Вероятностные тематические модели
Лекция 5. Регуляризаторы для АРТМ

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*: $\Phi\Theta = (\Phi S)(S^{-1}\Theta) = \Phi'\Theta'$.

Напоминание. Задача ARTM и регуляризованный EM-алгоритм

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

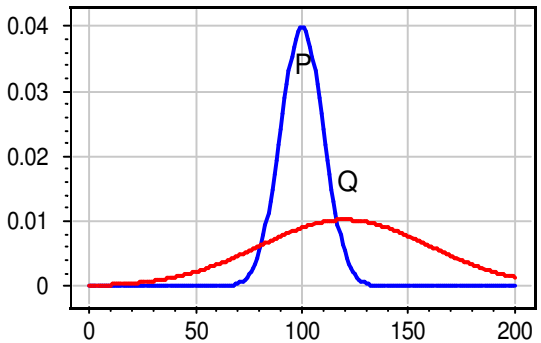
Напоминание. Дивергенция Кульбака–Лейблера

1. $\mathrm{KL}(P\|Q) \geqslant 0; \qquad \mathrm{KL}(P\|Q) = 0 \;\Leftrightarrow\; P = Q;$

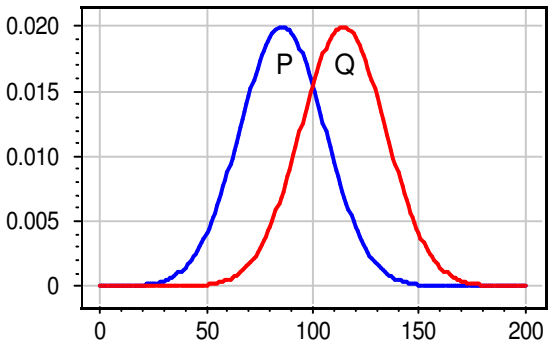
2. Минимизация KL эквивалентна максимизации правдоподобия:

$$\mathrm{KL}(P\|Q(\alpha)) = \sum_{i=1}^{n} p_i \ln \frac{p_i}{q_i(\alpha)} \to \min_{\alpha} \quad \Leftrightarrow \quad \sum_{i=1}^{n} p_i \ln q_i(\alpha) \to \max_{\alpha}$$

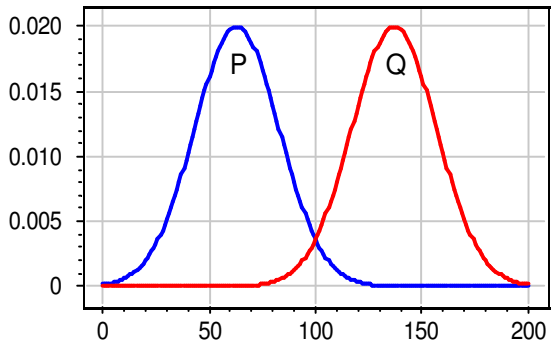
3. Если $\mathrm{KL}(P\|Q) < \mathrm{KL}(Q\|P)$, то P вложено в Q:



$\mathrm{KL}(P\|Q) = 0.44$
$\mathrm{KL}(Q\|P) = 2.97$



$\mathrm{KL}(P\|Q) = 0.44$
$\mathrm{KL}(Q\|P) = 0.44$



$\mathrm{KL}(P\|Q) = 2.97$
$\mathrm{KL}(Q\|P) = 2.97$

Регуляризатор сглаживания (переосмысление LDA)

Гипотеза сглаженности:
распределения ϕ_{wt} близки к заданному распределению β_w;
распределения θ_{td} близки к заданному распределению α_t.

$$\sum_{t\in T} \mathrm{KL}(\beta_w \| \phi_{wt}) \to \min_{\Phi}; \qquad \sum_{d\in D} \mathrm{KL}(\alpha_t \| \theta_{td}) \to \min_{\Theta}.$$

Максимизируем сумму регуляризаторов:

$$R(\Phi,\Theta) = \beta_0 \sum_{t\in T}\sum_{w\in W} \beta_w \ln \phi_{wt} + \alpha_0 \sum_{d\in D}\sum_{t\in T} \alpha_t \ln \theta_{td} \to \max.$$

Подставляем, получаем формулы M-шага LDA:

$$\phi_{wt} = \underset{w\in W}{\mathrm{norm}}(n_{wt} + \beta_0\beta_w), \qquad \theta_{td} = \underset{t\in T}{\mathrm{norm}}(n_{td} + \alpha_0\alpha_t).$$

Этого вы не найдёте в *D.Blei, A.Ng, M.Jordan.* Latent Dirichlet allocation // Journal of Machine Learning Research, 2003. — Vol. 3. — Pp. 993–1022.

