Мастер-класс по тематическому моделированию

Мурат Апишев
great-mel@yandex.ru

МГУ им. М. В. Ломоносова, Яндекс, ШАД

17 марта, 2017

Содержание занятия

1. **Теоретическое введение**
   - Приложения TM
   - TM как матричное разложение. Модель PLSA
   - APTM
   - M-APTM
2. **Библиотека BigARTM**
   - Онлайновый EM-алгоритм
   - Основные алгоритмические возможности
   - Пользовательские интерфейсы
3. **Пример реального эксперимента**
   - Подготовка эксперимента
   - Проведение эксперимента
   - Оценивание результатов
4. **Домашнее задание**

Тематическое моделирование

Тематическое моделирование (*Topic Modeling*) — приложение машинного обучения к статистическому анализу текстов.

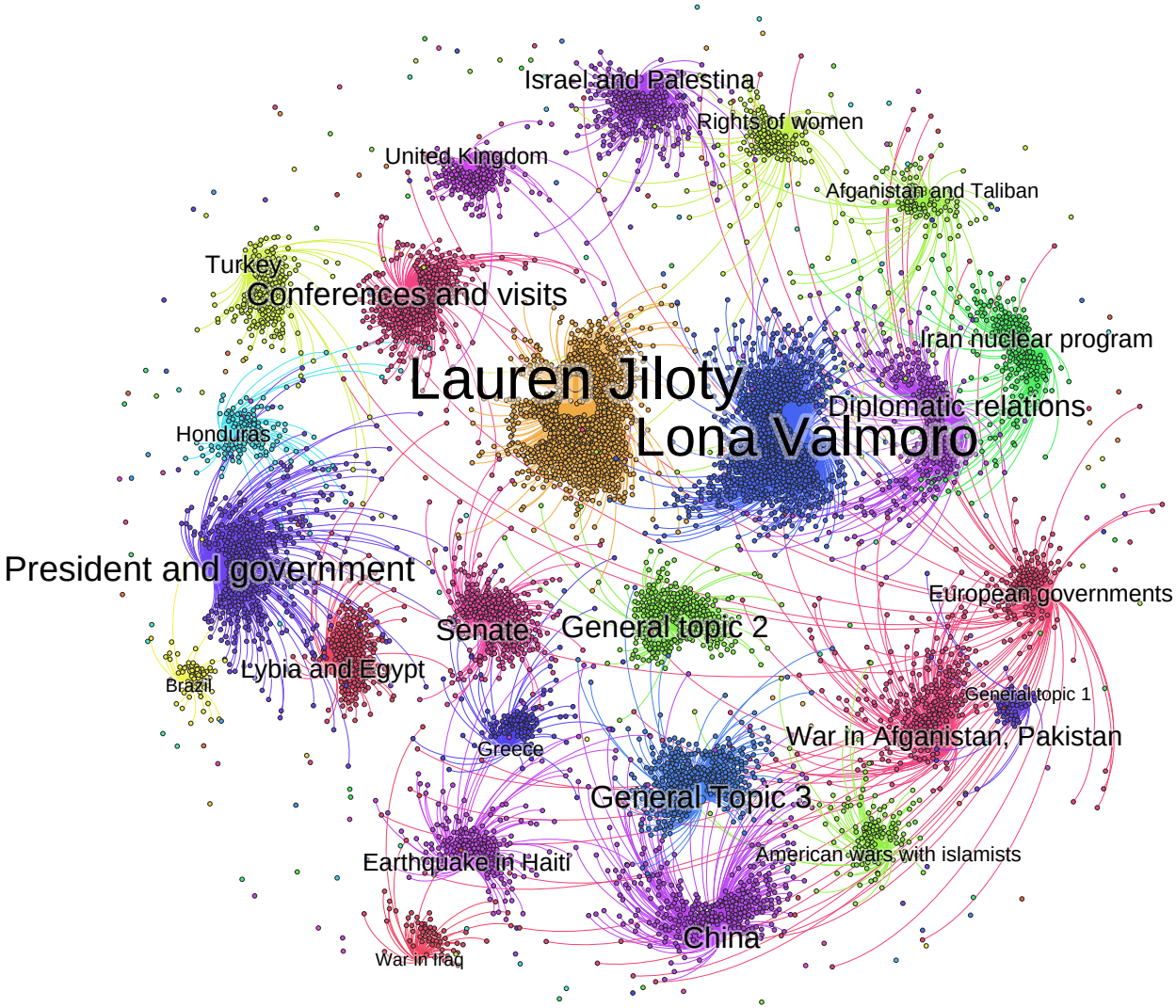
Тема — терминология предметной области, набор терминов (униграм или $n-$грам) часто встречающихся вместе в документах.

Тематическая модель исследует скрытую тематическую структуру коллекции текстов:

- *тема* t — это вероятностное распределение $p(w|t)$ над терминами w
- *документ* d — это вероятностное распределение $p(t|d)$ над темами t

Нестрого говоря, тема — это набор слов, глядя на которые можно сказать, какую предметную область они описывают.

Кластеризация и классификация документов



Анализ социальных медиа

- Извлечение из корпуса сообщений социальных медиа информации об обсуждаемых там темах.
- Исследование значимости тех или иных тем, определение интересов аудитории
- Выделение методами TM специфических тем из интересующей предметной области:
  - Национальные/этнические вопросы
  - Разного рода незаконная деятельность
  - Политические и экономические проблемы
- Отслеживание распространённости интересных тем в пространстве и времени

Информационный поиск

Информационный (разведочный) поиск — парадигма поиска, отличная от современных поисковых систем, направленная на изучение предметной области, связанной с запросом.

Поисковый запрос — документ или коллекция документов.

Поисковая потребность — информация о предметной области.

Концепция особенно полезна в ситуации, когда неясно, каким образом задавать короткий поисковый запрос:

- отсутствие подходящих слов
- проблема омонимии

Особенно полезным тематический поиск может быть для научного и инженерного сообществ.

Мешок слов

Мешок слов (Bag-Of-Words) — представление текстовых данных, в котором учитывается только частота встречаемости слов в документах. Порядок слов игнорируется.

Исходное предложение: `I can drink a milk can`

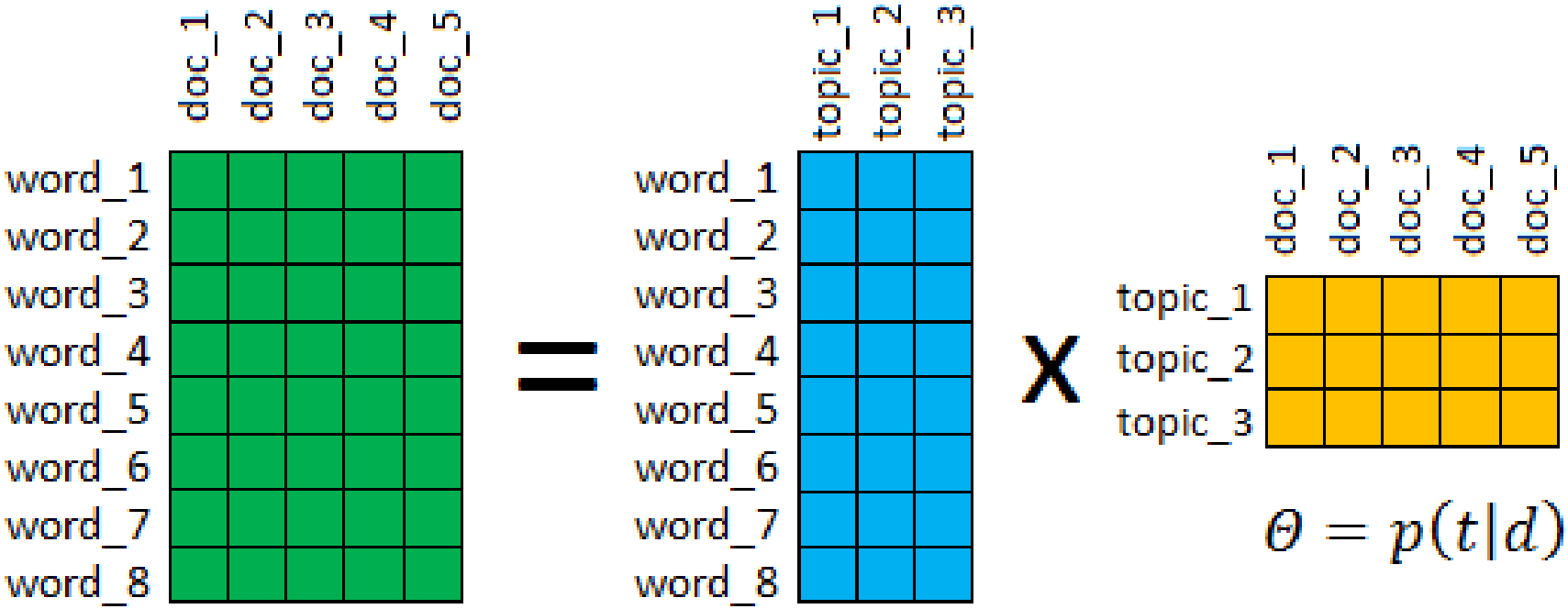
Его мешок слов:

```
I: 1
can: 2
drink: 1
a: 1
milk: 1
```

