

# Основы обработки текстов

## Лекция 11

### Тематическое моделирование

# Тематическое моделирование (Topic Modelling)

- Тематическая модель коллекции текстовых документов определяет к каким темам относится каждый документ и какие слова (термины) образуют каждую тему
- Тема - набор терминов, неслучайно часто встречающихся вместе в относительно узком подмножестве документов

# Задача тематического моделирования

- Вход
  - $D$  - коллекция текстовых документов
- Задача
  - Для каждого документа определить к каким темам и в какой степени он принадлежит
  - Для каждого слова определить к каким темам и в какой степени это слово принадлежит
- Задача мягкой кластеризации
- Тематическую модель можно использовать как языковую модель

# Применение

- Кластеризация документов
- Определение близости и рекомендательные системы
  - Определить насколько похожи интересы пользователей Твиттера на основе их постов
- Уменьшение размерности
  - Возможность решать задачу классификации в пространстве меньшей размерности
- Семантический поиск
- Анализ и агрегирование новостных потоков
- Поиск научной информации и фронта исследований

# Основные предположения

- Порядок документов в коллекции не важен
- Порядок слов в документе не важен
- **Предварительная обработка**
  - Лемматизация или стемминг
  - Выделение терминов и словосочетаний
  - Удаление стоп-слов и слишком редких слов