Регуляризатор разреживания (обобщение LDA)

Гипотеза разреженности: среди ϕ_{wt}, θ_{td} много нулей;
распределения ϕ_{wt} далеки от заданного распределения β_w;
распределения θ_{td} далеки от заданного распределения α_t.

$$\sum_{t\in T} \mathrm{KL}(\beta_w \| \phi_{wt}) \to \max_{\Phi}; \qquad \sum_{d\in D} \mathrm{KL}(\alpha_t \| \theta_{td}) \to \max_{\Theta}.$$

Максимизируем сумму регуляризаторов:

$$R(\Phi,\Theta) = -\beta_0 \sum_{t\in T}\sum_{w\in W} \beta_w \ln \phi_{wt} - \alpha_0 \sum_{d\in D}\sum_{t\in T} \alpha_t \ln \theta_{td} \to \max.$$

Подставляем, получаем «анти-LDA»:

$$\phi_{wt} = \underset{w\in W}{\mathrm{norm}}\big(n_{wt} - \beta_0\beta_w\big), \qquad \theta_{td} = \underset{t\in T}{\mathrm{norm}}\big(n_{td} - \alpha_0\alpha_t\big).$$

Varadarajan J., Emonet R., Odobez J.-M. A sparsity constraint for topic models — application to temporal activity mining // NIPS-2010.

Объединение сглаживания и разреживания

Общий вид регуляризаторов сглаживания и разреживания:

$$R(\Phi,\Theta) = \beta_0 \sum_{t\in T}\sum_{w\in W} \beta_{wt} \ln \phi_{wt} + \alpha_0 \sum_{d\in D}\sum_{t\in T} \alpha_{td} \ln \theta_{td} \to \max,$$

где $\beta_0 > 0$, $\alpha_0 > 0$ — коэффициенты регуляризации,
β_{wt}, α_{td} — параметры, задаваемые пользователем:

- $\beta_{wt} > 0$, $\alpha_{td} > 0$ — сглаживание
- $\beta_{wt} < 0$, $\alpha_{td} < 0$ — разреживание

Частичное обучение (semi-supervised learning) темы t:

- $\beta_{wt} = [w \in W_t]$ — *белый список* W_t терминов темы t
- $\alpha_{td} = [d \in D_t]$ — *белый список* D_t документов темы t
- $\beta_{wt} = -[w \in W_t]$ — *чёрный список* W_t терминов темы t
- $\alpha_{td} = -[d \in D_t]$ — *чёрный список* D_t документов темы t

Обобщённая KL-дивергенция

KL-дивергенция — это мера сходства векторов (β_w) и $(\ln \phi_w)$:

$$R(\Phi,\Theta) = \beta_0 \sum_{t\in T}\sum_{w\in W} \beta_{wt} \ln(\phi_{wt}) + \alpha_0 \sum_{d\in D}\sum_{t\in T} \alpha_{td} \ln(\theta_{td}) \to \max,$$

Почему бы не заменить ln другой монотонной функцией?

$$R(\Phi,\Theta) = \beta_0 \sum_{t\in T}\sum_{w\in W} \beta_{wt} \mu(\phi_{wt}) + \alpha_0 \sum_{d\in D}\sum_{t\in T} \alpha_{td} \mu(\theta_{td}) \to \max.$$

M-шаг для регуляризатора обобщённой KL-дивергенции:

$$\phi_{wt} = \underset{w\in W}{\operatorname{norm}}\big(n_{wt} + \beta_0\beta_{wt} f(\phi_{wt})\big), \quad \theta_{td} = \underset{t\in T}{\operatorname{norm}}\big(n_{td} + \alpha_0\alpha_{td} f(\theta_{td})\big),$$

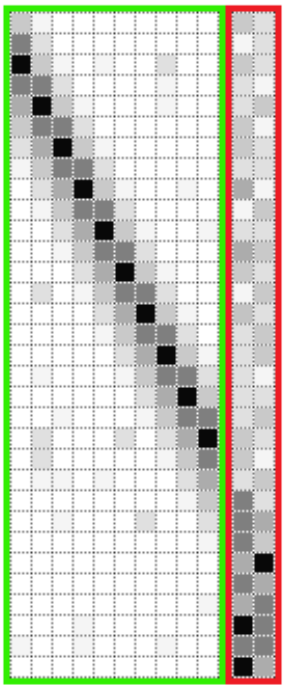
где $f(x) = x\mu'(x)$; в случае KL-дивергенции $\mu \equiv \ln$, $f(x) = 1$.