Проще, но теряется много полезной информации.

Матричное разложение

Можно представить данные в виде матрицы, а задачу — в виде **матричного разложения**:



Формальная постановка задачи

Дано: W — словарь терминов (униграм или $n-$биграм),
D — коллекция текстовых документов $d \subset W$,
n_{dw} — счётчик частоты появления слова w 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 максимизация логарифма правдоподобия:

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

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

PLSA и EM-алгоритм

Максимизация логарифма правдоподобия:

$$\sum_{d\in D}\sum_{w\in d} n_{dw} \ln \sum_t \phi_{wt}\theta_{td} \;\to\; \max_{\Phi,\Theta}$$

EM-алгоритм: метод простых итерация для решения системы уравнений

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

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

где $\underset{i\in I}{\text{norm}}\, x_i = \dfrac{\max\{x_i,0\}}{\sum\limits_{j\in I}\max\{x_j,0\}}$

Метрики качества и недостатки PLSA

Величина, характеризующая степень сходимости модели с заданным словарём W — *перплексия*:

$$\mathscr{P}(D) = \exp\left(-\frac{1}{n}\sum_{d\in D}\sum_{w\in d} n_{dw} \ln \sum_{t\in T} \phi_{wt}\theta_{td}\right), \quad n = \sum_d n_d.$$

Она построена на основе логарифма правдоподобия и характеризует степень качества описания коллекции моделью. Чем ниже — тем лучше. Алгоритм оптимизирует именно её.

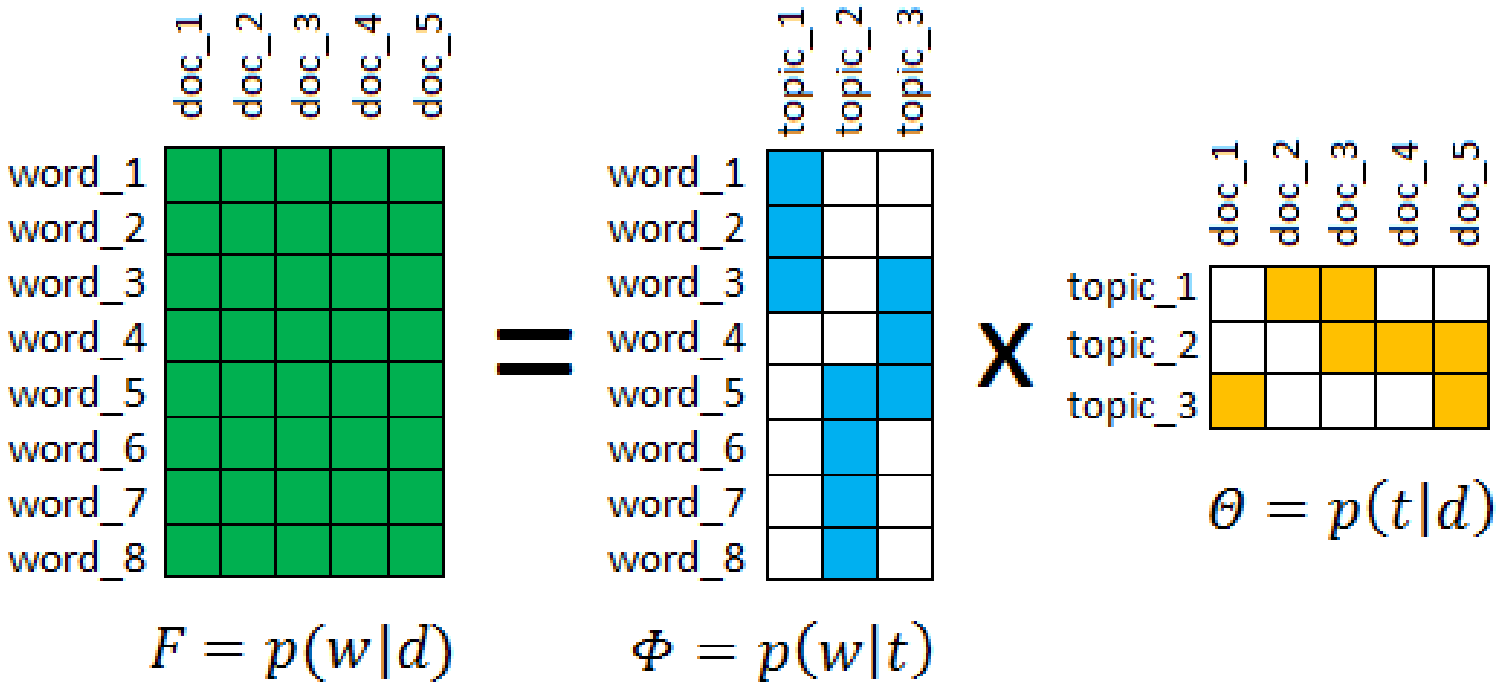
Но! Матричное разложение $F \approx \Phi \times \Theta$ имеет бесконечное множество решений: $\Phi\Theta = (\Phi S)(S^{-1}\Theta) = \Phi'\Theta' \Rightarrow$ можно подобрать Φ и Θ подходящего вида.

Существуют различные прикладные метрики качества. Они не оптимизируются моделью напрямую, но их значения хочется повышать. **Выход — регуляризация!**

Пример

Логичные предположения:

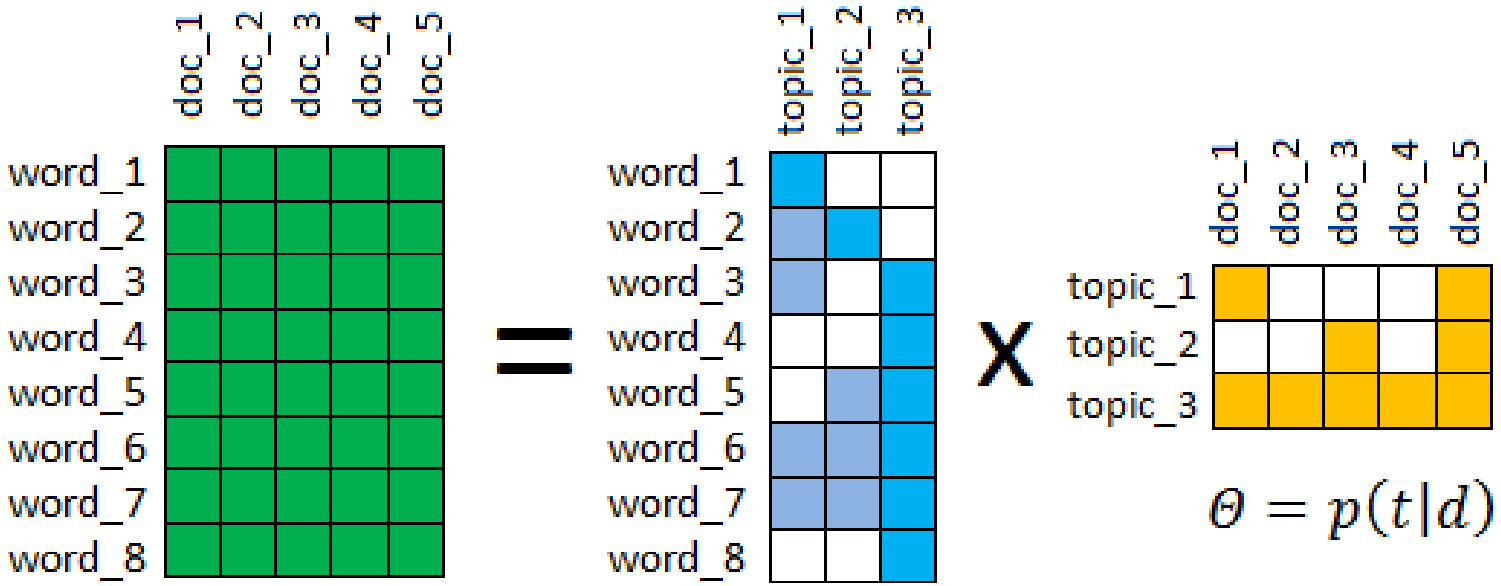
- темы должны состоять из небольшого числа слов, и эти множества слов не должны сильно пересекаться;
- каждый документ должен относиться к небольшому числу тем.



