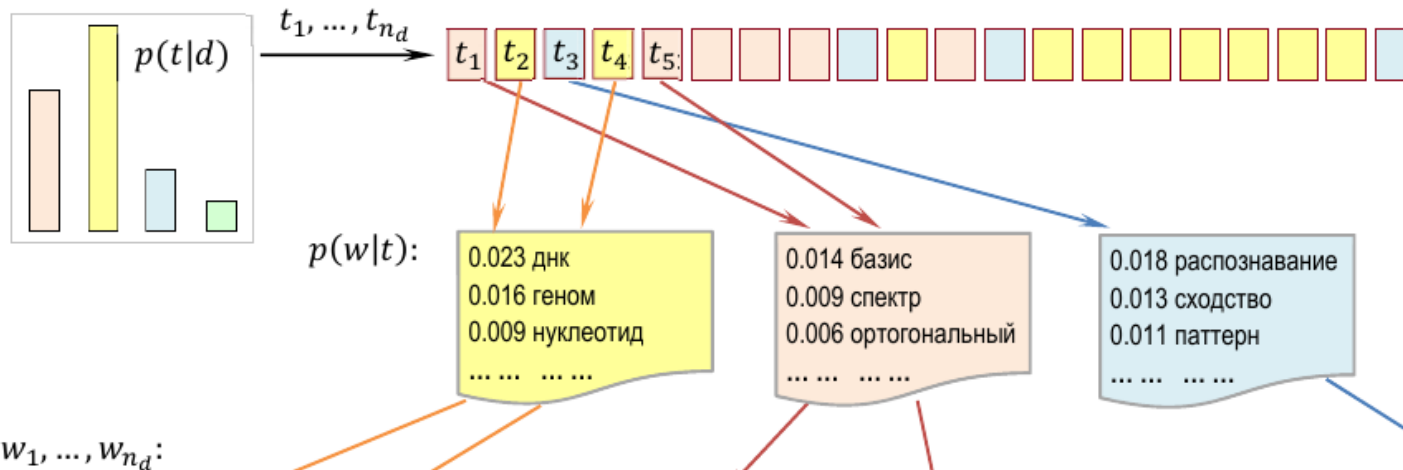


Машинное обучение

Тематическое моделирование



Разработан спектрально-аналитический подход к выявлению размытых протяженных повторов в геномных последовательностях. Метод основан на разномасштабном оценивании сходства нуклеотидных последовательностей в пространстве коэффициентов разложения фрагментов кривых GC- и GA-содержания по классическим ортогональным базисам. Найдены условия оптимальной аппроксимации, обеспечивающие автоматическое распознавание повторов различных видов (прямых и инвертированных, а также тандемных) на спектральной матрице сходства. Метод одинаково хорошо работает на разных масштабах данных. Он позволяет выявлять следы сегментных дупликаций и мегасателлитные участки в геноме, районы синтении при сравнении пары геномов. Его можно использовать для детального изучения фрагментов хромосом (поиска размытых участков с умеренной длиной повторяющегося паттерна).

Содержание лекции

- Постановка задачи
- Предыстория
- Латентный семантический анализ (LSA)
- Вероятностный LSA (PLSA)
- Латентное размещение Дирихле (LDA)
- Учет контекста

Предыстория

- Векторная модель документов:
 $d_j = (w_{1j}, w_{2j}, \dots, w_{nj})$
- w_{ij} – вес i -того слова в j -том документе
- Методы взвешивания термов:
 - булевский вес (0,1)
 - Tf - term frequency (функция от количества вхождений слова в документ)
 - Tf-idf = TF*IDF
- Близость между документами (или запросом и документом) вычислялась по правилу косинуса:

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

Недостатки векторной модели

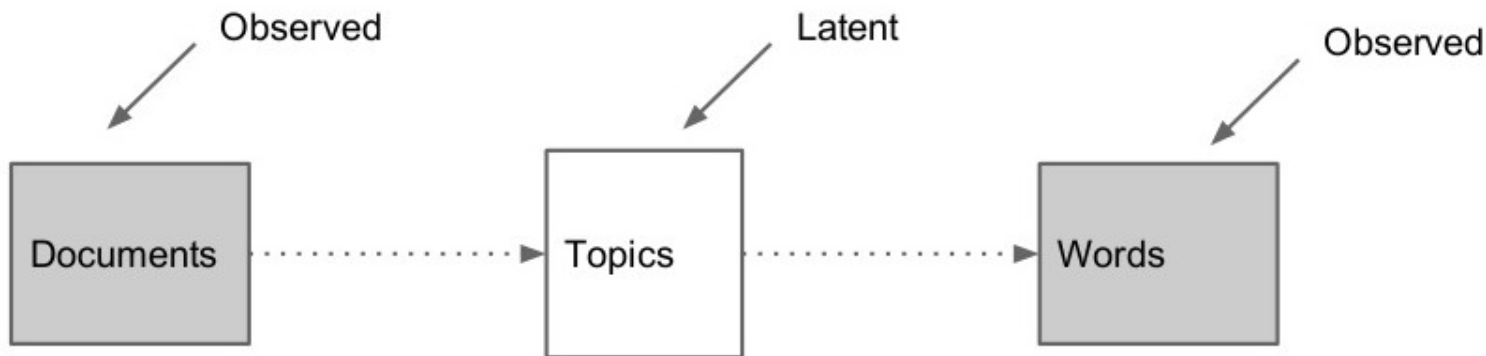
- Проблемы с большими документами (они дают приближенно равные маленькие скалярные произведения) – это “проклятие размерности”
- Поисковая система находит документы только со словами из запроса. Документы на ту же тему, но другими словами – не находятся

Тематическое моделирование

- Для анализа текстов удобно описывать их небольшим набором тем (математически: понизить размерность)
- Это позволит легко:
 - проводить категоризацию документов
 - аннотировать тексты
 - вычислять близость документов
- Сферы применения: информационный поиск, категоризация, поиск рецензентов/экспертов, рекомендательные системы, аннотирование изображений,...

Латентный семантический анализ

- Ключевая идея: любой текст является смесью небольшого количества скрытых (latent) составных элементов (тем)



- Пример: наша лекция сегодня состоит из линейной алгебры, теории вероятностей, моделирования
Значит, она с большой вероятностью должна содержать слова:
ЛА: вектор, скалярное произведение, ортогональный, SVD-разложение, ...
ТВ: вероятность, Байес, при условии, распределение, ...
М: модель, соответствие, проверка, ...

Латентный семантический анализ

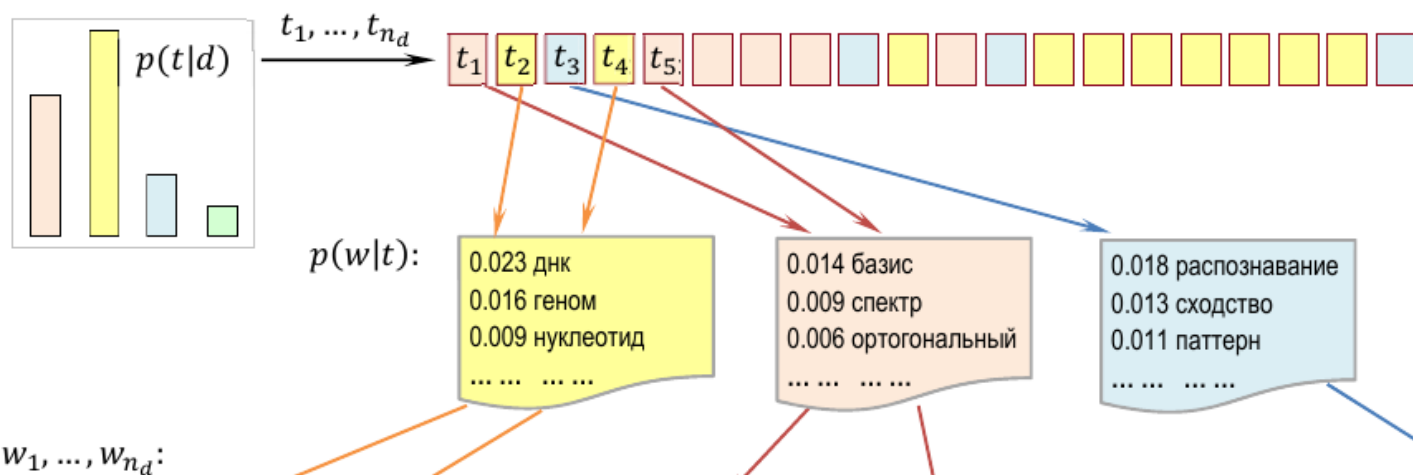
- [Deerwester и др. '90]:
- Считает частоты встречаемости слов в каждом документе
- Записывает их в term-document матрицу
- Понижает размерность по методу PCA: SVD-разложение + сокращение числа компонент
- Темы – небольшой набор ортогональных векторов, лучшим образом приближающий исходную линейную оболочку документов