Разделение тем на предметные и фоновые

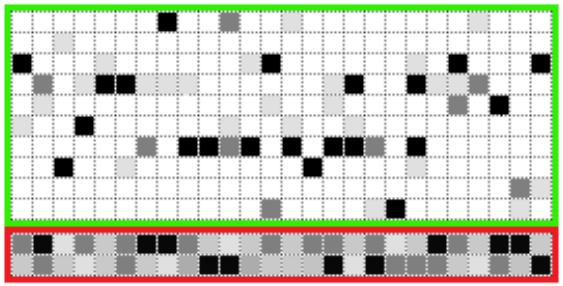
Предметные темы S содержат термины предметной области,
$p(w|t)$, $p(t|d)$, $t \in S$ — разреженные, существенно различные

Фоновые темы B содержат слова общей лексики,
$p(w|t)$, $p(t|d)$, $t \in B$ — существенно отличные от нуля

$\Phi_{W \times T}$



$\Theta_{T \times D}$



Регуляризатор декоррелирования тем

Цель — выделить *лексическое ядро* каждой темы, набор терминов, отличающий её от других тем.

Минимизируем ковариации между вектор-столбцами ϕ_t:

$$R(\Phi) = -\frac{\tau}{2}\sum_{t\in T}\sum_{s\in T\setminus t}\sum_{w\in W}\phi_{wt}\phi_{ws} \to \max.$$

Подставляем, получаем ещё один вариант разреживания — постепенное контрастирование строк матрицы Φ:

$$\phi_{wt} = \underset{w\in W}{\operatorname{norm}}\left(n_{wt} - \tau\phi_{wt}\sum_{s\in T\setminus t}\phi_{ws}\right).$$

Tan Y., Ou Z. Topic-weak-correlated latent Dirichlet allocation // 7th Int'l Symp. Chinese Spoken Language Processing (ISCSLP), 2010. — Pp. 224–228.

Регуляризатор для сокращения числа тем

Цель: избавиться от «мелких» незначимых тем. (заодно получается удалить зависимые и расщеплённые темы)

Разреживаем распределение $p(t) = \sum_d p(d)\theta_{td}$, максимизируя KL-дивергенцию между $p(t)$ и равномерным распределением:

$$R(\Theta) = -\tau \sum_{t \in S} \ln \sum_{d \in D} p(d)\theta_{td} \to \max.$$

Подставляем, получаем:

$$\theta_{td} = \underset{t \in T}{\text{norm}}\left(n_{td} - \tau \frac{n_d}{n_t}\theta_{td}\right), \text{ вариант: } \theta_{td} = \underset{t \in T}{\text{norm}}\left(n_{td}\left(1 - \frac{\tau}{n_t}\right)\right).$$

Эффект: строки матрицы Θ целиком обнуляются для тем t, собравших слишком мало слов по коллекции, $n_t = \sum_{d,w} n_{dwt}$.

Vorontsov K. V., Potapenko A. A., Plavin A. V. Additive Regularization of Topic Models for Topic Selection and Sparse Factorization // SLDS 2015.

Некоторые критерии качества тематической модели

Построение BTM — многокритериальная оптимизация.
Поэтому критериев для контроля качества модели тоже много.

- Перплексия контрольной коллекции: $\mathcal{P} = \exp(-\frac{1}{n}\mathscr{L})$
- Разреженность — доля нулевых элементов в Φ и Θ
- Характеристики интерпретируемости тем:
  - когерентность темы: [Newman, 2010]
  - размер ядра темы: $|W_t|$, ядро $W_t = \{w\colon p(t|w) > 0.25\}$
  - чистота темы: $\sum\limits_{w \in W_t} p(w|t)$
  - контрастность темы: $\frac{1}{|W_t|} \sum\limits_{w \in W_t} p(t|w)$
- Вырожденность тематической модели:
  - число тем: $|T|$
  - доля фона в коллекции: $\frac{1}{n} \sum\limits_{d,w} \sum\limits_{t \in B} p(t|d,w)$

Оценки интерпретируемости: когерентность

Когерентность темы t

$$\mathrm{PMI}_t = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i}^{k} \mathrm{PMI}(w_i, w_j)$$

где w_i — i-й термин в порядке убывания ϕ_{wt}.

