллекций научных статей, новостных потоков
- Рубрикация коллекций текстов, изображений, видео, музыки
- Задачи биоинформатики (например, аннотация генома)
- ...и многое другое

Содержание

Non-negative matrix factorization



W и H — матрицы с неотрицательными элементами.
$W \in \mathrm{Mat}_{n \times r}$, $H \in \mathrm{Mat}_{r \times m}$, $V \in \mathrm{Mat}_{n \times m}$.

Зачем?

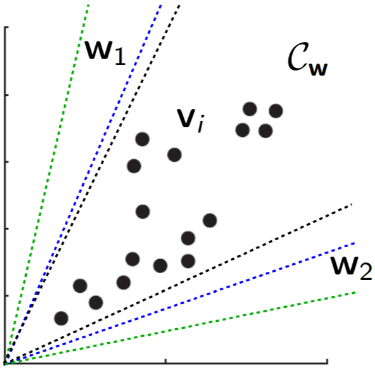
- Понижение размерности.
- Неотрицательные элементы удобно интерпретировать.
- Автоматически кластеризует данные: столбцы W — центроиды кластеров, H — индикаторы классов (если $H_{kj} > 0$, то j-ый столбец V принадлежит классу k).

Проблемы

- Решения может не существовать
- Или их бесконечно много:

$$V = WH = WBB^{-1}H.$$

- Требование на неотрицательность усложняет вычисления
- Чувствителен к начальному приближению



Как решать?

Фактически это задача оптимизации:

$$(W^*, H^*) = \mathrm{argmin}\ D(V, WH).$$

Функция потерь:

$$D(X, \hat{X}) = \sum_{i=1}^{n} \sum_{j=1}^{m} d(x_{ij}, \hat{x}_{ij}); \quad d(x, \hat{x}) \geq 0,\ d(x, \hat{x}) = 0 \leftrightarrow x = \hat{x}.$$

Функции потерь

Функция потерь — замаскированное правдопобие. Поэтому в разных областях применения (биоинформатика, тематическое моделирование, анализ аудитозапией и т.д.) используются разные функции потерь.
Несколько примеров:

	$d(x, \hat{x})$
Норма Фробениуса	$(x - \hat{x})^2$
Обобщеная дивергенция Кульбака-Лейблера	$x \ln \frac{x}{\hat{x}} - x + \hat{x}$
Дивергенция Итакура-Саито	$\ln \frac{\hat{x}}{x} + \frac{x}{\hat{x}} - 1$

Блочно-покоординатная оптимизация

Проблема: обычно функционал ошибки невыпуклый по совокупности элементов — нельзя использовать методы поиска глобального минимума.
Решение: Блочно-покоординатная оптимизация. Фиксируем одну из матриц, обновляем вторую матрицу. Повторяем, пока не сойдемся.

$$H = f(V, W, H_{\mathrm{prev}}),$$
$$W = g(V, W_{\mathrm{prev}}, H),$$
$$f(V, W, H) = g^T(V^T, W^T, H^T).$$

Инициализация

От выбора начального приближения зависит, в какой локальный минимум мы попадем.

Возможные варианты генерации начальных значений матриц W и H:

- Случайные матрицы
- Кластеризация грубым методом
- SVD

$$\underbrace{X(:,j)}_{j\text{th facial image}} \approx \sum_{k=1}^{r} \underbrace{W(:,k)}_{\text{facial features}} \underbrace{H(k,j)}_{\substack{\text{importance of features}\\ \text{in } j\text{th image}}} = \underbrace{WH(:,j)}_{\substack{\text{approximation}\\ \text{of } j\text{th image}}} .$$