Пример

Извлечение специфичной тематики по ключевым словам:

- хотим собрать темы около интересующих слов, а документы — около интересующих тем;
- прочие темы хотим сглаживать по неважным словам, чтобы собрать «мусор».



ARTM и регуляризованный EM-алгоритм

Максимизация логарифма правдоподобия с дополнительными аддитивными регуляризаторами R:

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

EM-алгоритм: метод простых итераций для системы уравнений

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

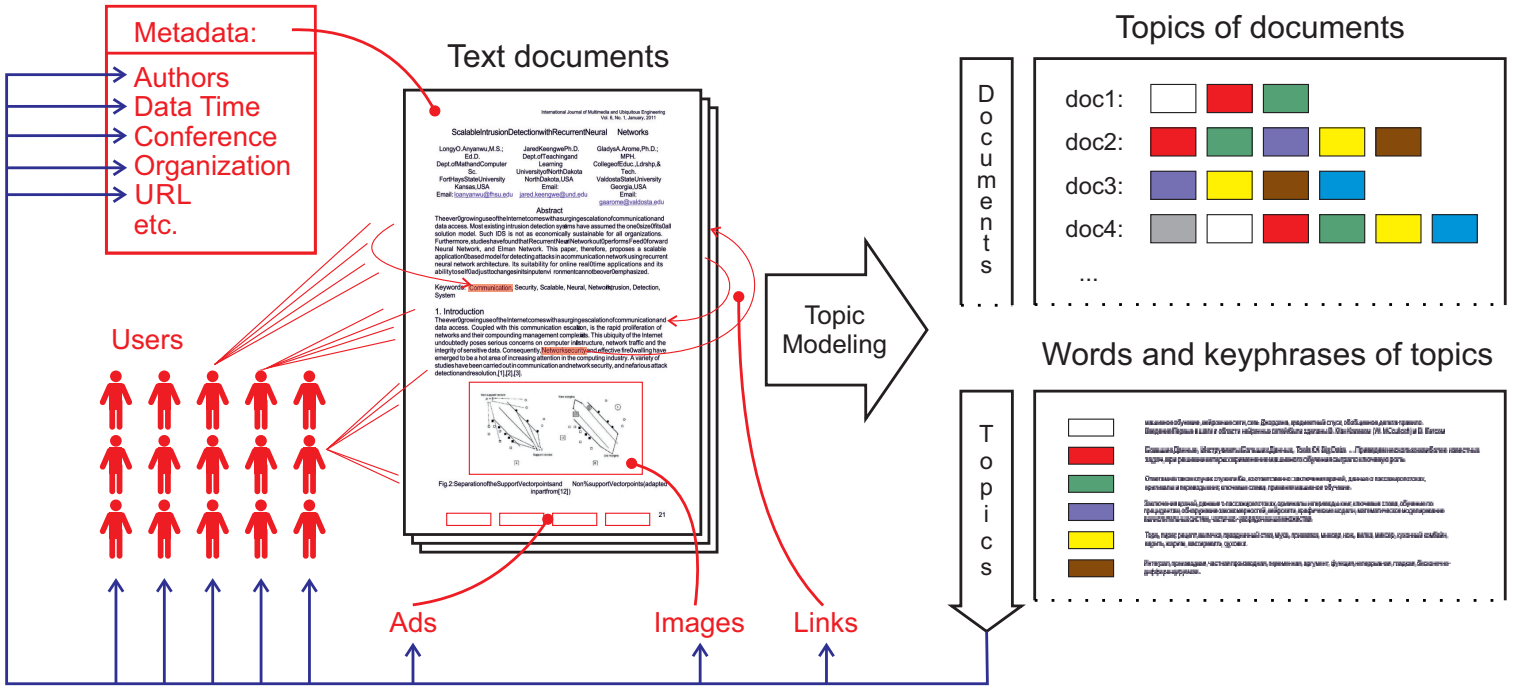
$$\text{M-шаг:} \quad \begin{cases} \phi_{wt} = \underset{w\in W}{\text{norm}}\left(n_{wt} + \phi_{wt}\dfrac{\partial R}{\partial \phi_{wt}} \right), \quad n_{wt} = \sum\limits_{d\in D} n_{dw} p_{tdw} \\ \theta_{td} = \underset{t\in T}{\text{norm}}\left(n_{td} + \theta_{td}\dfrac{\partial R}{\partial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

Примеры регуляризаторов

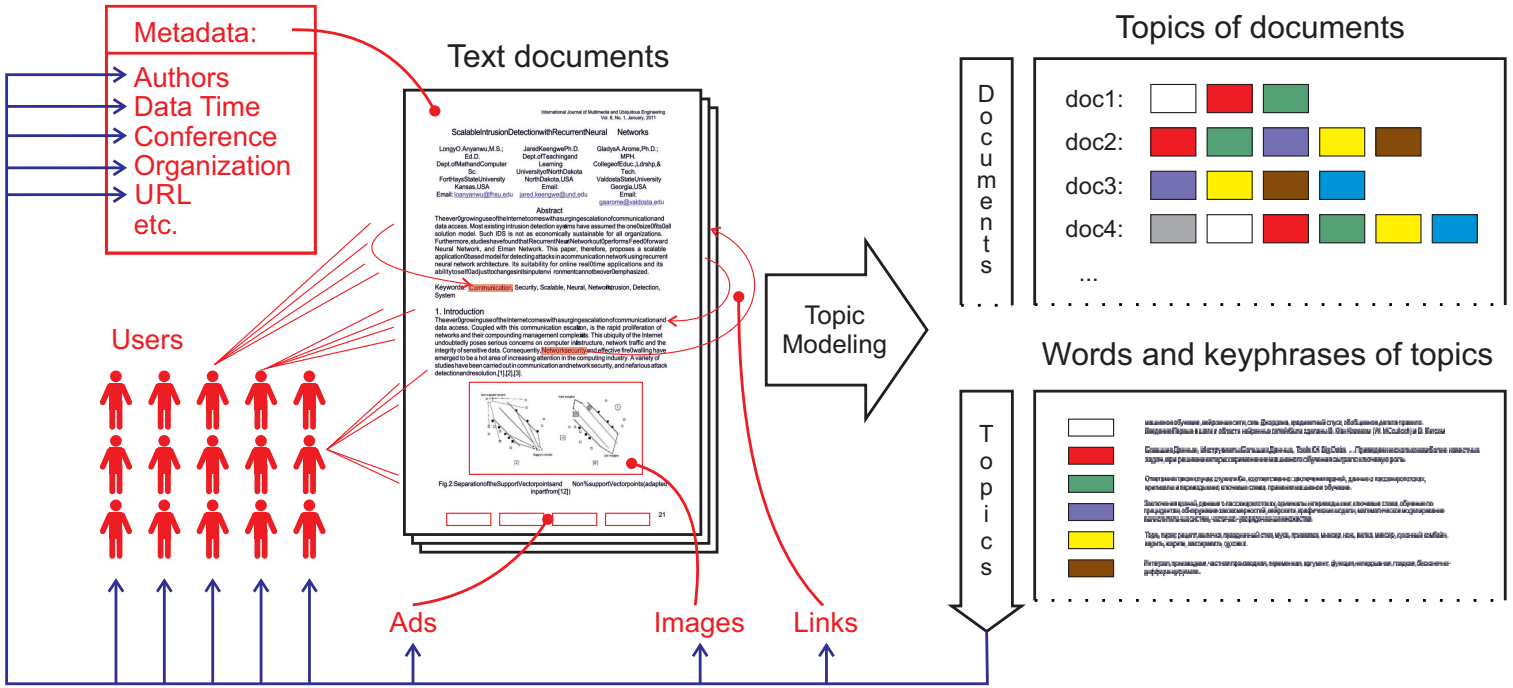
Существует много регуляризаторов. В APTM наиболее простые и популярные следующие:

1. **Сглаживание** Φ / Θ — приводит к известной модели LDA.
2. **Разреживание** Φ / Θ — повышает разреженность тем и, как следствие, их интерпретируемость и различность.
3. **Декорреляция тем в** Φ — повышение различности тем (тоже разреживающая стратегия).
4. **Частичное обучение** — использование сглаживания/разреживания с предопределёнными словарями ключевых слов

Модальности слов



Мультимодальная тематическая модель

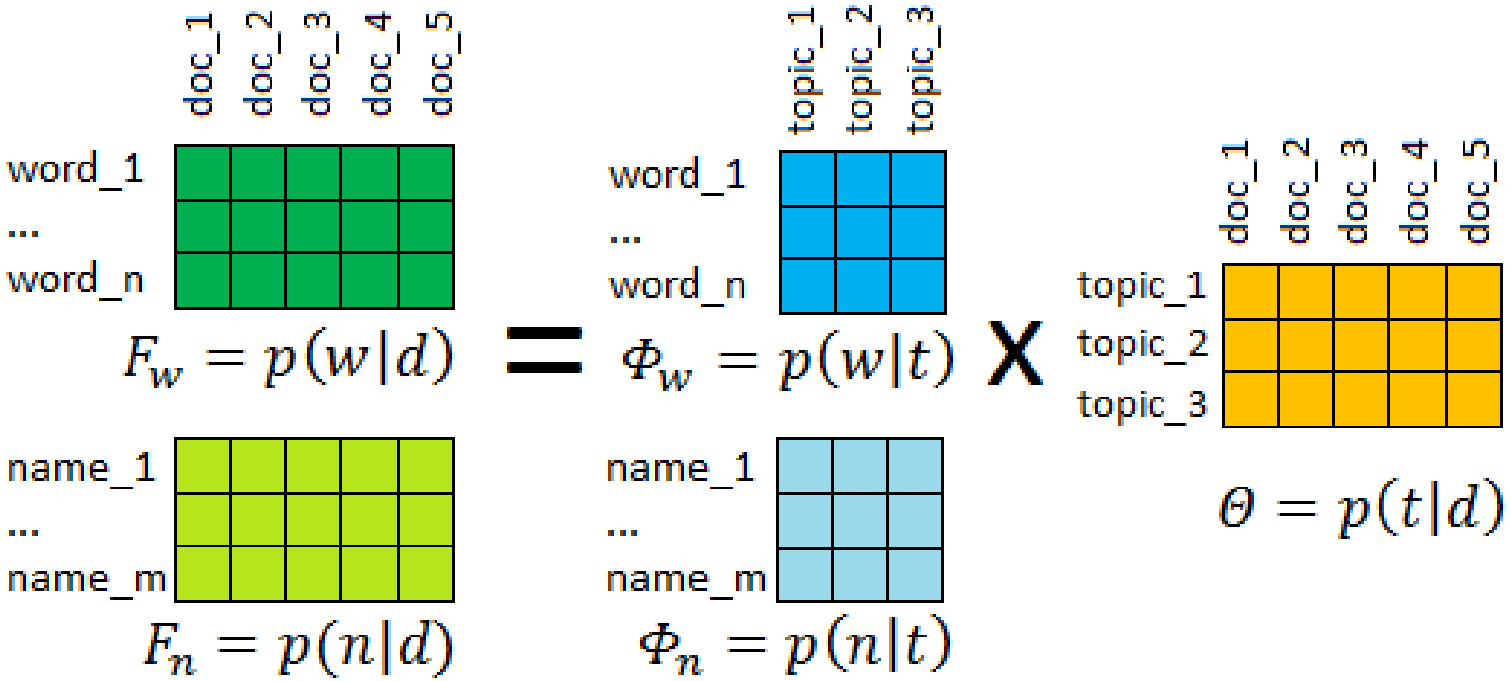


Мультимодальная TM строит распределения тем на терминах $p(w|t)$, авторах $p(a|t)$, метках времени $p(y|t)$, связанных документах $p(d'|t)$, рекламных баннерах $p(b|t)$, пользователях $p(u|t)$, и объединяет все эти модальности в одно тематическую модель.

Пример

Пусть у нас есть две модальности:

- обычные слова;
- слова-имена авторов (а можно, например, метки классов).



M-ARTM и мультимодальный регуляризованный EM-алгоритм

W^m — словарь терминов m-й модальности, $m \in M$,
$W = W^1 \sqcup W^m$ как объединение словарей всех модальностей.

Максимизация логарифма мультимодального правдоподобия с аддитивными регуляризаторами R:

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

EM-алгоритм: метод простых итерация для системы уравнений

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

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

Проект BigARTM

Особенности BigARTM:

- Быстрая[1] параллельная и онлайновая обработка данных;
- Поддержка мультимодальных регуляризованных тематических моделей;
- Встроенная расширяемая библиотека регуляризаторов и метрик качества;

Сообщество BigARTM:

- Открытый репозиторий `https://github.com/bigartm`
- Описание и документация `http://bigartm.org`

Лицензия BigARTM и программные особенности:

- Бесплатное коммерческое использование (BSD 3-Clause license)
- Кроссплатформенная — Windows, Linux, Mac OS X (32 bit, 64 bit)
- Программные API: command line, C++, Python

[1] Vorontsov K., Frei O., Apishev M., Romov P., Dudarenko M. BigARTM: Open Source Library for Regularized Multimodal Topic Modeling of Large Collections Analysis of Images, Social Networks and Texts. 2015

Алгоритм обучения

Оффлайн EM-алгоритм

1. Многократное итерирование по коллекции.
2. Однократный проход по документу.
3. Необходимость хранить матрицу Θ.
4. Φ обновляется в конце каждого прохода по коллекции.
5. Применяется при обработке небольших коллекций.

Онлайн EM-алгоритм

1. Однократный проход по коллекции.
2. Многократное итерирование по документу.
3. Нет необходимости хранить матрицу Θ.
4. Φ обновляется через определённое число обработанных документов.
5. Применяется при обработке больших коллекций в потоковом режиме.

Алгоритм обучения

В BigARTM в разное время работали разные варианты онлайнового алгоритма:

- Онлайновый синхронный (детерминированный, более медленный).
- Онлайновый асинхронный `Async` (недетерминированный, более быстрый).
- Онлайновый асинхронный детерминированный `DetAsync`[2].

Все версии соответствующим изменением параметров могут быть превращены в оффлайновые.