$\mathrm{PMI}(u, v) = \ln \frac{P_{uv}}{P_u P_v}$ — поточечная взаимная информация (pointwise mutual information),

P_{uv} — доля документов, в которых термины u, v хотя бы один раз встречаются рядом (в окне 10 слов),

P_u — доля документов, в которых u встретился хотя бы 1 раз.

Newman D., Lau J.H., Grieser K., Baldwin T. Automatic evaluation of topic coherence // Human Language Technologies, HLT-2010, Pp. 100–108.

Разреживание + Сглаживание + Декорреляция + Отбор тем

M-шаг при комбинировании 6 регуляризаторов:

$$\phi_{wt} = \underset{w}{\text{norm}}\Big(n_{wt} + \tau_1 \underbrace{\beta_w[t \in B]}_{\substack{\text{сглаживание}\\ \text{фоновых}\\ \text{тем}}} - \tau_2 \underbrace{\beta_w[t \in S]}_{\substack{\text{разреживание}\\ \text{предметных}\\ \text{тем}}} - \tau_3 \underbrace{\phi_{wt} \sum_{s \in S \setminus t} \phi_{ws}}_{\text{декорреляция}} \Big)$$

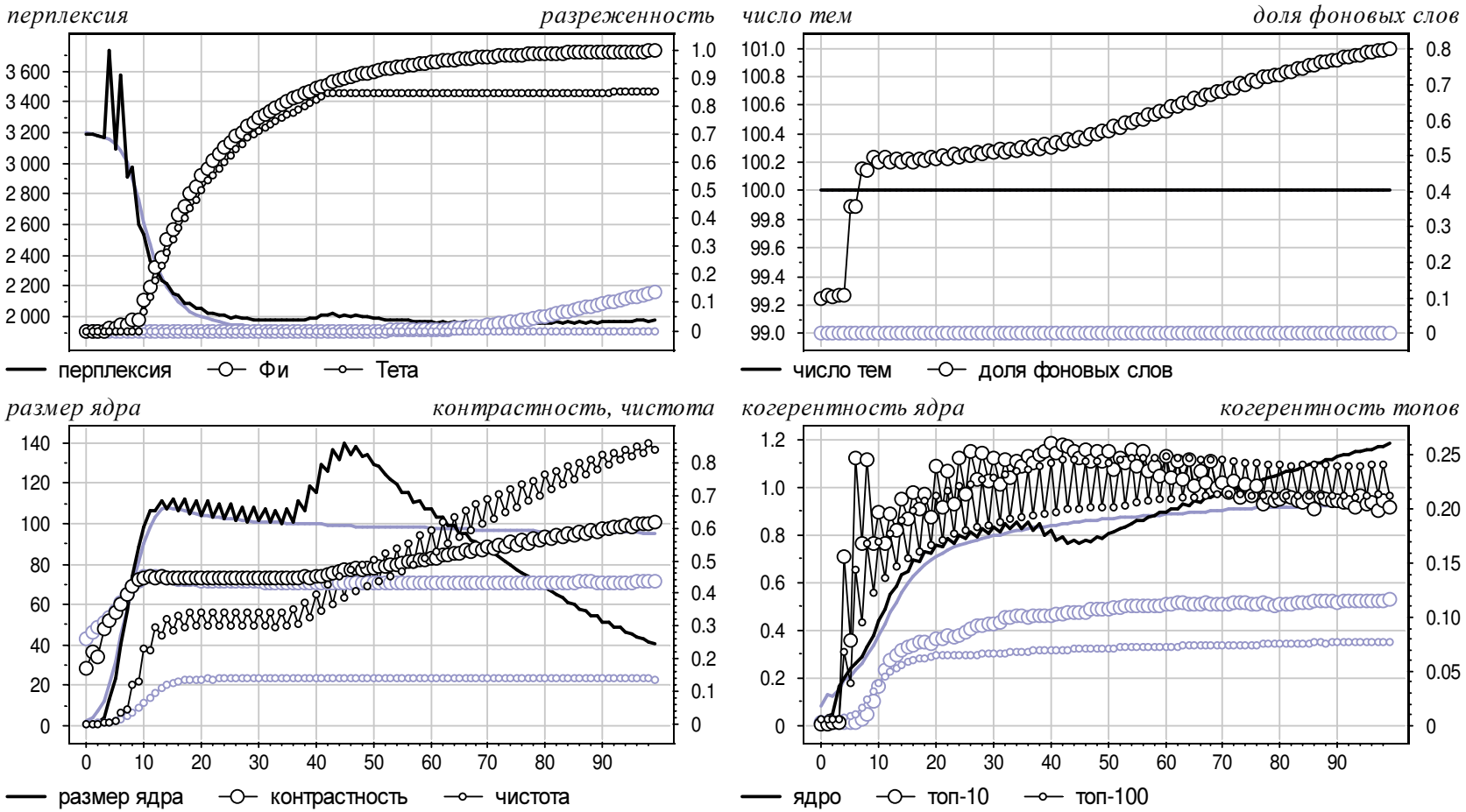
$$\theta_{td} = \underset{t}{\text{norm}}\Big(n_{td} + \tau_4 \underbrace{\alpha_t[t \in B]}_{\substack{\text{сглаживание}\\ \text{фоновых}\\ \text{тем}}} - \tau_5 \underbrace{\alpha_t[t \in S]}_{\substack{\text{разреживание}\\ \text{предметных}\\ \text{тем}}} - \tau_6 \underbrace{\frac{n_d}{n_t}\theta_{td}}_{\substack{\text{удаление}\\ \text{малых тем}}} \Big)$$