Вероятностный подход к LSA (PLSA)

- Вместо сокращения размерности документов предположим, что документы – случайны и порождены совместными вероятностными распределениями:
(слова, темы) и (темы, документы)
- Найдем параметры этого распределения
- Математически получается та же формула, что и в SVD. Но SVD находит темы, оптимизируя евклидово расстояние, а PLSA - правдоподобие

Случайный процесс порождения документов в модели PLSA

$$p(w|d) = \sum_{t \in T} p(w|t) p(t|d)$$



Разработан спектрально-аналитический подход к выявлению размытых протяженных повторов в геномных последовательностях. Метод основан на разномасштабном оценивании сходства нуклеотидных последовательностей в пространстве коэффициентов разложения фрагментов кривых GC- и GA-содержания по классическим ортогональным базисам. Найдены условия оптимальной аппроксимации, обеспечивающие автоматическое распознавание повторов различных видов (прямых и инвертированных, а также тандемных) на спектральной матрице сходства. Метод одинаково хорошо работает на разных масштабах данных. Он позволяет выявлять следы сегментных дупликаций и мегасателлитные участки в геноме, районы синтении при сравнении пары геномов. Его можно использовать для детального изучения фрагментов хромосом (поиска размытых участков с умеренной длиной повторяющегося паттерна).

Восстановление плотности $p(w|t)$ и $p(t|d)$

- Дано: n_{dw} – количество вхождений слова w в документ d , n_d – размер документа

$$\frac{n_{dw}}{n_d} \approx p(w|d)$$

- Найти: вероятности терминов в темах, вероятности тем в документах

$$\phi_{wt} = p(w|t)$$

$$\theta_{td} = p(t|d)$$

так, чтобы выполнялось равенство:

$$p(w|d) = \sum_{t \in T} \phi_{wt} \theta_{td}$$

Принцип максимума правдоподобия

- Правдоподобие коллекции документов:

$$\prod_{i=1}^n p(d_i, w_i) = \prod_{d \in D} \prod_{w \in d} p(d, w)^{n_{dw}}$$

- Максимизация логарифма правдоподобия

$$\sum_{d \in D} \sum_{w \in d} n_{dw} \ln p(w|d)p(d) \rightarrow \max_{\Phi, \Theta}$$

эквивалентна максимизации функционала:

$$\mathcal{L}(\Phi, \Theta) = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_t \phi_{wt} \theta_{td} \rightarrow \max_{\Phi, \Theta}$$

EM-алгоритм

- E-шаг 1:
Оцениваем число слов в документе d , порожденных темой t . По формуле Байеса:

$$p(t|d, w) = \frac{p(w, t|d)}{p(w|d)} = \frac{p(w|t)p(t|d)}{p(w|d)} = \frac{\phi_{wt}\theta_{td}}{\sum_s \phi_{ws}\theta_{sd}}$$

