# Вероятностные тематические модели
# Лекция 4. Аддитивная регуляризация тематических моделей

К. В. Воронцов
vokov@forecsys.ru

Этот курс доступен на странице вики-ресурса
http://www.MachineLearning.ru/wiki
«Вероятностные тематические модели (курс лекций, К.В.Воронцов)»

ВМК МГУ • весна 2016

# Содержание

# Напоминание. Задача тематического моделирования

**Дано:** $W$ — словарь терминов (слов или словосочетаний),
$D$ — коллекция текстовых документов $d \subset W$,
$n_{dw}$ — сколько раз термин $w$ встретился в документе $d$.

**Найти:** модель $p(w|d) = \sum_{t \in T} \phi_{wt}\theta_{td}$ с параметрами $\underset{W \times T}{\Phi}$ и $\underset{T \times D}{\Theta}$:
$\phi_{wt} = p(w|t)$ — вероятности терминов $w$ в каждой теме $t$,
$\theta_{td} = p(t|d)$ — вероятности тем $t$ в каждом документе $d$.

**Критерий** максимума логарифма правдоподобия:

$$\sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \phi_{wt}\theta_{td} \;\to\; \max_{\phi,\theta};$$

$$\phi_{wt} \geqslant 0; \quad \textstyle\sum_w \phi_{wt} = 1; \qquad \theta_{td} \geqslant 0; \quad \textstyle\sum_t \theta_{td} = 1.$$

**Проблема:** задача стохастического матричного разложения *некорректно поставлена*: $\Phi\Theta = (\Phi S)(S^{-1}\Theta) = \Phi'\Theta'$.

# ARTM — Аддитивная Регуляризация Тематических Моделей

Максимизация ln правдоподобия с регуляризатором $R$:

$$\sum_{d,w} n_{dw} \ln \sum_t \phi_{wt}\theta_{td} + R(\Phi,\Theta) \to \max_{\Phi,\Theta}$$

EM-алгоритм: метод простой итерации для системы уравнений

$$\text{E-шаг:} \quad p_{tdw} = \underset{t\in T}{\text{norm}}\left(\phi_{wt}\theta_{td}\right)$$

$$\text{M-шаг:} \quad \begin{cases} \phi_{wt} = \underset{w\in W}{\text{norm}}\left( n_{wt} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}}\right), & n_{wt} = \sum_{d\in D} n_{dw} p_{tdw} \\ \theta_{td} = \underset{t\in T}{\text{norm}}\left( n_{td} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}\right), & n_{td} = \sum_{w\in d} n_{dw} p_{tdw} \end{cases}$$

PLSA: $R(\Phi,\Theta) = 0$

LDA: $R(\Phi,\Theta) = \sum_{t,w} \beta_w \ln \phi_{wt} + \sum_{d,t} \alpha_t \ln \theta_{td}$

# Условия невырожденности решения

Решение может быть вырожденным для некоторых тем (столбцов матриц $\Phi$) и документов (столбцов матрицы $\Theta$).

*Тема $t$ невырождена*, если хотя бы для одного термина $w \in W$

$$n_{wt} + \phi_{wt} \frac{\partial R}{\partial \phi_{wt}} > 0.$$

Если тема $t$ вырождена, то $p(w|t) = \phi_{wt} \equiv 0$, это означает, что тема исключается из модели (происходит отбор тем).

*Документ $d$ невырожден*, если хотя бы для одной темы $t \in T$

$$n_{td} + \theta_{td} \frac{\partial R}{\partial \theta_{td}} > 0.$$

Если документ $d$ вырожден, то $p(t|d) = \theta_{td} \equiv 0$, это означает, что модель не в состоянии описать данный документ.

# Напоминание. Условия Каруша–Куна–Таккера

Задача математического программирования:

$$\begin{cases} f(x) \to \min\limits_x; \\ g_i(x) \leqslant 0, \quad i = 1, \ldots, m; \\ h_j(x) = 0, \quad j = 1, \ldots, k. \end{cases}$$

Необходимые условия. Если $x$ — точка локального минимума, то существуют множители $\mu_i$, $i = 1, \ldots, m$, $\lambda_j$, $j = 1, \ldots, k$:

$$\begin{cases} \dfrac{\partial \mathscr{L}}{\partial x} = 0, \quad \mathscr{L}(x; \mu, \lambda) = f(x) + \sum\limits_{i=1}^{m} \mu_i g_i(x) + \sum\limits_{j=1}^{k} \lambda_j h_j(x); \\ g_i(x) \leqslant 0; \ h_j(x) = 0; \ \text{(исходные ограничения)} \\ \mu_i \geqslant 0; \ \text{(двойственные ограничения)} \\ \mu_i g_i(x) = 0; \ \text{(условие дополняющей нежёсткости)} \end{cases}$$

# Вывод системы уравнений из условий Каруша–Куна–Таккера

1. Условия ККТ для $\phi_{wt}$ (для $\theta_{td}$ всё аналогично):

$$\sum_d n_{dw} \frac{\theta_{td}}{p(w|d)} + \frac{\partial R}{\partial \phi_{wt}} = \lambda_t - \mu_{wt}; \quad \mu_{wt} \geqslant 0; \quad \mu_{wt}\phi_{wt} = 0.$$

2. Умножим обе части равенства на $\phi_{wt}$ и выделим $p_{tdw}$:

$$\phi_{wt}\lambda_t = \sum_d n_{dw} \frac{\phi_{wt}\theta_{td}}{p(w|d)} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}} = n_{wt} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}}.$$

3. Если $\lambda_t \leqslant 0$, то тема $t$ вырождена, $\phi_{wt} \equiv 0$ для всех $w$.
4. Если $\lambda_t > 0$, то либо $\phi_{wt} = 0$, либо $n_{wt} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}} > 0$:

$$\phi_{wt}\lambda_t = \left(n_{wt} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}}\right)_+.$$

5. Суммируем обе части равенства по $w \in W$:

$$\lambda_t = \sum_{w \in W} \left(n_{wt} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}}\right)_+.$$

6. Подставим $\lambda_t$ из (5) в (4), получим требуемое. ■

# Комбинирование регуляризаторов

Максимизация ln правдоподобия с $k$ регуляризаторами $R_i$:

$$\sum_{d,w} n_{dw} \ln \sum_t \phi_{wt}\theta_{td} + \sum_{i=1}^{k} \tau_i R_i(\Phi,\Theta) \;\to\; \max_{\Phi,\Theta},$$