Данные: статьи NIPS (Neural Information Processing System)
$|D| = 1566$ статей, $n = 2.3\,\text{M}$, $|W| = 13\,\text{K}$,
контрольная коллекция: $|D'| = 174$.

Vorontsov K. V., Potapenko A. A. Tutorial on Probabilistic Topic Modeling: Additive Regularization for Stochastic Matrix Factorization. AIST'2014.

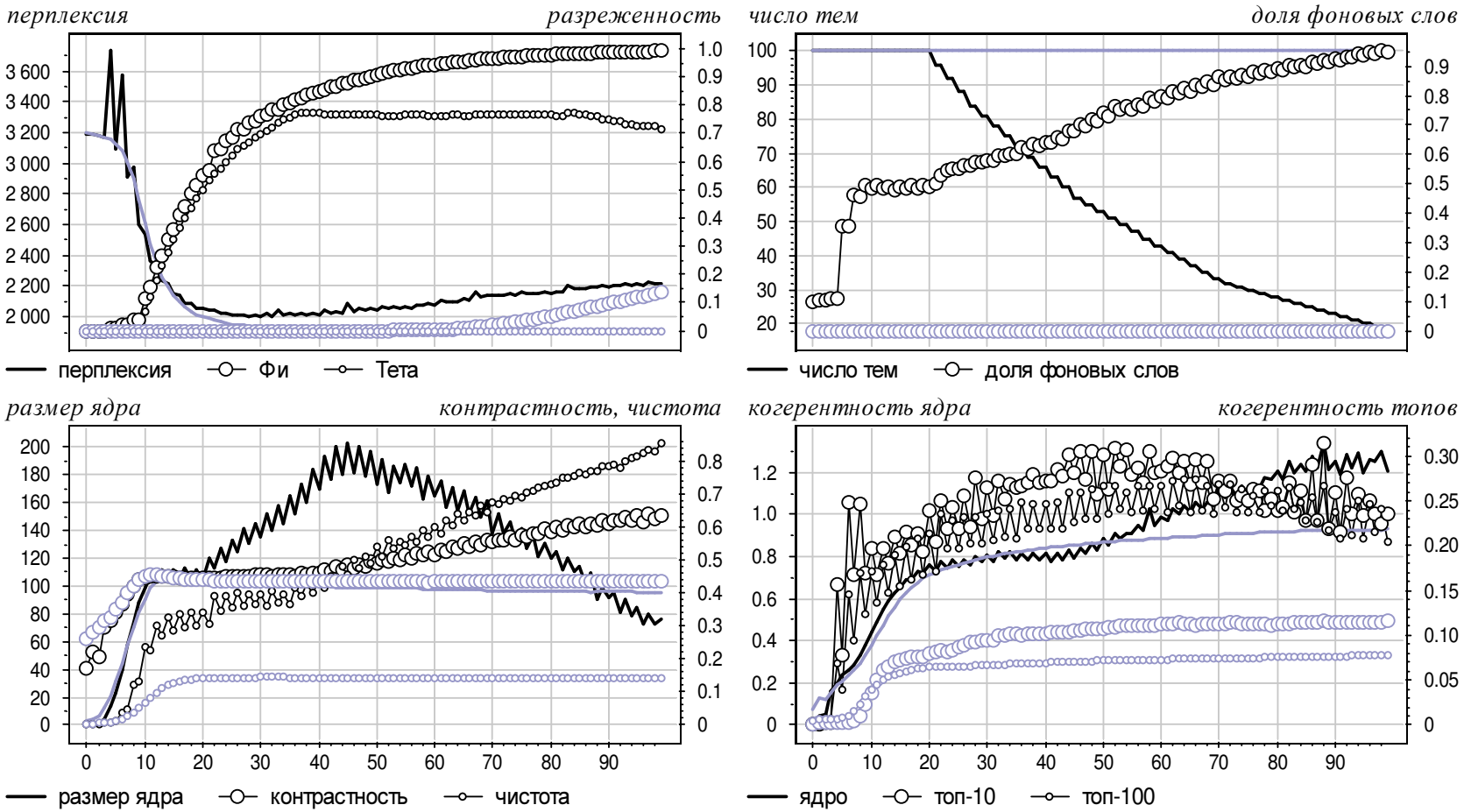
Разреживание, сглаживание, декорреляция

Зависимости критериев качества от итераций EM-алгоритма (серый — PLSA, чёрный — ARTM)



Те же регуляризаторы, плюс отбор тем

Зависимости критериев качества от итераций EM-алгоритма (серый — PLSA, чёрный — ARTM)



Выводы

Одновременное улучшение многих критериев качества:

- *разреженность* выросла от 0 до 95%–98%
- *когерентность тем* выросла от 0.1 до 0.3
- *чистота тем* выросла от 0.15 до 0.8