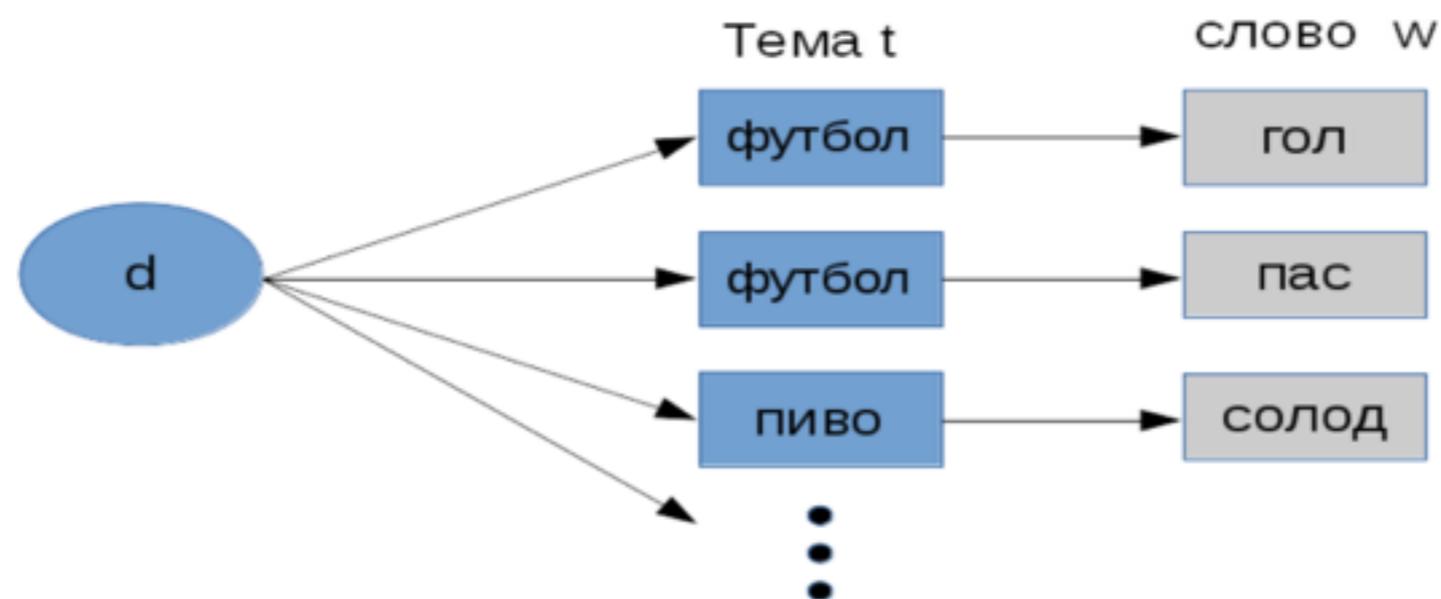
# Вероятностная формализация

- Для каждой темы  $t$  и документа  $d$  зададим вероятность темы в документе  $p(t|d)$
- То же самое сделаем для слов и тем:  
 $p(w|t)$  - вероятность встретить слово  $w$  в теме  $t$
- Предположим что слова в документе зависят только от темы  $p(w|d, t) = p(w|t)$
- Вероятностная модель порождения документа

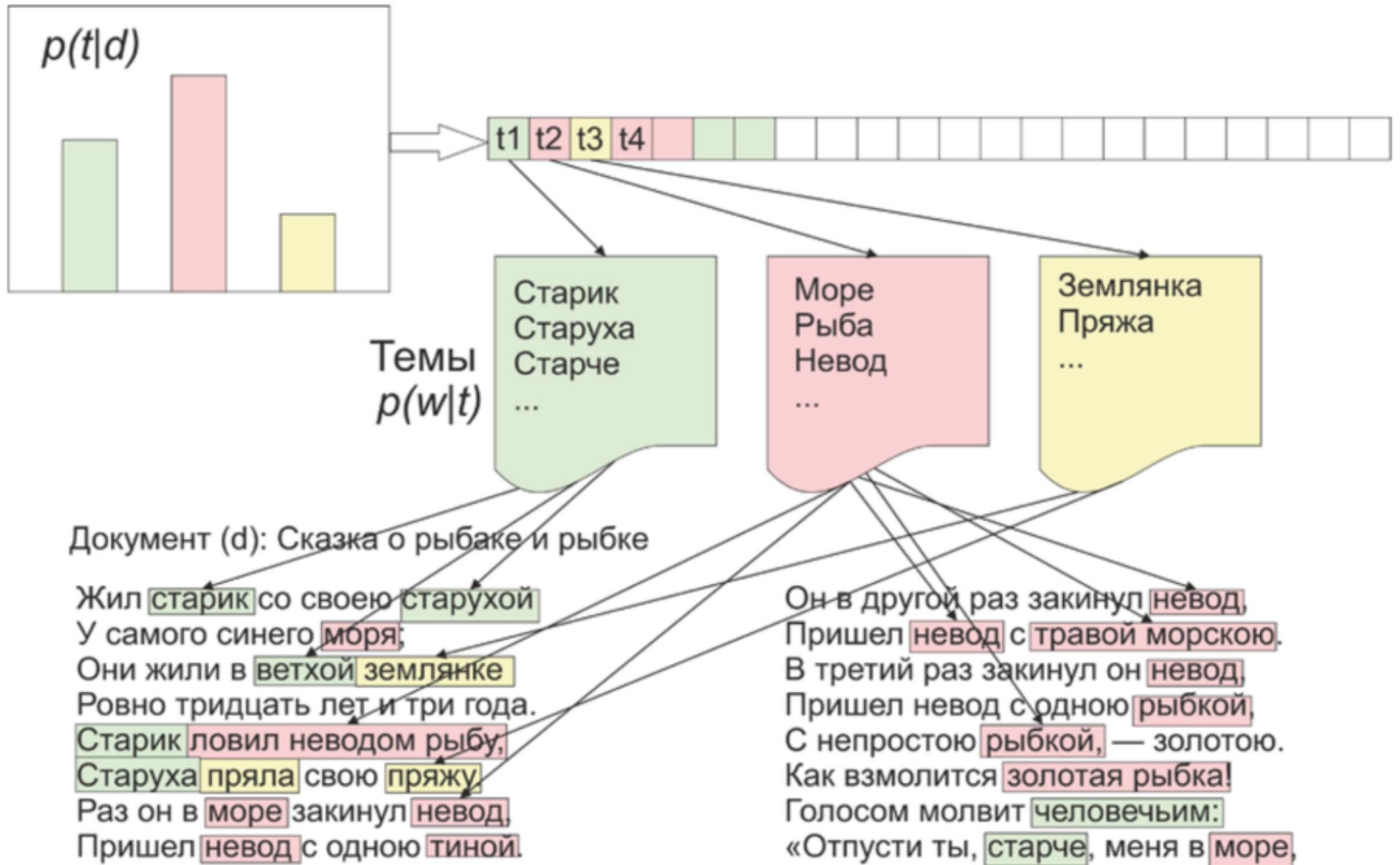
$$p(w|d) = \sum_{t \in T} p(w|d, t)p(t|d) = \sum_{t \in T} p(w|t)p(t|d)$$