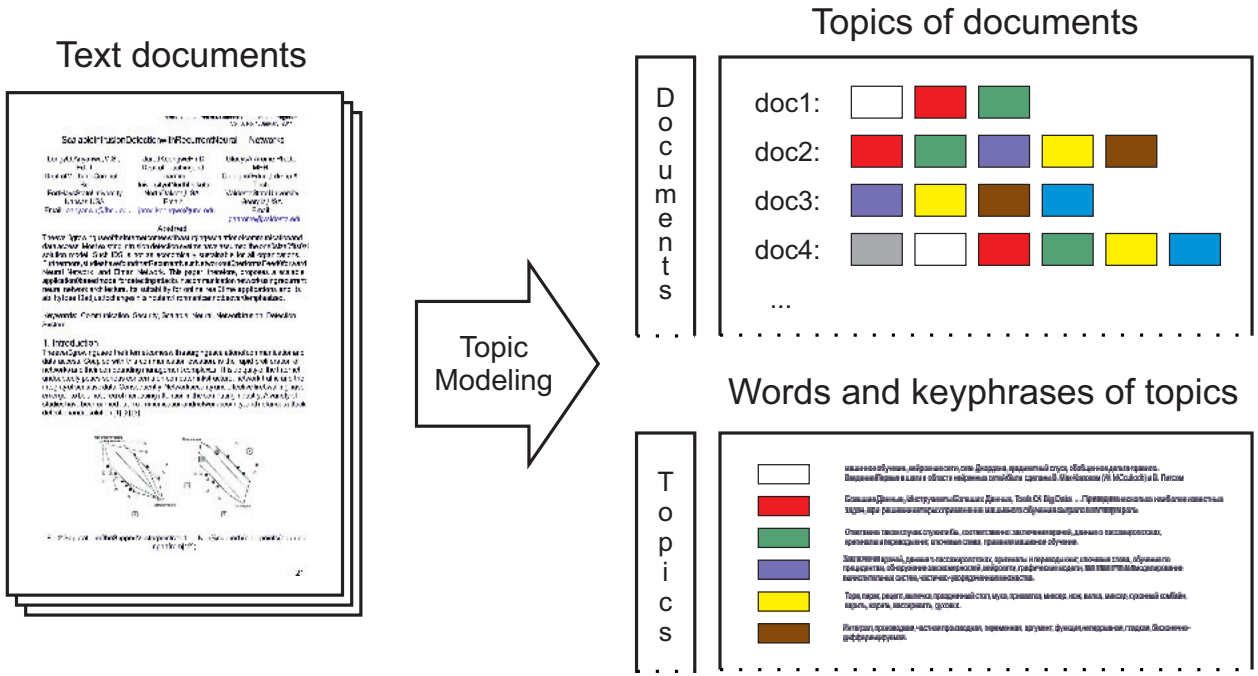
где $\tau_i$ — коэффициенты регуляризации.

EM-алгоритм: метод простой итерации для системы уравнений

$$\begin{cases} \text{E-шаг:} & p_{tdw} = \underset{t\in T}{\text{norm}}\left(\phi_{wt}\theta_{td}\right) \\ \text{M-шаг:} & \phi_{wt} = \underset{w\in W}{\text{norm}}\left(\sum\limits_{d\in D} n_{dw}p_{tdw} + \phi_{wt}\sum\limits_{i=1}^{k}\tau_i\frac{\partial R_i}{\partial \phi_{wt}}\right) \\ & \theta_{td} = \underset{t\in T}{\text{norm}}\left(\sum\limits_{w\in d} n_{dw}p_{tdw} + \theta_{td}\sum\limits_{i=1}^{k}\tau_i\frac{\partial R_i}{\partial \theta_{td}}\right) \end{cases}$$

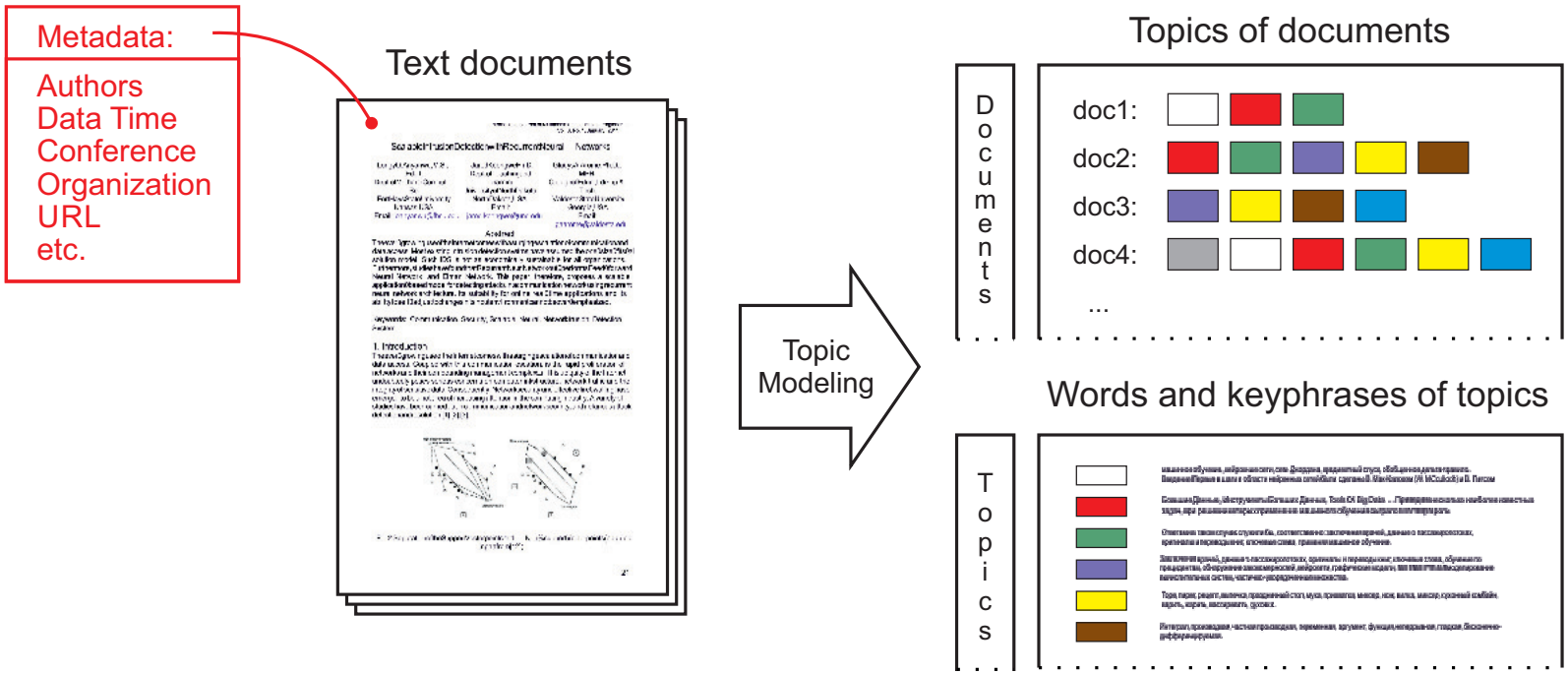
# Мультимодальная тематическая модель

находит тематику документов $p(t|d)$, терминов $p(t|w)$,...