отсюда:

$$n_{td} = \sum_w n_{wd} \frac{\phi_{wt}\theta_{td}}{\sum_s \phi_{ws}\theta_{sd}}$$

- M-шаг 1:
оценка вероятности темы в документе

$$\theta_{td} = p(t|d) = \frac{n_{td}}{n_d}$$

EM-алгоритм

- E-шаг 2:
Оцениваем количество вхождений слова w в тему t

$$n_{wt} = \sum_d n_{wd} \frac{\phi_{wt} \theta_{td}}{\sum_s \phi_{ws} \theta_{sd}}$$

- M-шаг 2:
оценка вероятности вхождения слова в тему

$$\phi_{wt} = p(w|t) = \frac{n_{wt}}{n_t}$$

Неединственность решения

- Если задача разрешима, то решений бесконечно много:

$$\begin{pmatrix} n_{dw} \\ n_d \end{pmatrix}_{W \times D} \approx \underset{W \times T}{\Phi} \cdot \underset{T \times D}{\Theta} = (\Phi S)(S^{-1} \Theta) = \underset{W \times T}{\Phi'} \cdot \underset{T \times D}{\Theta'}$$

S – произвольная невырожденная матрица

Недостатки PLSA

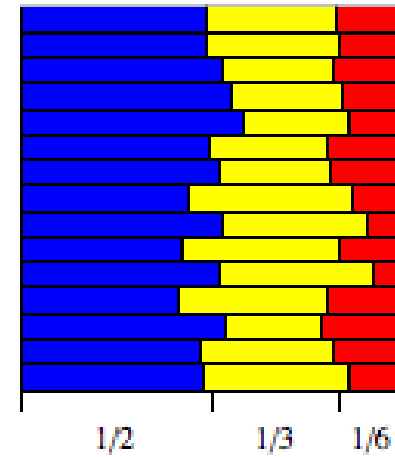
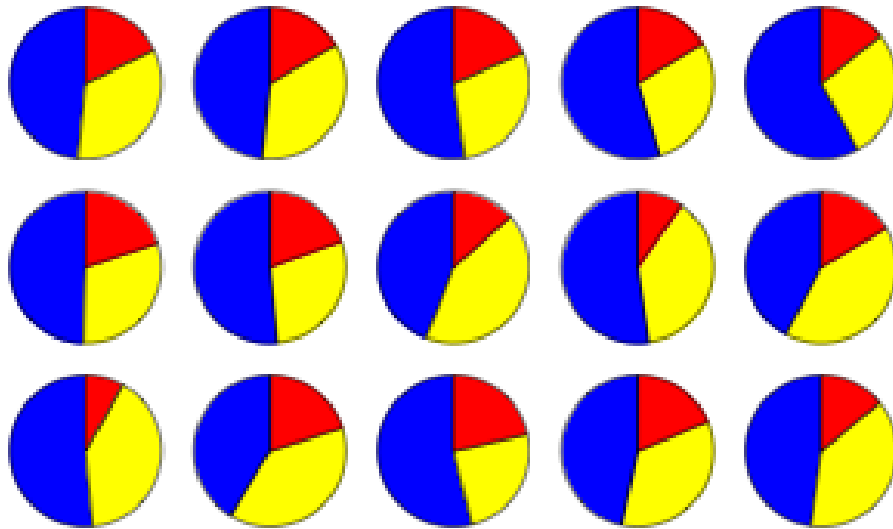
- Переобучение
- Некоторые документы имеют пару выраженных тем, а для многих – почти все темы по чуть-чуть присутствуют
- Если уменьшать число тем – получится плохая модель, если увеличивать – много документов с тематической неопределенностью
- Как сделать тем много и решить задачу с условием: в каждом документе не более 3-5 тем?
Именно для этого служит LDA.