[2] Oleksandr Frei and Murat Apishev. Parallel Non-blocking Deterministic Algorithm for Online Topic Modeling // AIST'2016, Analysis of Images, Social networks, and Texts. Springer International Publishing Switzerland, 2016. (to appear in 661 volume of Communications in Computer and Information Science)

Сравнение с Gensim и Vowpal Wabbit LDA

- 3.7M статей англ. Википедии, $|W|$=100k

Фреймворк	procs	обучение	вывод	перплексия
BigARTM (Async)	1	35 min	72 sec	4000
LdaModel	1	369 min	395 sec	4161
VW.LDA	1	73 min	120 sec	4108
BigARTM (Async)	4	9 min	20 sec	4061
LdaMulticore	4	60 min	222 sec	4111
BigARTM (Async)	8	**4.5 min**	**14 sec**	4304
LdaMulticore	8	57 min	224 sec	4455

- *procs* = число параллельных потоков
- *вывод* = время подсчёта распределений θ_d для теста в 100k док-в
- Новый `DetAsync` сходится ещё быстрее старого `Async`.

Список имеющихся регуляризаторов

BigARTM реализует концепцию M-ARTM. Ниже представлены некоторые имеющиеся регуляризаторы, всегда можно добавлять новые:

1. Сглаживание/разреживание Θ (SmoothSparseThetaRegularizer[3])
2. Сглаживание/разреживание Φ (SmoothSparsePhiRegularizer)
3. Декоррелирование тем в Φ (DecorrelatorPhiRegularizer)
4. Регуляризатор отбора тем по матрице Θ (TopicSelectionThetaRegularizer)
5. Регуляризатор повышения когерентности[4] (ImproveCoherencePhiRegularizer)

Полный список с описаниями можной найти в онлайн-документации.

[3] названия классов в Python API

[4] мера качества, коррелирующая с человеческими оценками интерпретируемости

Список имеющихся метрик качества

По аналогичному модульному принципу в BigARTM реализованы метрики качества моделирования. Ниже представлены некоторые имеющиеся метрики, всегда можно добавлять новые:

1. Перплексия (PerplexityScore[5])
2. Разреженность Φ (SparsityPhiScore)
3. Разреженность Θ (SparsityThetaScore)
4. Характеристики ядер тем + когерентность (TopicKernelScore[6])
5. Наиболее вероятные в темах слова + когерентность (TopTokensScore)

Полный список с описаниями можной найти в онлайн-документации.

[5] названия классов в Python API

[6] Vorontsov K. V., Potapenko A. A. Tutorial on Probabilistic Topic Modeling: Additive Regularization for Stochastic Matrix Factorization // AIST'2014, Analysis of Images, Social networks and Texts. — Springer International Publishing Switzerland, 2014. Communications in Computer and Information Science (CCIS). Vol. 436. pp. 29–46.

Другие особенности

- Все регуляризаторы и метрики приспособлены для работы с мультимодальными моделями.
- Библиотека позволяет напрямую работать с матрицей вспомогательных переменных p_{tdw}, в том числе и регуляризовывать её.
- Есть возможность построения иерархических тематических моделей.
- Помимо представления «мешка слов» BigARTM позволяет работу с последовательным текстом.
- Поддерживается как обработка данных с диска, так и потоковая в RAM.

Про входные форматы данных

BigARTM оперирует данными во внутреннем бинарном представлении, называемыми *батчами*.

Получить батчи из своих данных можно с помощью встроенного парсера, который поддерживает несколько типов входных форматов, основной — формат Vowpal Wabbit.

Это текстовый файл, в котором каждая строка является одним документом. Формат строк следующий:

[<title>] [|@default_class] {token_1[:counter_1]} {other modalities}

doc1 Alpha Bravo:10 Charlie:5 |@author Ola_Nordmann
doc2 Bravo:5 Delta Echo:3 |@author Ivan_Ivanov

Детальное описание всех форматов есть в онлайн-документации.

Command Line Interface

У библиотеки есть CLI, поддерживающий почти все её возможности. Подробное его описание можно найти в онлайн-документации.

Пример запуска утилиты:

bigartm -c docword.vw.txt -v vocab.vw.txt -t 20 –num_collection_passes 10

В документации доступно скачивание нескольких подготовленных датасетов для модельных экспериментов.

Также в BigARTM имеется программный интерфейс для языка C++.

Python API

Особенности:

- Основной интерфейс для исследователей.
- Похож на интерфейс библиотек из scikit-learn.
- Охватывает все возможности BigARTM.
- Поддерживает и Python 2, и Python 3.

Основные классы:

- **artm.ARTM** — основной класс, модель M-ARTM
- **artm.LDA** — модель LDA, более простая в использовании
- **artm.hARTM** — иерархическая модель M-APTM
- **artm.BatchVectorizer** — класс-обёртка для входных данных
- **artm.Dictionary** — класс словаря с данными о коллекции
- Классы регуляризаторов и метрик качества

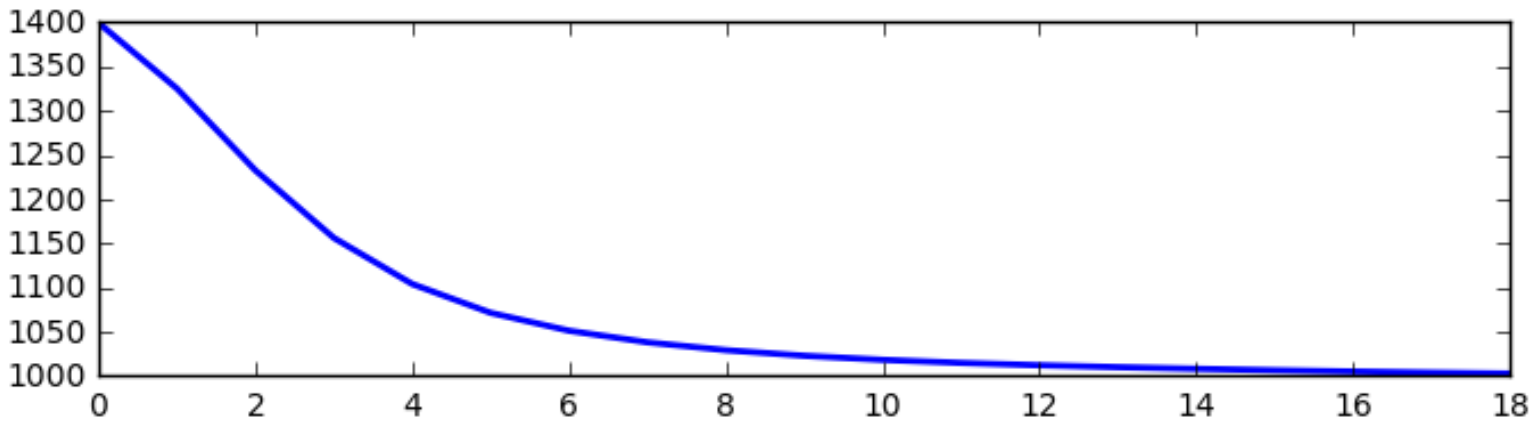
Модель PLSA

Обучим модель PLSA с помощью оффлайн-алгоритма:

```
import artm

bv = artm.BatchVectorizer(data_path='vw.mmro.txt',
                          data_format='vowpal_wabbit',
                          batch_size=1000,
                          target_folder='my_batches',
                          gather_dictionary=True)

model = artm.ARTM(num_topics=10,
                  cache_theta=True,
                  dictionary=bv.dictionary)

model.scores.add(
    artm.PerplexityScore(name='perp',
                         dictionary=bv.dictionary))
```

Модель PLSA – обучение

```
1 %matplotlib inline
2 import matplotlib.pyplot as plt
3
4 model.fit_offline(batch_vectorizer=bv,
5                   num_collection_passes=20)
6
7 perp_values = model.score_tracker['perp'].value[1: ]
8 plt.figure(figsize=(8,2))
9 plt.plot(xrange(len(perp_values)), perp_values)
```



Модель PLSA – обучение

```
model.scores.add(
    artm.TopTokensScore(name='top_tok',
                        num_tokens=8))

model.fit_offline(batch_vectorizer=bv,
                  num_collection_passes=30)

tokens = model.score_tracker['top_tok'].last_tokens
for topic_name in model.topic_names:
    print '{}: '.format(topic_name),
    for token in tokens[topic_name]:
        print token,
    print
```

Модель PLSA – оценивание

topic_0:	объект признак класс распознавание множество классификация образ пространство
topic_1:	последовательность слово метод признак который быть текст структура
topic_2:	алгоритм оценка выборка множество ошибка обучение метод обучать
topic_3:	система дать модель решение анализ который метод задача
topic_4:	задача решение множество алгоритм вектор число последовательность который
topic_5:	функция метод параметр преобразование система оценка быть значение
topic_6:	сигнал дать анализ метод быть результат обработка время

Проблемы

В целом, темы дают представление о том, о чём говорится в коллекции (это тексты статей конференции ММРО).