- *контрастность тем* выросла от 0.4 до 0.6
- почти без потери *перплексии* (правдоподобия) модели

Подобраны траектории регуляризации:

- разреживание включать постепенно после 10-20 итераций
- сглаживание включать сразу
- декорреляцию включать сразу и как можно сильнее
- сокращение числа тем включать постепенно,
- никогда не совмещая с декорреляцией на одной итерации

Эксперименты с регуляризатором отбора тем

Коллекция статей NIPS (Neural Information Processing System)

- $|D| = 1566$ обучающих документов; $|D'| = 174$ тестовых
- $|W| = 13\,\text{K}$ — мощность словаря

Синтетическая коллекция:

- строим PLSA за 500 итераций, $|T_0| = 50$ тем на NIPS
- генерируем (n^0_{dw}) из полученных Φ и Θ:

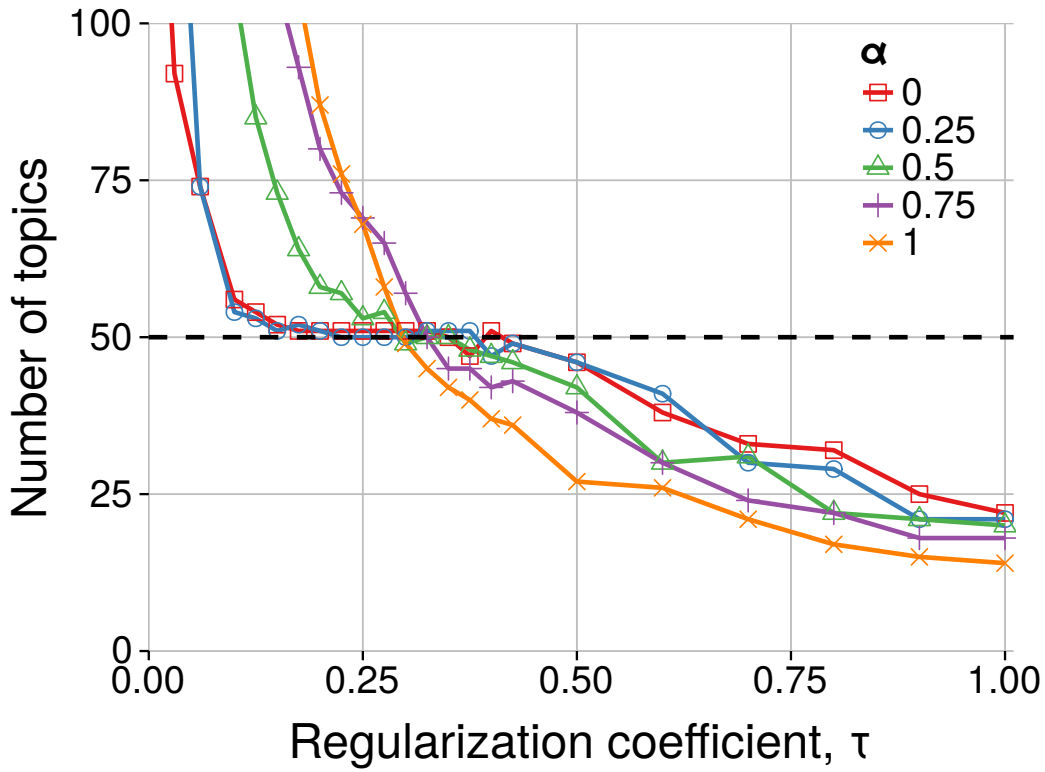
$$n^0_{dw} = n_d \sum_{t \in T} \phi_{wt} \theta_{td}$$