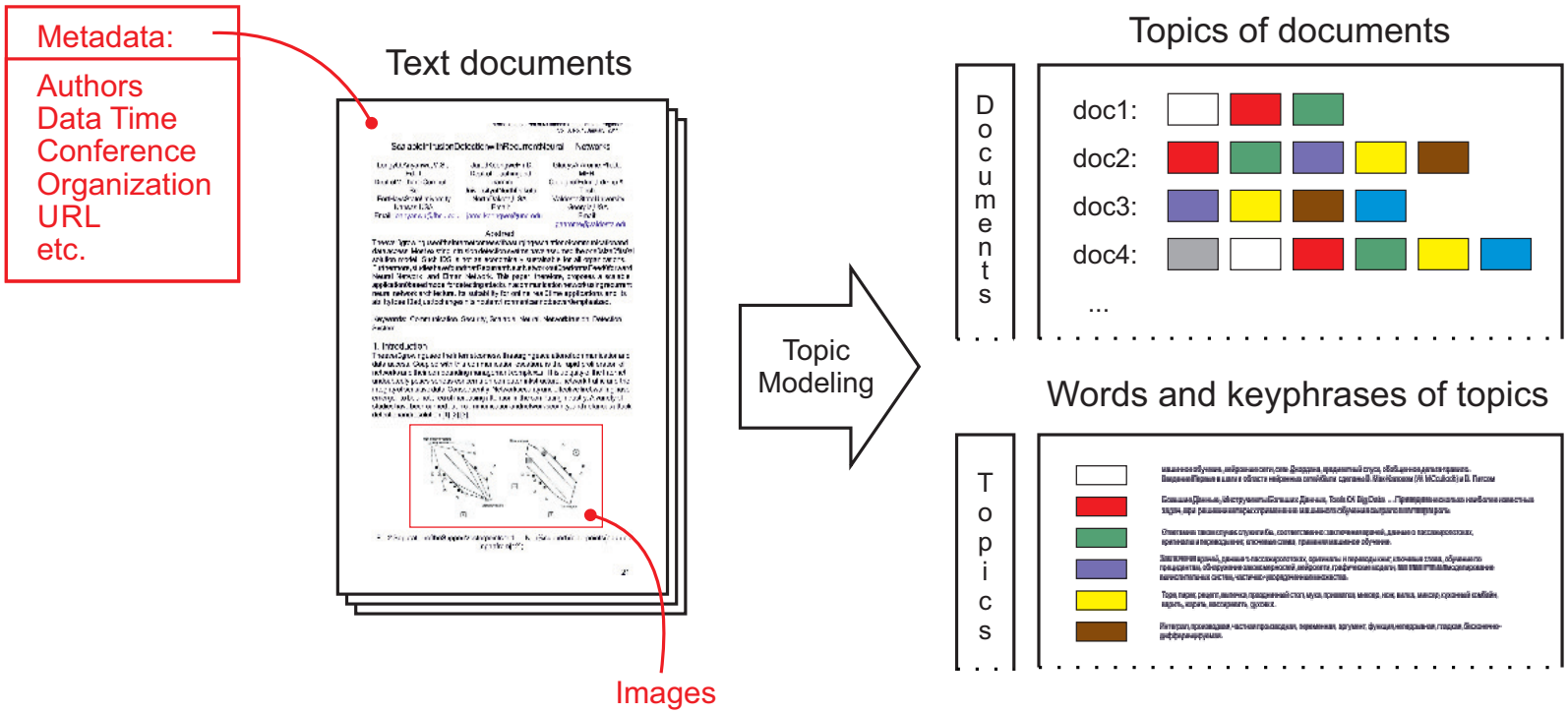
# Мультимодальная тематическая модель

находит тематику документов $p(t|d)$, терминов $p(t|w)$,
авторов $p(t|a)$, времени $p(t|a)$,...