# Генерация документов

- Пусть мы хотим сгенерировать документ в 100 слов. Документ написан про футбол (на 0.7), про пиво (на 0.2) и про космические ракеты (на 0.1)
  1. Выбираем тему  $t$  для первого слова (каждая тема  $t$  выбирается с вероятностью  $p(t|d)$ )
  2. Из этой темы выбираем слово  $w$  (слово  $w$  выбирается с вероятностью  $p(w|t)$ )
  3. Повторяем шаги 1 и 2 для остальных 99 слов
- Как видим, слова генерируются независимо друг от друга



# Пример



# Принцип максимума правдоподобия

- **Правдоподобие** - плотность распределения выборки

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

–  $n_{dw}$  - число вхождений термина  $w$  в документ  $d$

- Обозначим

–  $\phi_{wt}$  - распределение терминов по темам

–  $\theta_{td}$  - распределение тем по документам

- **Задача:** найти максимум (логарифма) правдоподобия

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

с ограничениями

$$\forall t \sum_w p(w|t) = 1, \quad \forall d \sum_t p(t|d) = 1$$

$$\forall t, w \quad p(w|t) \geq 0, \quad \forall d, t \quad p(t|d) \geq 0$$

# Некоторые постановки задачи

- Можно не делать предположение об априорном распределении слов по темам и тем по документам (**PLSA**: вероятностный латентный семантический анализ)
- Можно предполагать, что распределения слов по темам и тем по документам получены из распределения Дирихле (**LDA**: скрытое размещение Дирихле)
- Можно учитывать редкие и общие слова (**Robust PLSA**)

# PLSA

- PLSA не делает никаких предположений относительно распределений
- Параметры будем оценивать с помощью EM-алгоритма
  - Оцениваем число слов в документе  $d$ , порожденных темой  $t$
  - Уточняем распределения документов по темам
  - Уточняем распределение тем по словам
- По правилу Байеса

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

# Уточнение распределения тем по документам

- **E-шаг: Оценка числа слов из темы**

- Оцениваем число слов документа  $d$ , порожденных из темы  $t$

$$n_{td} = \sum_w n_{wd} \frac{\phi_{wt} \theta_{td}}{\sum_t \phi_{wt} \theta_{td}}$$

- **M-шаг: Оценка вероятности темы в документе**

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

# Уточнение распределения слов по темам

- **E-шаг: Оценка числа слов из темы**

–Оцениваем число слов в теме  $t$

$$n_{wt} = \sum_d n_{wd} \frac{\phi_{wt} \theta_{td}}{\sum_t \phi_{wt} \theta_{td}}$$