Параметрическое семейство полусинтетических данных:

- n^α_{dw} — смесь синтетических данных n^0_{dw} и реальных n_{dw}:

$$n^\alpha_{dw} = \alpha n_{dw} + (1 - \alpha) n^0_{dw}$$

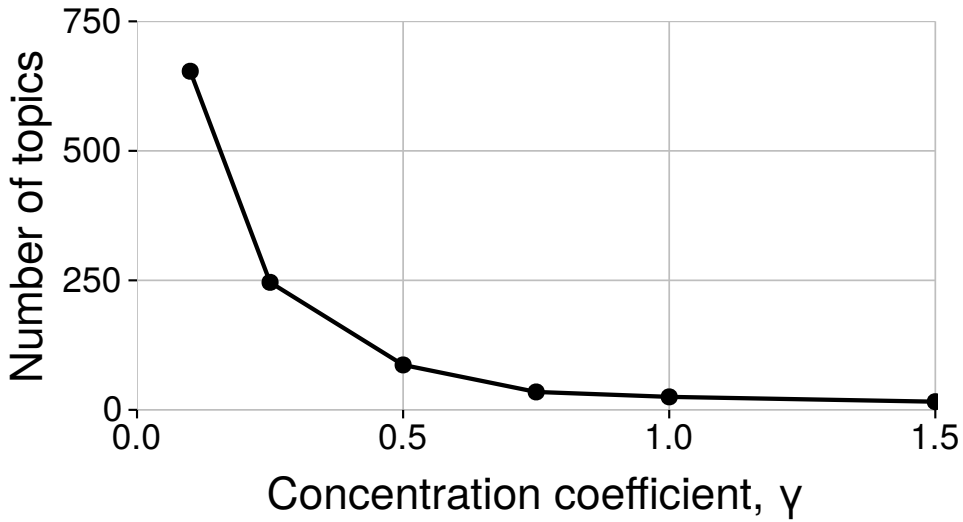
Попытка определения числа тем



- На синтетических данных надёжно находим $|T| = 50$,
- в широком интервале значений коэффициента τ;
- однако на реальных данных нет столь чёткого интервала.