Распределение Дирихле

- Пусть отрезок длины 1 разрезан на K частей. Рассмотрим эксперимент с фиксированным исходом: случайно (равномерно) бросили несколько точек на отрезок и в каждой части оказалось ровно $\alpha_i - 1$ штук. Какими могут быть длины частей?
- Ответ: они распределены по закону Дирихле!
- Вероятность того, что вероятность каждого из K взаимоисключающих событий равна x_i при условии, что каждое событие наблюдалось $\alpha_i - 1$ раз

$$f(x_1, \dots, x_K; \alpha_1, \dots, \alpha_K) = \frac{1}{B(\alpha)} \prod_{i=1}^K x_i^{\alpha_i - 1}$$
$$\sum_{i=1}^K x_i = 1$$

Наглядная трактовка



$$\alpha_1=3 \quad \alpha_2=2 \quad \alpha_3=1$$

Что получится, если возьмем $\alpha_1=30 \quad \alpha_2=20 \quad \alpha_3=10$?

Другая трактовка: урны и шары

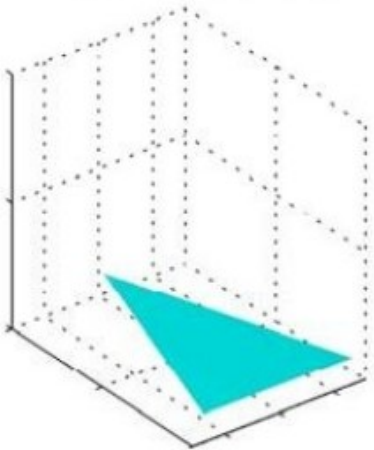
- Рассмотрим урну с шарами K различных цветов. Изначально в ней α_1 шаров цвета 1, α_2 – цвета 2, ...
- Возьмем случайно из урны шар и положим его назад вместе с шаром того же цвета
- Если повторять это бесконечно много раз, то пропорции цветов в урне будут подчинены распределению Дирихле $\text{Dir}(\alpha_1, \dots, \alpha_K)$

Распределение Дирихле

- Распределение тем по документам и слов по темам можно моделировать распределением Дирихле!

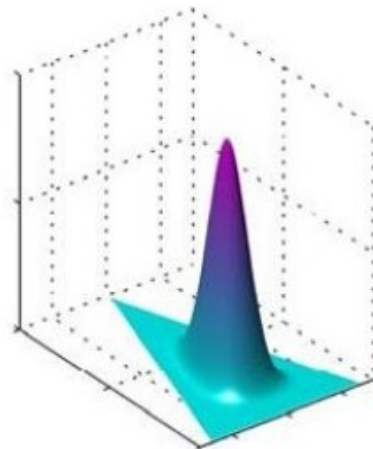
N=3:

Params = [1, 1, 1]



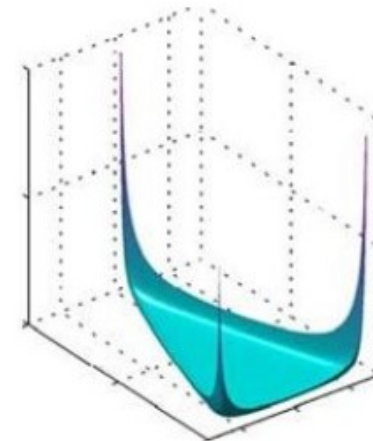
Bigger than 1

Params = [10, 10, 10]



Less than 1

Params = [.1, .1, .1]



Латентное размещение Дирихле

- Upgrade PLSA:
приблизительно оценим (или зададим) среднее число тем в документе K и среднее число ключевых слов в теме V
- Смоделируем распределение тем по документам и слов по темам размещениями Дирихле с параметрами:
 K -мерный вектор α (чем меньше α , тем меньше выраженных тем в документе)
 V -мерный вектор β (чем меньше β , тем меньше слов, характеризующих тему)
- Обычно все координаты векторов α и β берут одинаковыми