- **M-шаг: Оценка вероятности темы в документе**

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

# Недостатки PLSA

- PLSA переобучается, т.к. число параметров  $\phi_{wt}$  и  $\theta_{td}$  слишком велико  $|D| \cdot |T| + |W| \cdot |T|$
- PLSA не позволяет управлять разреженностью
  - если в начале  $\phi_{wt} = 0$ , то в финале  $\phi_{wt} = 0$
  - если в начале  $\theta_{td} = 0$ , то в финале  $\theta_{td} = 0$
- PLSA неверно оценивает вероятность новых слов: если

$$n_w = 0 \quad \text{то} \quad \hat{p}(w|d) = 0, \quad \forall t \in T$$

# Модели с регуляризацией

- Решение задачи  $L(\Phi, \Theta) \rightarrow \max$  не единственно
- Будем вместе с правдоподобием  $L(\Phi, \Theta)$  максимизировать регуляризатор  $R(\Phi, \Theta)$ , зависящий от параметров модели
- Будем максимизировать сумму двух критериев  $L(\Phi, \Theta) + R(\Phi, \Theta)$  (аддитивная регуляризация тематической модели)

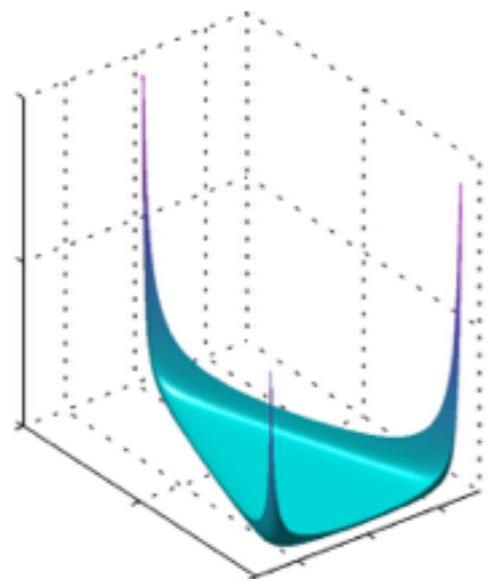
# Модель LDA

- Пусть распределение тем по документам и слов по темам имеет априорное распределение Дирихле (симметричное)

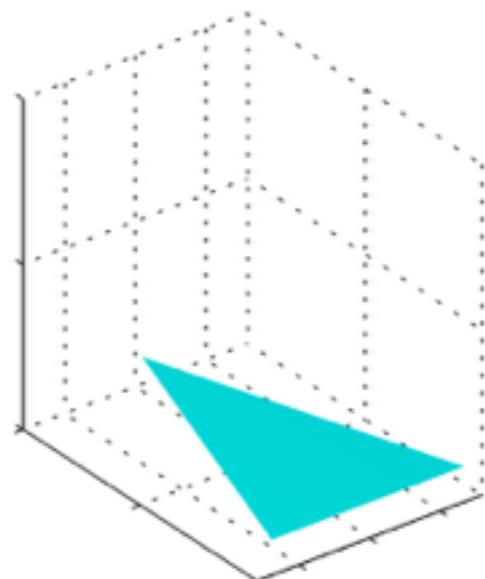
$$f(x_1, x_2, \dots, x_n) = C x_1^{\alpha-1} \times x_2^{\alpha-1} \times \dots \times x_n^{\alpha-1}$$

- Чем больше параметр  $\alpha$ , тем более **сглаженные** распределения будем получать

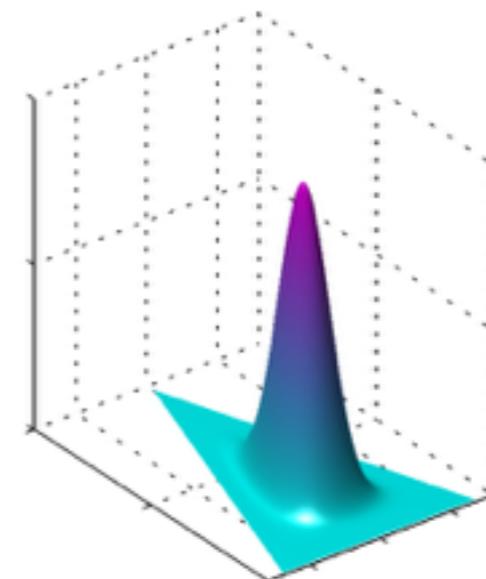
# Распределение Дирихле



$$\alpha_1 = \alpha_2 = \alpha_3 = 0.1$$



$$\alpha_1 = \alpha_2 = \alpha_3 = 1$$



$$\alpha_1 = \alpha_2 = \alpha_3 = 10$$

- Почему именно распределение Дирихле?
  - Математическое удобство
  - Порождает как сглаженные, так и разреженные векторы
  - Неплохо описывает кластерные структуры на симплексе