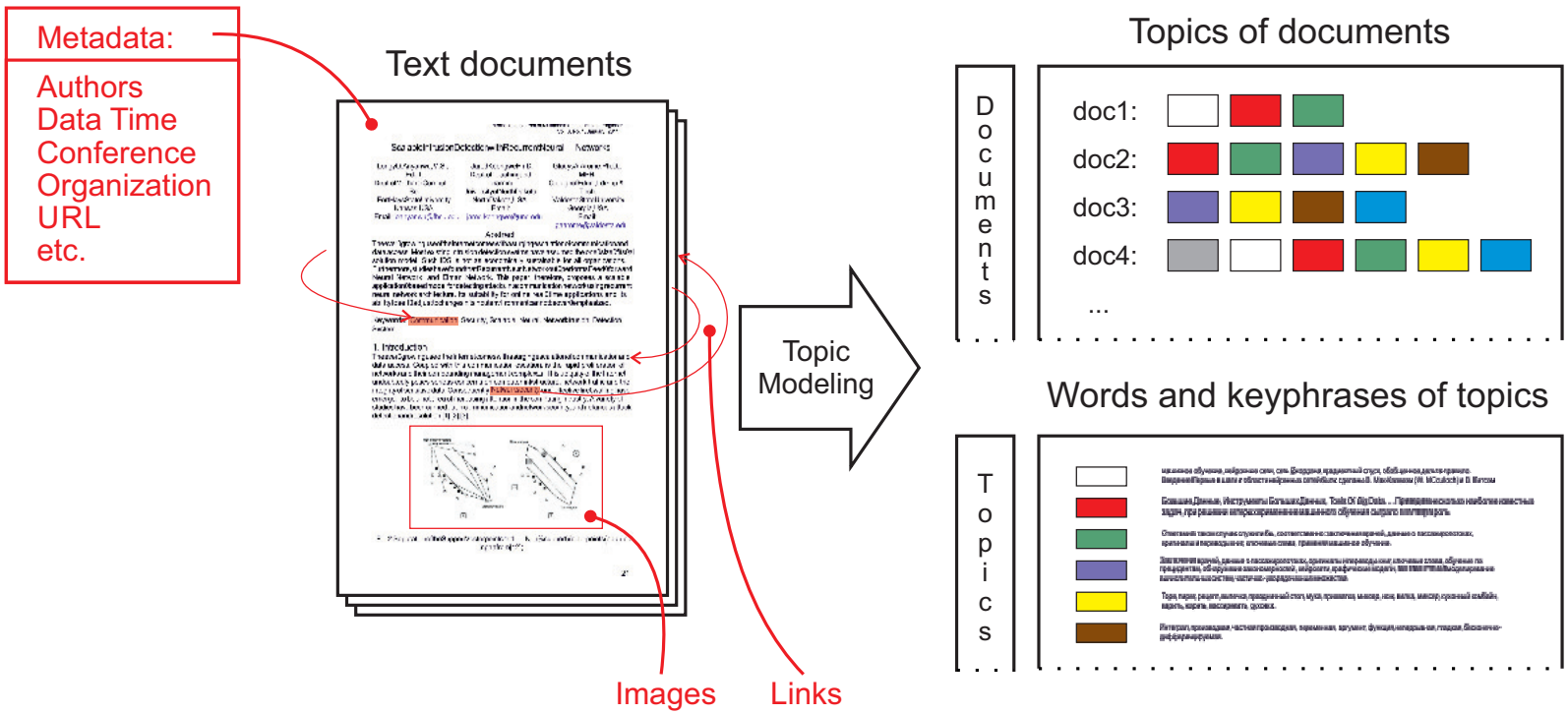
# Мультимодальная тематическая модель

находит тематику документов $p(t|d)$, терминов $p(t|w)$, авторов $p(t|a)$, времени $p(t|a)$, элементов изображений $p(t|e)$,...



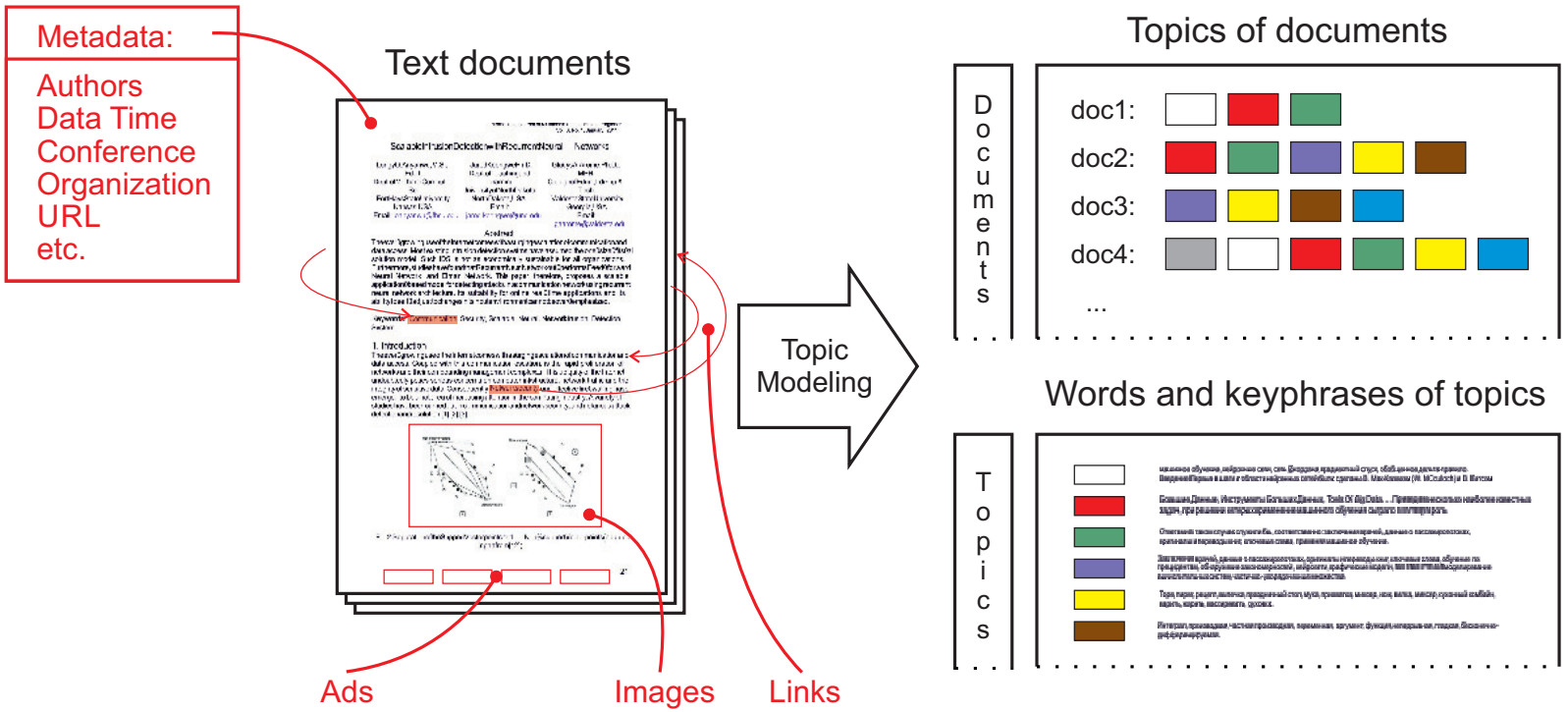
# Мультимодальная тематическая модель

находит тематику документов $p(t|d)$, терминов $p(t|w)$, авторов $p(t|a)$, времени $p(t|a)$, элементов изображений $p(t|e)$, ссылок $p(d'|r)$,...