Но! Никакой конкретики. По топ-словам тем нельзя с уверенностью сказать о том, о чём именно они.

Почему так получилось:

1. Документов в коллекции довольно мало (1062).
2. Коллекция не была должным образом фильтрована (слово «быть» — стоп-слово).
3. В модели слишком мало тем.
4. Было проделано недостаточное количество итераций EM-алгоритма.
5. Не были использованы регуляризаторы.
6. Использовались только униграммы (без качественных биграмм).

Модель ARTM – обучение

Обучим модель тем же кодом, изменив три вещи:

1. зададим большее количество тем (50);
2. добавим регуляризатор декорреляции тем ($\tau = 10^5$);
3. увеличим количество итераций по коллекции (100).

```
1 model.regularizers.add(
2     artm.DecorrelatorPhiRegularizer(name='decor',
3                                     tau=1e+5))
```

Задача подбора регуляризаторов, их коэффициентов и траектории регуляризации до сих пор является открытой проблемой.

Как правило, это требует некоторого опыта, чутья и использования накопленных эвристик и соображений.

Модель ARTM – оценивание

Темы получились несколько лучше. Здесь приведены наиболее качественные, но и это далеко не предел. Использование биграмм могло бы улучшить результат колоссально.

кластер кластеризация тренд центр кластерный нормировка средний тест

анализ время ряд временной момент основа интервал особенность

алгоритм оценка выборка множество ошибка обучение метод обучать

диагностика заболевание больной диагностический врач медицинский жизнь пациент

платон идеальный математика внимание идеал ступень идея ткань

цена рынок участник индекс мировой финансовый фондовый торг

сигнал спектр частота источник активность магнитный мозг частотный

Что нужно, кроме BigARTM

Помимо самого пакета BigARTM, установленного и настроенного для работы из Python, желательно уметь пользоваться следующими инструментами:

- Jupyter Notebook
- Лемматизаторы (pymorphy2, pymystem)
- Базовые средства обработки текстов из nltk
- Модули numpy, pandas, re и matplotlib
- Программы для просмотра больших текстовых файлов (Windows: emeditor, Linux/MacOS: less)

Постановка задачи

Дано:

- Коллекция постов сайта LiveJournal.
- Словарь этнонимов (слов, связанных с различными этничностями).

Задача: выявить как можно большее количество качественных тем, связанных с этно-проблемами.

Метрика качества: оценки асессоров.

Параметры коллекции

Параметры коллекции:

- 1.58 млн. документов в виде «мешка слов»;
- 860 тыс. слов словаре;
- коллекция прошла лемматизацию.

Особенности:

- много слов с ошибками;
- коллекция русскоязычная, но присутствую термины на английском, украинском;
- много жаргонных слов и терминов специфических областей — сложно понимать и интерпретировать темы!

Подготовка данных

Парсим данные в формат Vowpal Wabbit, подходящий для BigARTM.
Сохраним только те слова, которые:

1. содержат только символы киррилицы и дефис;
2. содержат не более одного дефиса (есть слова вроде --, ----);
3. имеют длину не менее 3-х символов (есть слова вроде 'а', 'ж');
4. встречаются в коллекции не менее 20 раз;

Объём итогового словаря: 90 тыс слов.

В таких случаях бывают полезны регулярные выражения.

Составление словаря этнонимов

Описание проблемы:

- Имеется словарь из нескольких сотен этнонимов.
- Слова собраны в списки (например [абхаз, абхазец, абхазка])
- Пересечение слов этого словаря со словарём LJ непустое, но многие слова теряются.
- Нужно составить аналогичный словарь, специфичный для LJ.

Можно сделать вручную:

1. преобразовать списки всех слов в один линейный список;
2. пройтись по этому списку и для каждого слова найти все максимально похожие на него;
3. выбрать вручную в получившемся множестве все наиболее этничные слова, по 1-2 на каждый этноним исходного списка.

Объём итогового словаря этнонимов: 250 слов.

Примеры этнонимов

османский	русич
восточноевропейский	сингапурец
эвенк	перуанский
швейцарская	словенский
аланский	вепсский
саамский	ниггер
латыш	адыги
литовец	сомалиец
цыганка	абхаз
ханты-мансийский	темнокожий
карачаевский	нигериец
кубинка	лягушатник
гагаузский	камбоджиец

Про словари в BigARTM

Словари в BigARTM играют огромную роль, они используются:

- инициализации тематической модели;
- в работе некоторых метрик качества;
- в работе некоторых регуляризаторов.

О словарях можно очень много прочесть в нескольких разделах документации.

Словарь в Python можно сохранить на диск методом `artm.Dictionary.save_text(filename)`, отредактировать и загрузить обратно двойственным методом `load_text()`.

Про словари в BigARTM

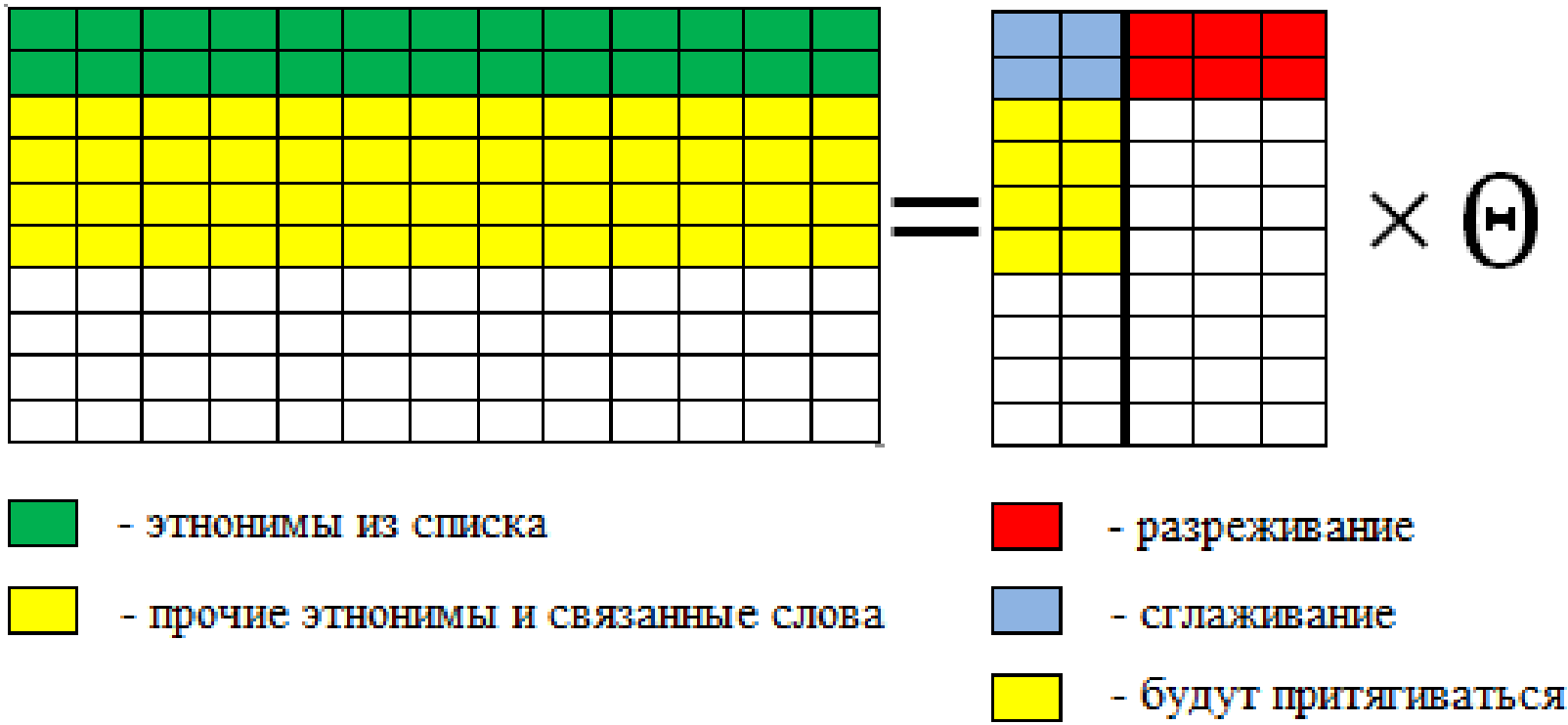
В текстовом виде Dictionary представляет собой набор строк, каждая строка (кроме первой заголовочной) соответствует одному уникальному слову из словаря коллекции.