# Отличие LDA от PLSA

- В PLSA - несмещенные оценки максимума правдоподобия:

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

- В LDA - сглаженные байесовские оценки:

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

# Недостатки LDA

- Слабые лингвистические обоснования “особой роли” распределения Дирихле
- В оригинальном методе сложный вывод параметров (требует интегрирования по пространству параметров модели)
- Сглаживание вместо разреживания
- На практике на достаточно больших данных нет значимых различий между PLSA и LDA

# Robust PLSA

$$p(w|d) = \frac{Z_{dw} + \gamma\pi_{dw} + \epsilon\pi_w}{1 + \gamma + \epsilon}$$

 $Z_{dw}$ 

Тематическая компонента. Совпадает с моделью PLSA. Если она плохо объясняет избыточную частоту слова в документа, то слово относят к фону или шуму

 $\pi_{dw} \equiv p_{noise}(w|d)$ 

Шумовая компонента. Слова специфичные для конкретного документа  $d$ , либо редкие термины, относящиеся к темам, слабо представленным в данной коллекции

 $\pi_w \equiv p_{bgr}(w)$ 

Фоновая компонента. Общеупотребительные слова, в частности, стоп-слова, не отброшенные на стадии предварительной обработки

 $\gamma, \epsilon$ 

Параметры, ограничивающие долю слов в каждой компоненте

# EM-алгоритм для Robust PLSA

$$H_{dwt} = p(t|d, w) = \frac{\phi_{wt}\theta_{td}}{Z_{dw} + \gamma\pi_{dw} + \epsilon\pi_w}$$

$$H_{dw} = \frac{\epsilon\pi_w}{Z_{dw} + \gamma\pi_{dw} + \epsilon\pi_w} \quad H'_{dw} = \frac{\epsilon\pi_w}{Z_{dw} + \gamma\pi_{dw} + \epsilon\pi_w}$$

- M-шаг

$$\pi_{dw} = \frac{\nu_{dw}}{\nu_d},$$

$$\nu_{dw} = n_{dw}H_{dw},$$

$$\nu_d = \sum_{w \in D} \nu_{dw},$$

$$\pi_w = \frac{\nu'_w}{\nu'},$$

$$\nu'_w = \sum_{d \in D} n_{dw}H'_{dw},$$

$$\nu' = \sum_{w \in W} \nu'_w,$$

- E-шаг

$$\pi_{dw} = \left( \frac{n_{dw}}{\nu_d} - \frac{Z_{dw} + \epsilon\pi_w}{\gamma} \right)_+$$

# Реализации тематических моделей

- **Gensim** - реализация для Python
- **BigARTM** - Распределенная реализация PLSA с аддитивной регуляризацией на C
- Topic Modelling в **Spark** - распределенная реализация Robust PLSA с аддитивной регуляризацией на фреймворке Spark (<https://github.com/akopich/dplsa>)