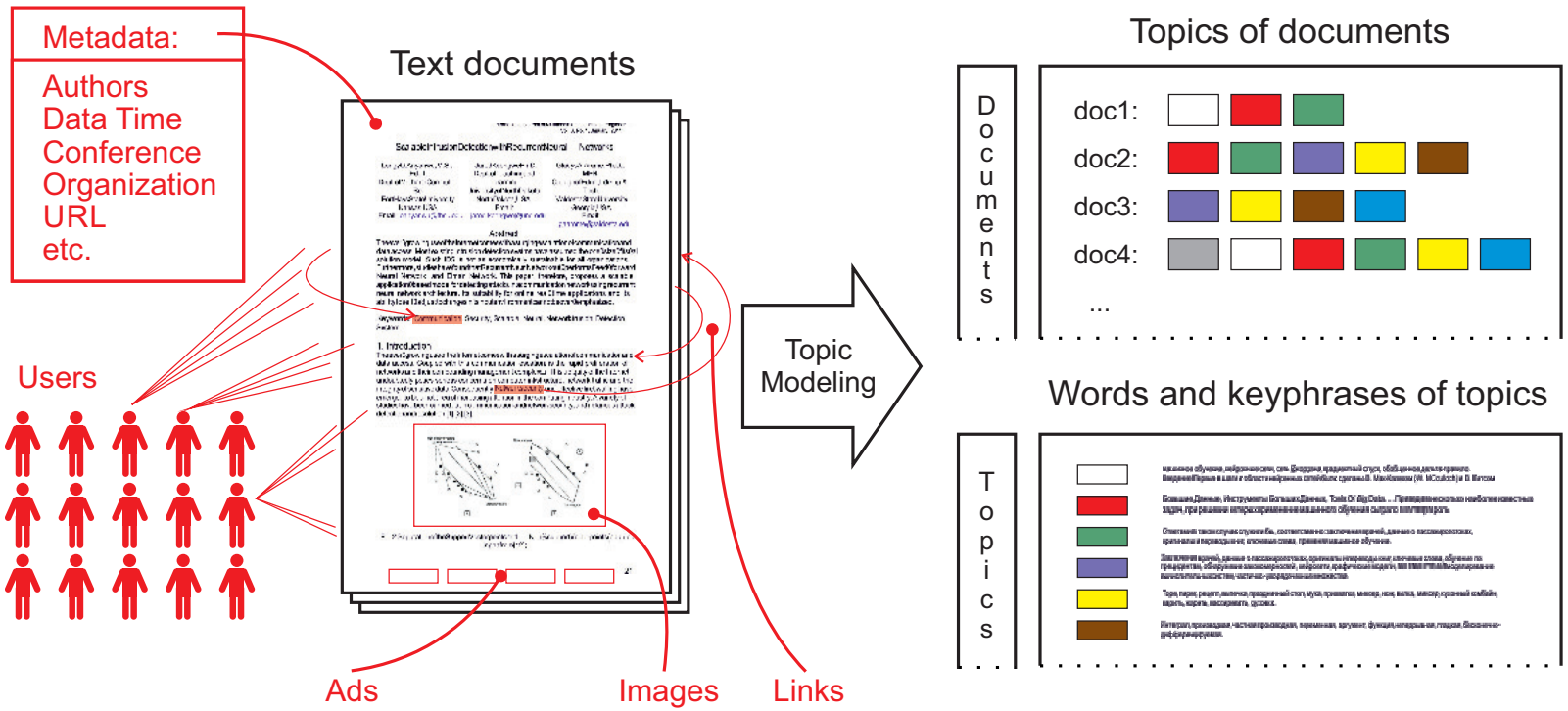
# Мультимодальная тематическая модель

находит тематику документов $p(t|d)$, терминов $p(t|w)$, авторов $p(t|a)$, времени $p(t|a)$, элементов изображений $p(t|e)$, ссылок $p(d'|r)$, баннеров $p(t|b)$,...



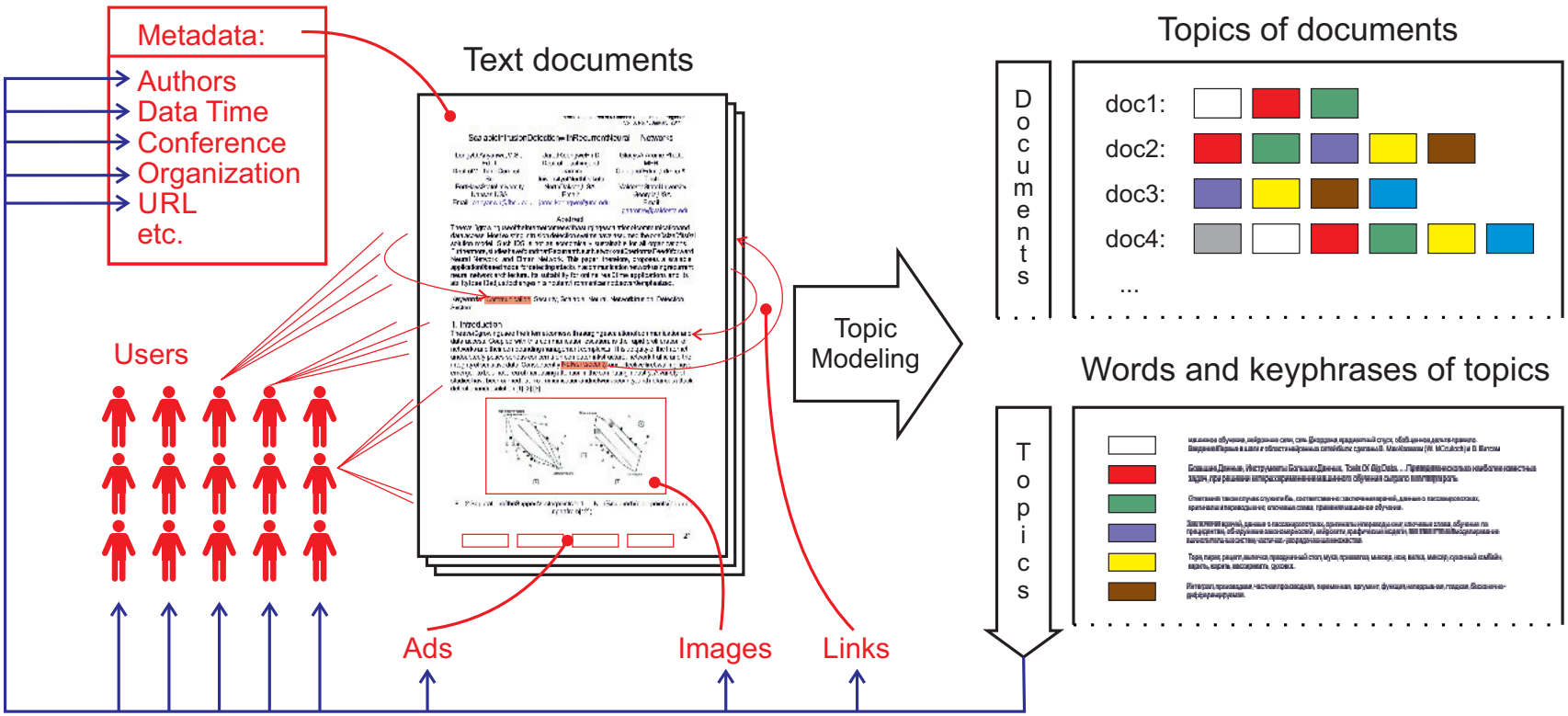
# Мультимодальная тематическая модель

находит тематику документов $p(t|d)$, терминов $p(t|w)$, авторов $p(t|a)$, времени $p(t|a)$, элементов изображений $p(t|e)$, ссылок $p(d'|r)$, баннеров $p(t|b)$, пользователей $p(t|u)$,...