Случайный процесс порождения документов в модели LDA

- Дано: количество тем K в документе и слов в теме V , параметры α и β
- Для каждого документа генерируем вероятности тем из распределения Дирихле с параметром α
- Для каждой темы генерируем вероятности слов из распределения Дирихле с параметром β
- Для каждой позиции в документе
 - Выбираем случайно тему согласно сгенерированным вероятностям
 - Выбираем случайно слово, согласно вероятностям слов в теме

Принцип максимума апостериорной вероятности

- Принцип максимума правдоподобия модели $f(x|\theta)$ для случайной величины x :

$$\hat{\theta}_{\text{ML}}(x) = \operatorname{argmax}_{\theta} f(x|\theta)$$

- Если параметр θ – случайная величина с известным априорным распределением $g(\theta)$, то какое θ будет самым вероятным? $\operatorname{Argmax} g(\theta)$? Но нам ведь известна выборка x !
- По формуле Байеса можно вычислить апостериорное распределение θ .
- Принцип максимума апостериорной вероятности:

$$\hat{\theta}_{\text{MAP}}(x) = \operatorname{argmax}_{\theta} \frac{f(x|\theta) g(\theta)}{\int_{\Theta} f(x|\theta') g(\theta') d\theta'} = \operatorname{argmax}_{\theta} f(x|\theta) g(\theta)$$

Принцип максимума апостериорной вероятности

$$\ln \prod_{d \in D} \prod_{w \in d} p(d, w)^{n_{dw}} \prod_{t \in T} \text{Dir}(\phi_t | \beta) \prod_{d \in D} \text{Dir}(\theta_d | \alpha) \rightarrow \max_{\Phi, \Theta}$$

$$\begin{aligned} & \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \phi_{wt} \theta_{td} + \\ & + \sum_{t \in T} \sum_{w \in W} \ln \phi_{wt}^{\beta_w - 1} + \sum_{d \in D} \sum_{t \in T} \ln \theta_{td}^{\alpha_t - 1} \rightarrow \max_{\Phi, \Theta} \end{aligned}$$

Регуляризованный EM-алгоритм

$$\underbrace{\sum_{d,w} n_{dw} \ln \sum_t \phi_{wt} \theta_{td}}_{\text{ln правдоподобия } \mathcal{L}(\Phi, \Theta)} + \underbrace{\sum_{t,w} \tilde{\beta}_w \ln \phi_{wt} + \sum_{d,t} \tilde{\alpha}_t \ln \theta_{td}}_{\text{критерий регуляризации } R(\Phi, \Theta)} \rightarrow \max_{\Phi, \Theta}$$

Если коэффициенты регуляризации > 0 , тогда чем больше логарифмы тем лучше. А когда мы берем α и $\beta < 1$, то получается чем больше нулевых вероятностей, тем лучше!

- В PLSA:

$$\phi_{wt} = \frac{n_{wt}}{n_t} \quad \theta_{td} = \frac{n_{td}}{n_d}$$

- В LDA:

$$\phi_{wt} = \frac{n_{wt} + \beta_w}{n_t + \beta_0} \quad \theta_{td} = \frac{n_{td} + \alpha_t}{n_d + \alpha_0}$$

Контекстные модели

- BagOfWords не учитывает порядок слов => машина не может полностью понять смысл предложений.
- Словосочетания сильно увеличивают словарь и требуют очень большой выборки для обучения
- Разложение документов и слов по темам не позволяет достаточно хорошо понять текст, чтобы, например, перевести его на другой язык
- Если в представлении слова закодировать не темы, а его контекст – машина будет лучше его понимать