# Пример LDA на

- Википедия, в качестве коллекции

```
>>> lda.print_topics(20)
topic #0: 0.009*river + 0.008*lake + 0.006*island + 0.005*mountain + 0.004*area + 0.004*park + 0.004*antarctic + 0
topic #1: 0.026*relay + 0.026*athletics + 0.025*metres + 0.023*freestyle + 0.022*hurdles + 0.020*ret + 0.017*divis
topic #2: 0.002*were + 0.002*he + 0.002*court + 0.002*his + 0.002*had + 0.002*law + 0.002*government + 0.002*polic
topic #3: 0.040*courcelles + 0.035*centimeters + 0.023*mattythewhite + 0.021*wine + 0.019*stamps + 0.018*oko + 0.0
topic #4: 0.039*al + 0.029*sysop + 0.019*iran + 0.015*pakistan + 0.014*ali + 0.013*arab + 0.010*islamic + 0.010*ar
topic #5: 0.020*copyrighted + 0.020*northamerica + 0.014*uncopyrighted + 0.007*rihanna + 0.005*cloudz + 0.005*know
topic #6: 0.061*israel + 0.056*israeli + 0.030*sockpuppet + 0.025*jerusalem + 0.025*tel + 0.023*aviv + 0.022*pales
topic #7: 0.015*melbourne + 0.014*rovers + 0.013*vfl + 0.012*australian + 0.012*wanderers + 0.011*afl + 0.008*dina
topic #8: 0.011*film + 0.007*her + 0.007*she + 0.004*he + 0.004*series + 0.004*his + 0.004*episode + 0.003*films +
topic #9: 0.019*wrestling + 0.013*château + 0.013*ligue + 0.012*discus + 0.012*estonian + 0.009*uci + 0.008*hockey
topic #10: 0.078*edits + 0.059*notability + 0.035*archived + 0.025*clearer + 0.022*speedy + 0.021*deleted + 0.016*
topic #11: 0.013*admins + 0.009*acid + 0.009*molniya + 0.009*chemical + 0.007*ch + 0.007*chemistry + 0.007*compoun
topic #12: 0.018*india + 0.013*indian + 0.010*tamil + 0.009*singh + 0.008*film + 0.008*temple + 0.006*kumar + 0.00
topic #13: 0.047*bwebs + 0.024*malta + 0.020*hobart + 0.019*basa + 0.019*columella + 0.019*huon + 0.018*tasmania +
topic #14: 0.014*jewish + 0.011*rabbi + 0.008*bgwhite + 0.008*lebanese + 0.007*lebanon + 0.006*homs + 0.005*beirut
topic #15: 0.025*german + 0.020*der + 0.017*von + 0.015*und + 0.014*berlin + 0.012*germany + 0.012*die + 0.010*des
topic #16: 0.003*can + 0.003*system + 0.003*power + 0.003*are + 0.003*energy + 0.002*data + 0.002*be + 0.002*used
topic #17: 0.049*indonesia + 0.042*indonesian + 0.031*malaysia + 0.024*singapore + 0.022*greek + 0.021*jakarta + 0
topic #18: 0.031*stakes + 0.029*webs + 0.018*futsal + 0.014*whitish + 0.013*hyun + 0.012*thoroughbred + 0.012*dnf
topic #19: 0.119*oblast + 0.034*uploaded + 0.034*uploads + 0.033*nordland + 0.025*selsoviet + 0.023*raion + 0.022*
```

- 6 часов 20 минут на MacBook Pro, Intel Core i7 2.3GHz, 16GB DDR3 RAM, OS X

# Для дальнейшего изучения

- *Thomas Hofmann*. Probabilistic latent semantic analysis // Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval. 1999.
- *David M. Blei, Andrew Ng, Michael Jordan*. Latent Dirichlet allocation // Journal of Machine Learning Research (3) 2003 pp. 993-1022.
- Воронцов К. В., Потапенко А. А. Модификации EM-алгоритма для вероятностного тематического моделирования // Машинное обучение и анализ данных. — 2013
- *Коршунов Антон, Гомзин Андрей* Тематическое моделирование текстов на естественном языке // Труды Института системного программирования РАН : журнал. — 2012.
- Воронцов К. В. Лекции по вероятностным тематическим моделям
- Байесовские методы машинного обучения (курс лекций, Д.П. Ветров, Д.А. Кропотов)
- [machinelearning.ru](http://machinelearning.ru)

# Заключение

- Тематические модели являются одним из способов моделирования языка
- Тематические модели являются генеративными моделями: каждый документ определяет темы, а каждая тема определяет слова
- Тематическое моделирование можно рассматривать как задачу мягкой кластеризации документов