# Мультимодальная тематическая модель

Каждая модальность $m \in M$ описывается своим словарём $W^m$, документы могут содержать *токены* разных модальностей, каждая тема имеет своё распределение $p(w|t)$, $w \in W^m$



# Мультимодальная ARTM

$W^m$ — словарь токенов $m$-й модальности, $m \in M$

$W = W^1 \sqcup \cdots \sqcup W^M$ — объединённый словарь всех модальностей

Максимизация суммы ln правдоподобий с регуляризацией:

$$\sum_{m \in M} \tau_m \sum_{d \in D} \sum_{w \in W^m} n_{dw} \ln \sum_t \phi_{wt} \theta_{td} + R(\Phi, \Theta) \;\to\; \max_{\Phi, \Theta}$$

EM-алгоритм: метод простой итерации для системы уравнений

$$\text{E-шаг:} \quad \begin{cases} p_{tdw} = \underset{t \in T}{\text{norm}}\left(\phi_{wt} \theta_{td}\right) \\ \phi_{wt} = \underset{w \in W^m}{\text{norm}}\left( \sum_{d \in D} \tau_{m(w)} n_{dw} p_{tdw} + \phi_{wt} \frac{\partial R}{\partial \phi_{wt}} \right) \\ \theta_{td} = \underset{t \in T}{\text{norm}}\left( \sum_{w \in d} \tau_{m(w)} n_{dw} p_{tdw} + \theta_{td} \frac{\partial R}{\partial \theta_{td}} \right) \end{cases}$$

# Рациональный (оффлайновый) EM-алгоритм для ARTM

**Идея:** E-шаг встраивается внутрь M-шага

**Вход**: коллекция $D$, число тем $|T|$, число итераций $i_{\max}$;
**Выход**: матрицы терминов тем $\Theta$ и тем документов $\Phi$;

инициализация $\phi_{wt}$, $\theta_{td}$ для всех $d \in D$, $w \in W$, $t \in T$;
**для всех** итераций $i = 1, \ldots, i_{\max}$
  $n_{wt}, n_{td}, n_t, n_d := 0$ для всех $d \in D$, $w \in W$, $t \in T$;
  **для всех** документов $d \in D$ и всех слов $w \in d$
    $p_{tdw} = \underset{t \in T}{\text{norm}}\left(\phi_{wt}\theta_{td}\right)$ для всех $t \in T$;
    $n_{wt}, n_{td}, n_t, n_d \mathrel{+}= n_{dw} p_{tdw}$ для всех $t \in T$;
  $\phi_{wt} := \underset{w \in W}{\text{norm}}\left( \sum_{d \in D} n_{dw} p_{tdw} + \phi_{wt} \frac{\partial R}{\partial \phi_{wt}} \right)$ для всех $w \in W$, $t \in T$;
  $\theta_{td} := \underset{t \in T}{\text{norm}}\left( \sum_{w \in d} n_{dw} p_{tdw} + \theta_{td} \frac{\partial R}{\partial \theta_{td}} \right)$ для всех $d \in D$, $t \in T$;

# Онлайновый параллельный EM-алгоритм для ARTM

**Вход**: коллекция $D$, разбитая на пакеты $D_b$, $b = 1, \ldots, B$;
коэффициент затухания $\rho \in [0, 1]$;
**Выход**: матрица $\Phi$;

инициализировать $\phi_{wt}$ для всех $w \in W$, $t \in T$;
$n_{wt} := 0$, $\tilde{n}_{wt} := 0$ для всех $w \in W$, $t \in T$;
**для всех** пакетов $D_b$, $b = 1, \ldots, B$
  $(\tilde{n}_{wt}) := (\tilde{n}_{wt}) + \textbf{ProcessBatch}(D_b, \Phi)$;
  $n_b := \sum_{d \in D_b} n_d$ — длина пакета $D_b$ в словах;
  **если** пора выполнить синхронизацию, **то**
    $n_{wt} := (1 - \rho) n_{wt} + \rho \frac{n}{n_b} \tilde{n}_{wt}$ для всех $w \in W$, $t \in T$;
    $\phi_{wt} := \underset{w \in W}{\operatorname{norm}} \left( n_{wt} + \phi_{wt} \frac{\partial R}{\partial \phi_{wt}} \right)$ для всех $w \in W$, $t \in T$;
    $\tilde{n}_{wt} := 0$ для всех $w \in W$, $t \in T$;

# Онлайновый параллельный EM-алгоритм для ARTM

**ProcessBatch** обрабатывает пакет $D_b$ при фиксированной $\Phi$.

**Вход**: пакет $D_b$, матрица $\Phi = (\phi_{wt})$;
**Выход**: матрица $(\tilde{n}_{wt})$;

$\tilde{n}_{wt} := 0$ для всех $w \in W,\ t \in T$;
**для всех** $d \in D_b$
  инициализировать $\theta_{td} := \frac{1}{|T|}$ для всех $t \in T$;
  **повторять**
    $p_{tdw} := \underset{t \in T}{\text{norm}}\left(\phi_{wt}\theta_{td}\right)$ для всех $w \in d,\ t \in T$;
    $\theta_{td} := \underset{t \in T}{\text{norm}}\left(\sum_{w \in d} n_{dw} p_{tdw} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}\right)$ для всех $t \in T$;
  **пока** $\theta_d$ не сойдётся;
  $\tilde{n}_{wt} := \tilde{n}_{wt} + n_{dw} p_{tdw}$ для всех $w \in d,\ t \in T$;

# Основные особенности библиотеки BigARTM

- *Теоретическая основа:* ARTM
- *Открытый код:* `http://bigartm.org`
- *Онлайновый алгоритм:* потоковая обработка данных
- *Параллельная реализация:* одна из самых быстрых
- *Кроссплатформенность:* Windows, Linux, Mac OS
- *Регуляризаторы:* наращиваемая библиотека
- *Функционалы качества:* наращиваемая библиотека
- *Пользовательские API:* C++, Python, ...

# Компоненты библиотеки

- **Ядро библиотеки** — параллельная (многопоточная) реализация онлайнового EM-алгоритма
- **Регуляризаторы** — плагины, которые можно добавлять к библиотеке без структурных изменений ядра
- **Функционалы** — плагины, во многом похожи на регуляризаторы
- **Словари** — внешние объекты с различными дополнительными данными о коллекции
- **Парсер** — преобразование входных данных в форматы BigARTM: UCI-BoW, Vowpal Wabbit, plain text.