Строка имеет следующий формат:

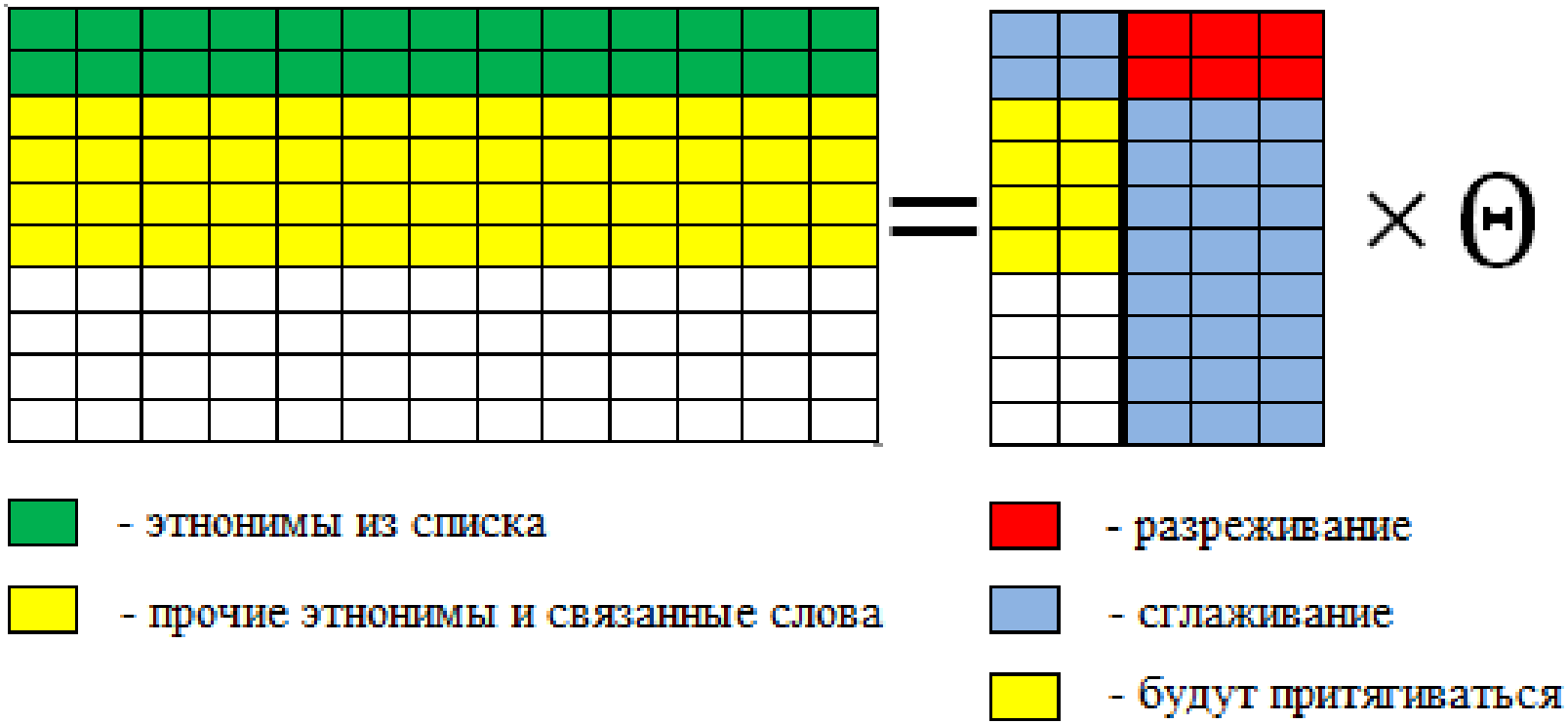
```
token   modality   value   tf   df
```

1. Первые два элемента — это само слово в виде строки и его модальность, последние два — значения tf и df данного слова. Все эти значения считаются библиотекой в процессе парсинга.
2. Поле `value` тоже считается при парсинге, и представляет собой нормированное значение tf. Но его можно переопределять. Оно используется в регуляризаторе SmoothSparsePhi как дополнительный коэффициент регуляризации для данного слова.