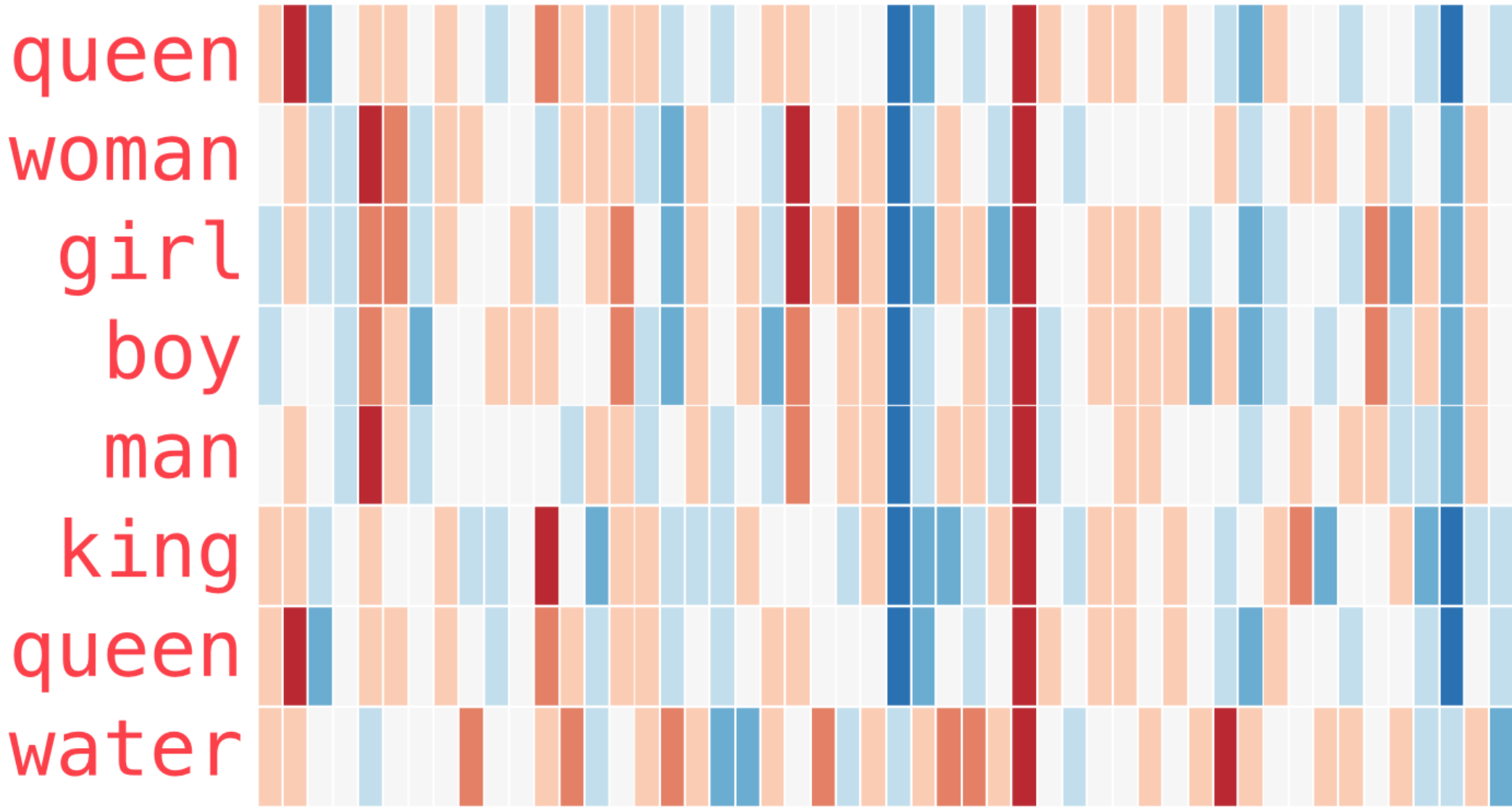
Word2vec

“Я увлекаюсь NLP и я люблю собак”

Составим матрицу встречаемости пар слов

	Я	увлекаюсь	NLP	и	люблю	собак
Я	0	1	0	1	1	0
увлекаюсь	1	0	1	0	0	0
NLP	0	1	0	1	0	0
и	1	0	1	0	0	0
люблю	1	0	0	0	0	1
собак	0	0	0	0	1	0

Word2vec – понижение размерности для матрицы встречаемости



Word2vec

king - man + woman \approx queen