# Регуляризаторы

- Сглаживание/разреживание/частичное обучение $\Phi$
- Сглаживание/разреживание/частичное обучение $\Theta$
- Декорреляция тем в $\Phi$
- Заданное разреживание $\Phi$
- Балансирование классов в $\Phi$
- Повышение когерентности тем в $\Phi$
- Отбор тем в $\Theta$

# Функционалы качества

- Перплексия
- Разреженность $\Phi$
- Разреженность $\Theta$
- Характеристики ядер тем (чистота, контрастность)
- Самые вероятные слова
- Срез матрицы $\Theta$
- Число обработанных документов
- Доля фоновых тем в модели

# Установка

Инструкция по установке библиотеки можно найти здесь:

http://bigartm.readthedocs.org/en/stable/installation/index.html

Все вопросы отправлять сюда (на английском):

bigartm-users@googlegroups.com

**Внимание:** в инструкциях ошибок нет!
Если что-то не работает, стоит сперва проверить пути к файлам, переменные окружения, разок попробовать перезагрузить компьютер. Больше половины проблем с установкой на этом благополучно разрешаются.

**Пример** использования библиотеки в простом эксперименте:

https://github.com/bigartm/bigartm-book/blob/master/BigARTM_example_RU.ipynb

# Подготовка данных

По аналогии с sklearn входные данные — объект `BatchVectorizer`:

```
BatchVectorizer(batches=None,
                collection_name=None,
                data_path='',
                data_format='batches',
                target_folder='',
                batch_size=1000):
```

**Какие могут быть входные данные:**

1. готовые пакеты
   - `batches` = путь к директории с пакетами

# Подготовка данных

## Какие могут быть входные данные:

2. мешок слов в формате UCI-Bow
   - `collection_name` = имя коллекции по названию файла,
   - `data_path` = путь к директории с данными,
   - `target_folder` = путь к директории пакетов и словаря,
   - `data_format = 'bow_uci'`
3. текст в формате Vowpal Wabbit
   - `data_path` = путь к директории с данными,
   - `target_folder` = путь к директории пакетов и словаря,
   - `data_format = 'vowpal_wabbit'`

## Пример:

```
batch_vectorizer = artm.BatchVectorizer( \
    data_path='', data_format='bow_uci', \
    collection_name='kos', target_folder='kos')
```

# Создание модели

```
ARTM(num_processors=0, num_topics=10, \
     topic_names=None, class_ids=None, \
     cache_theta=True, num_document_passes=1)
```

`num_processors` — число потоков-обработчиков

`num_topics` — число тем (вариант попроще)

`topic_names` — список с именами тем (вариант посложнее)

`class_ids` — dict, ключ – имя модальности, значение – вес

`cache_theta` — флаг кэширования $\Theta$

`num_document_passes` — число итераций по документу

**Пример:**

```
topic_names = ['topic_name_{}'.format(n) for n in xrange(15)]
model = artm.ARTM(num_processors=4, topic_names=topic_names)
```

# Словари

**Словарь** — объект, содержащий информацию о терминах коллекции, их частоты и другие полезные данные.

Случаи использования словаря:

- при инициализации модели;
- в функционалах качества;
- в регуляризаторах.

Словарь собирается по пакетам с документами:

```
model.gather_dictionary( \
           dictionary_name='dictionary', \
           dictionary_path=batch_vectorizer.data_path)
```

# Построение модели PLSA

Теперь инициализируем модель, используя собранный словарь:

```
ARTM.initialize(dictionary_name='dictionary')
```

Запустим 10 итераций оффлайн-алгоритма PLSA
на созданных пакетах с документами:

```
model.fit_offline(batch_vectorizer=batch_vectorizer, \
                  num_collection_passes=10)
```

# Добавим функционал качества и регуляризатор

Добавим функционал качества «разрженность матрицы Φ»:

```
model_plsa.scores.add( \
        artm.SparsityPhiScore(name='SparsityPhiScore'))
```

и регуляризатор разреживания Θ:

```
model_artm.regularizers.add( \
  artm.SmoothSparseThetaRegularizer(name='SparseTheta', \
                                    tau=-0.1))
```

Запустим ещё 10 итераций:

```
model.fit_offline(batch_vectorizer=batch_vectorizer, \
                  num_collection_passes=10)
```

Посмотрим на финальное значение разреженности:

```
print model.score_tracker['SparsityPhiScore'].last_value
```

# Регуляризаторы

У каждого регуляризатора есть свои параметры (см. док-строки `bigartm/python/artm/regularizers.py`)

**Пример:** регуляризатор сглаживания/разреживания Φ:

- `name` — имя регуляризатора, строка
- `tau` — коэффициент регуляризации, вещественное число
- `topic_names` — список имён регуляризуемых тем, список строк
- `class_ids` — список имён регуляризуемых модальностей, список строк
- `dictionary_name` — имя словаря, который нужен регуляризатору, строка

Все параметры опциональные, все можно поменять в любой момент между итерациями обучения модели.

# Функционалы качества

У каждого функционала есть свои параметры
(см. док-строки `bigartm/python/artm/scores.py`)

**Пример:** функционал самых вероятных слов в каждой теме:

- `name` — имя функционала, строка
- `topic_names` — список имён оцениваемых тем, список строк
- `class_id` — имя оцениваемой модальности, строка
- `num_tokens` — максимальное число искомых топ-токенов, целое число
- `dictionary_name` — имя словаря, в данном случае с информацией о совместной встречаемости слов для подсчёта когерентности по топ-токенам, строка

Все параметры опциональные, все можно поменять в любой момент между итерациями обучения модели.