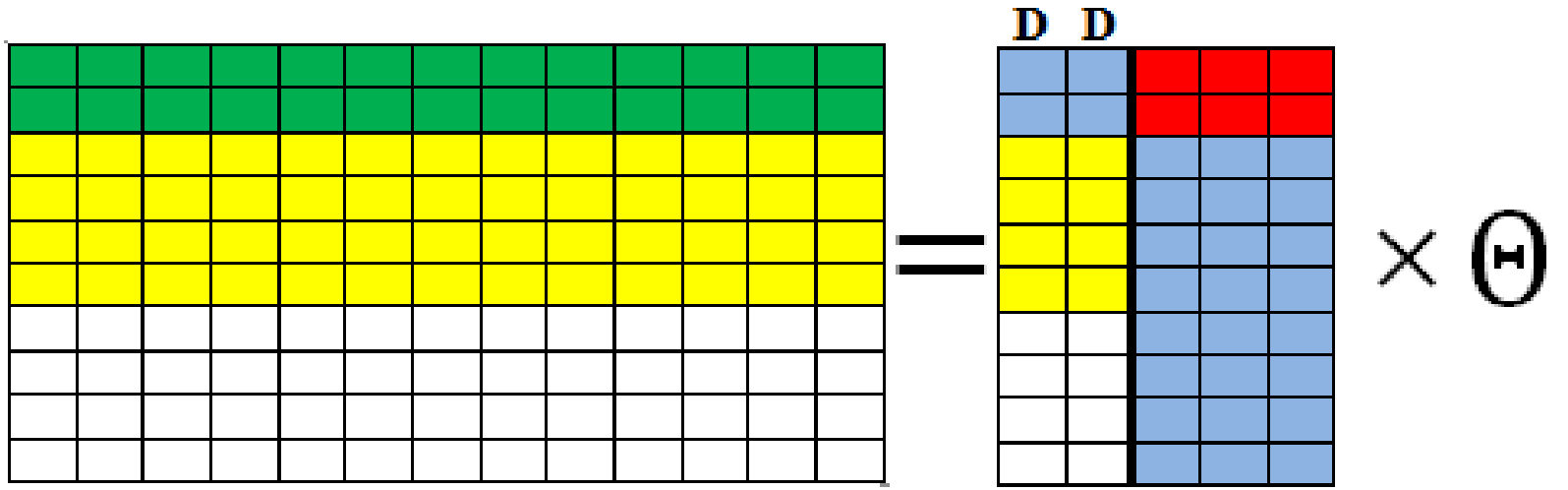
Сглаживание/разреживание этнонимов



+ сглаживание обычных слов



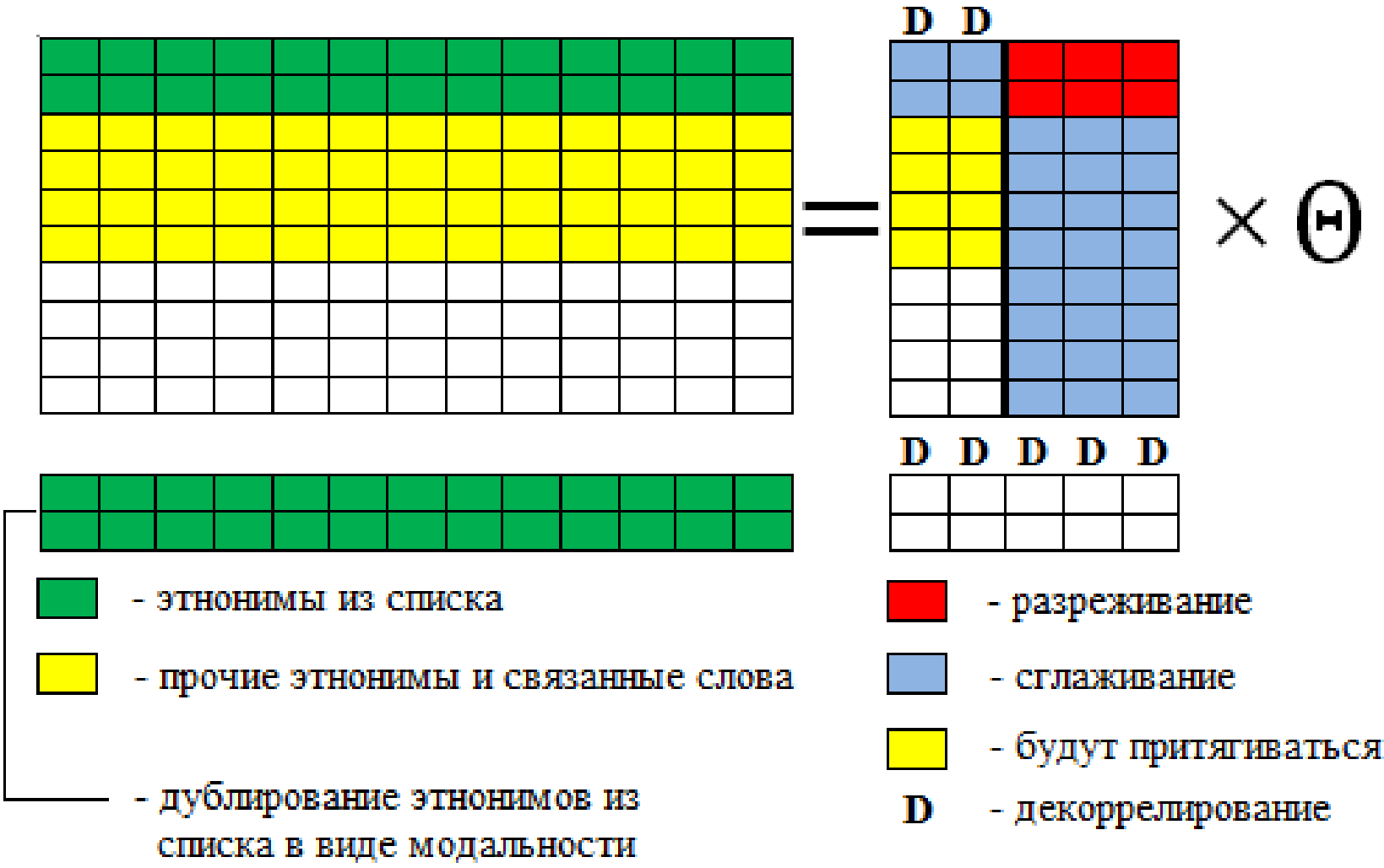
+ декорреляция этничных тем



- этнонимы из списка
- прочие этнонимы и связанные слова
- разреживание
- сглаживание
- будут притягиваться

D - декоррелирование

+ модальность этнонимов



Примеры лучших тем

(русские): акция, организация, митинг, движение, активный, мероприятие, совет, русский, участник, москва, оппозиция, россия, пикет, протест, проведение, националист, поддержка, общественный, проводить, участие,
(славяне, византийцы): славянский, святослав, жрец, древние, письменность, рюрик, летопись, византия, мефодий, хазарский, русский, азбука,
(сирийцы): сирийский, асад, боевик, район, террорист, уничтожать, группировка, дамаск, оружие, алесио, оппозиция, операция, селение, сша, нусра, турция,
(турки): турция, турецкий, курдский, эрдоган, стамбул, страна, кавказ, горин, полиция, премьер-министр, регион, курдистан, ататюрк, партия,
(иранцы): иран, иранский, сша, россия, ядерный, президент, тегеран, сирия, оон, израиль, переговоры, обама, санкция, исламский,
(палестинцы): террорист, израиль, терять, палестинский, палестинец, террористический, палестина, взрыв, территория, страна, государство, безопасность, арабский, организация, иерусалим, военный, полиция, газ,
(ливанцы): ливанский, боевик, район, ливан, армия, террорист, али, военный, хизбалла, раненый, уничтожать, сирия, подразделение, квартал, армейский,
(ливийцы): ливан, демократия, страна, ливийский, каддафи, государство, алжир, война, правительство, сша, арабский, али, муаммар, сирия,

(евреи): израиль, израильский, страна, израил, война, нетаньяху, тель-авив, время, сша, сирия, египет, случай, самолет, еврейский, военный, ближний,

Некоторые результаты

Модель	Лучших тем	Хороших тем	Удовл. тем	Всего
PLSA (300)	9	11	18	38
PLSA (400)	12	15	17	44
С. + Р. + Д. (200+100)	18	33	20	71
С. + Р. + Д. (250+150)	21	27	20	68
С. + Р. + Д. + М. (300+100)	28	23	23	74
С. + Р. + Д. + М. (250+150)	22	25	33	80
С. + Р. + Д. + М. (250+150) (после настройки)	32	42	40	104

Что можно делать ещё?

- Эти эксперименты были продолжены на более крупной и сложной коллекции IQBuzz постов разных русскоязычных социальных медиа (в основном Вконтакте).
- Был вручную собран новый, более полный и насыщенный существительными словарь этнонимов.
- Постановка задачи была усложнена: в дополнение для каждой релевантной темы требовалось исследовать её изменение в пространстве и времени.
- Для этого строились мультимодальные модели с дополнительными модальностями геотегов авторов, а также меток времени публикации сообщения.

Пример темы с привязкой ко времени и пространству

Топ-слова:
чеченский, чечня, кадыров, боевик, террорист, убийство, рамзан, грозный, спецназ, наемник, кавказ, погибать, операция, теракт, вооруженный, боевой, заложник, дудаев, лидер, командир

Топ-геотеги:
Москва, Санкт-Петербург, Чечня

Топ-метки времени:
Сосредоточены в начале и конце декабря 2014

Комментарий:
Совпадает с датой 20-тилетия начала войны в Чечне.

Данные IQBuzz охватывают период 2014-2015 годов.

Тем такого же качества - больше 10% от общего количества и примерно 30% от общего числа признанных этничными.

Постановка задачи

Задача: обучить тематическую модель для классификации документов.

Метрика качества: точность предсказания, т.е. отношение числа верно классифицированных документов к общему числу документов.

Модель будет иметь две модальности, обычные слова (@default_class) и слова-метки классов (@labels_class). Повысить качество её работы можно с помощью регуляризаторов и варьирования различных гиперпараметров.

Данные: коллекция текстов 20newsgroup (тексты предобработаны и разбиты на обучающую и тестовую выборки).

Описание данных и правила

- Датасет состоит из ≈ 18 тыс. документов, разбитых на 20 классов. Каждый документ относится только к одному классу, классы сбалансированны $\Rightarrow$ задача несложная.
- Задание выполняется с использованием BigARTM (интерфейс любой, но рекомендуется использовать Python).
- Датасет и подробные описания будут выложены на kaggle.com.
- Баллы за задание (максимум — 100):
  1. побитие baseline — 40 баллов.
  2. 3 место на private — 70 баллов
  3. 2 место на private — 85 баллов
  4. 1 место на private — 100 баллов
- **Вся** необходимая информация о построении подходящей модели есть в документации BigARTM.

Спасибо за внимание! :)

bigartm.org