# Алгоритм обучения

## Оффлайн EM-алгоритм:

1. многократное итерирование коллекции
2. однократный проход по документу
3. хранение матрицы $\Theta$
4. обновление $\Phi$ в конце каждого прохода по коллекции
5. применяется при обработке небольших коллекций

## Онлайн EM-алгоритм:

1. однократный проход по коллекции
2. многократное итерирование документа
3. нет необходимости хранить матрицу $\Theta$
4. обновление $\Phi$ через заданное число пакетов
5. применяется при потоковой обработке больших коллекций

# Онлайновый алгоритм. Постановка эксперимента

- **Цель эксперимента:** исследовать зависимость качества модели от параметров онлайнового EM-алгоритма
- **Исходные данные:**
  - коллекция, сообщений социальной сети VK
  - коллекция статей википедии
  - коллекция статей arXiv.org
- **Параметры эксперимента** (онлайнового алгоритма):
  - `decay_weight` $= (1 - \rho)$, $\rho$ — коэффициент затухания
  - `apply_weight` $= \rho \frac{n}{n_d}$
  - `batch_size` — число документов в каждом пакете
  - `update_every` — через сколько пакетов обновлять $\Phi$
  - `num_document_passes` — число итераций каждого документа
- **Представление результатов:**
  - зависимости перплексии и других функционалов качества от параметров эксперимента

# Тематическая сегментация. Постановка эксперимента

- **Цель эксперимента:** проверить способность ARTM восстанавливать тематическую сегментацию текста
- **Исходные данные:**
  - коллекция, генерируемая по заданным матрицам $\Phi$, $\Theta$
  - каждый документ — серия монотематичных сегментов
- **Параметры эксперимента:**
  - средняя длина документа $\bar{n}_d$
  - средняя длина сегмента $\bar{n}_s$
  - доля фоновых слов (в первом эксперименте 0)
- **Идея:** пост-обработка $p(t|d,w)$ на E-шаге
- **Представление результатов:**
  - зависимость доли ошибок восстановления $p(t|d,w)$ от параметров эксперимента

# Порождающая тематическая модель сегментированного текстов

Процесс порождения документов $d$, состоящих из сегментов $s$:

**Вход**: распределения $\phi_{wt} = p(w|t)$ и $\theta_{td} = p(t|d)$;
длины документов $n_d$; средняя длина сегментов $m$;

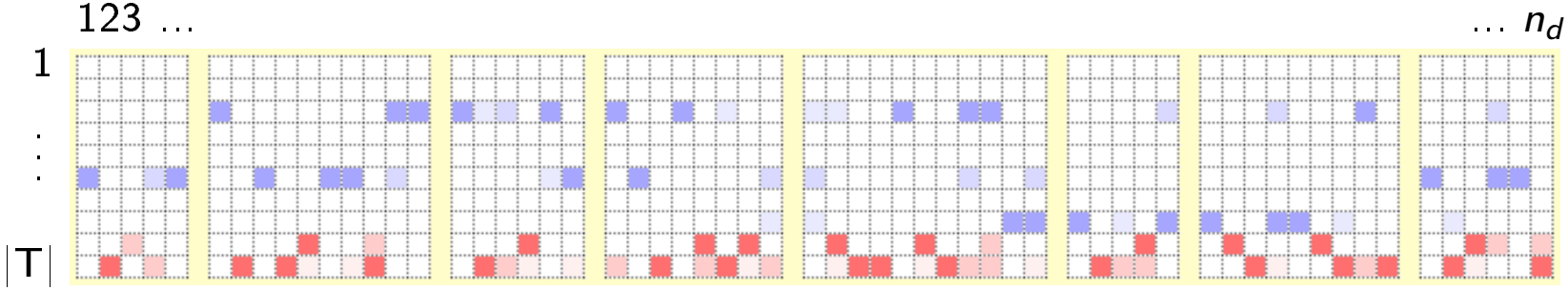
**Выход**: коллекция документов;

**для всех** документов $d \in D$
- $k_t := \lceil n_d \theta_{td} / m \rceil$ — число сегментов для каждой $t \in T$;
- $k := \sum_t k_t$ — суммарное число сегментов всех тем;
- $i := 1$ — текущая позиция в порождаемом документе $d$;
- **для всех** порождаемых сегментов $j = 1, \ldots, k$
  - выбрать тему сегмента $t_j$ из $p(t|d)$;
  - выбрать длину сегмента $n_j \approx n_d \theta_{t_j d} / k_{t_j}$;
  - **для всех** порождаемых $n_j$ слов
    - выбрать слово $w_i$ из $p(w|t_j)$; $i{+}{+}$;

# Гипотеза о тематической сегментации текста

Документ $d = \{w_1, \ldots, w_{n_d}\}$, $n_d$ — длина документа $d$

Матрица *тематических профилей слов* $p(t|d, w_i)$ размера $T \times n_d$:




Предположения разреженности и непрерывности тематики:

- текст разбивается на тематически однородные сегменты
- каждый сегмент относится к 1–2 предметным темам
- слова общей лексики не влияют на тематику сегмента
- соседние сегменты часто имеют общие темы

# Аннотирование и поиск. Постановка эксперимента

- **Цель эксперимента:** оценить согласованность автоматического аннотирования и тематического поиска
- **Исходные данные:**
  - коллекция статей научных конференций ММРО/ИОИ
  - коллекция статей arXiv.org
- **Параметры эксперимента:**
  - параметры оценки тематической значимости фраз
  - длина аннотации
  - мера тематической близости документов
- **Идея:** согласованность аннотирования и поиска должна означать высокое качество реализации обеих функций
- **Представление результатов:**
  - зависимость качества поиска (номер исходного документа в списке выдачи) от параметров эксперимента

# Алгоритм формирования аннотации документа

Документ $d$ — последовательность фраз $s_j$, $j = 1, \ldots, k_d$.
Фраза $s$ — последовательность терминов $w_{si}$, $i = 1, \ldots, n_s$.

**Вход**: тематическая модель $\phi_{wt} = p(w|t)$ и $\theta_{td} = p(t|d)$;
документ $d$; распределение $p(t|d, w_{si})$; бюджет $m$;
**Выход**: аннотация — набор фраз из $d$ суммарной длины $\leqslant m$;

**для всех** фраз $s$ документа $d$
- $Q_s := \max\limits_{t \in T} \sum_i p(w_{si}|t)$ — оценка ценности фразы $s$;
- $p(t|s) := \text{avr}_i\, p(t|d, w_{si})$ — тематический вектор фразы $s$;

найти подмножество фраз $S = \{s_j\}$ такое, что:

$$\begin{cases} \sum_s n_s \leqslant m & \text{— длина в рамках бюджета} \\ \sum_s Q_s \to \max & \text{— ценность максимальна} \\ \rho\big(\text{avr}_s\, p(t|s), p(t|d)\big) \to \min & \text{— релевантность максимальна} \end{cases}$$

# Резюме

- Задача тематического моделирования некорректно поставлена, её решение не единственно и не устойчиво.
- Регуляризация — стандартный приём решения таких задач.
- Подход ARTM позволяет комбинировать регуляризаторы, строить тематические модели с требуемыми свойствами, иметь общую для всех моделей реализацию.
- Модель LDA — частный случай регуляризатора сглаживания.
- Теория ARTM реализована в проекте BigARTM.
- Регуляризаторы и модальности — в следующих лекциях.