Сравнение с байесовской тематической моделью HDP

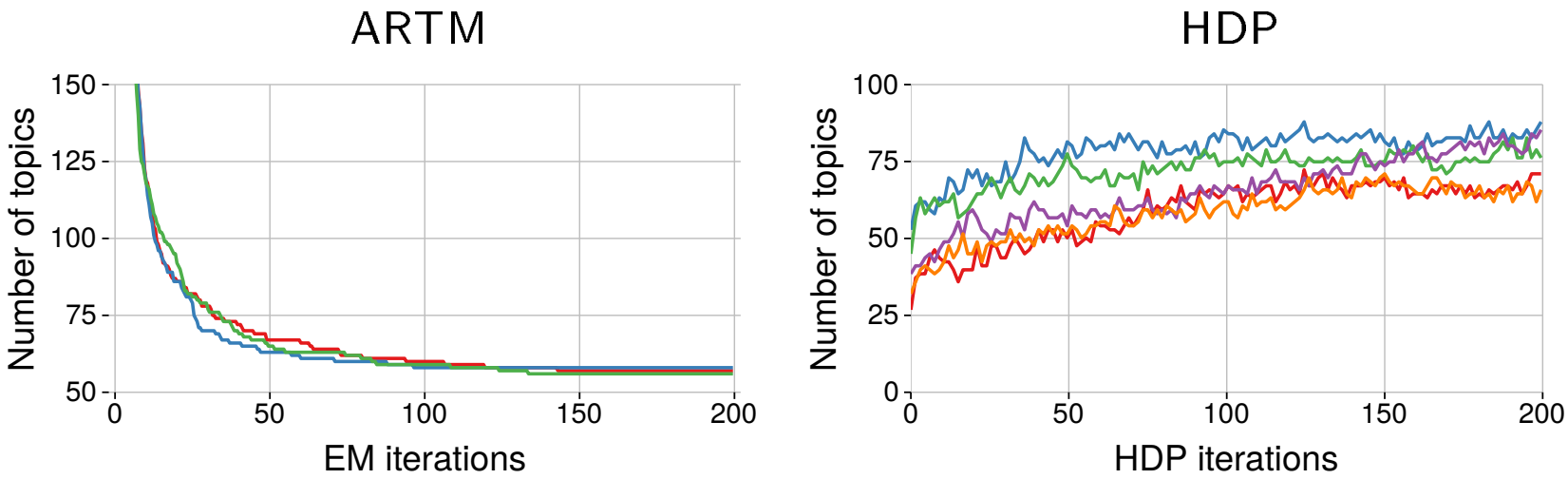
HDP (Hierarchical Dirichlet Process, Tech et. al, 2006) — «state-of-the-art» байесовский подход к определению числа тем



- *Коэффициент концентрации* γ в HDP влияет на $|T|$ так же сильно, как выбор коэффициента τ в ARTM.

Сравнение ARTM и HDP по устойчивости

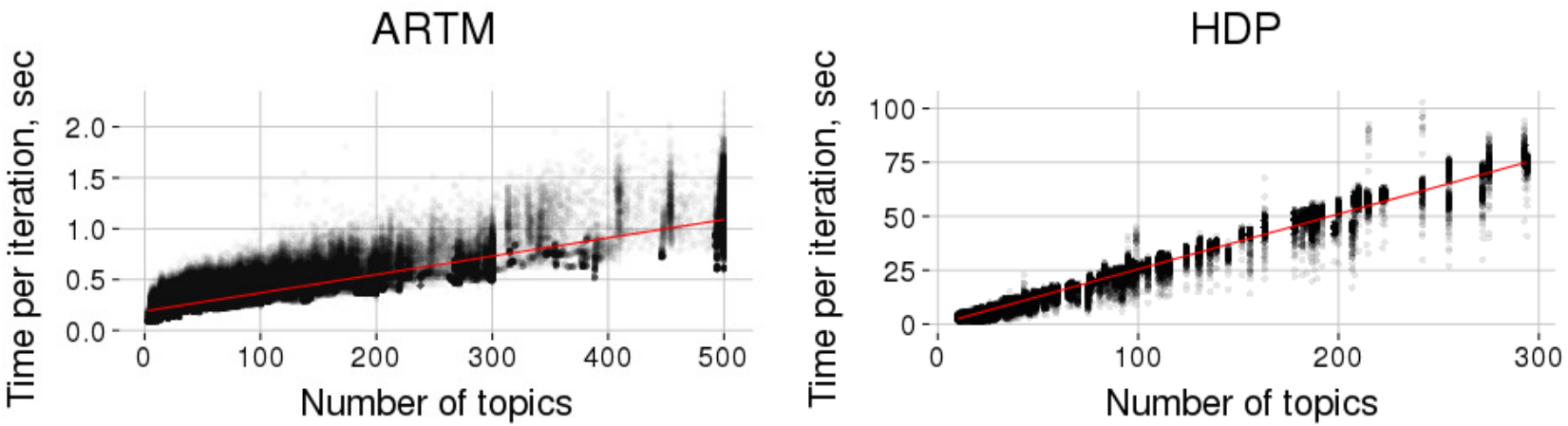
Запуск ARTM и HDP много раз из случайных инициализаций:



- HDP менее устойчив, причём в двух смыслах:
  - число тем сильнее флуктуирует от итерации к итерации;
  - результаты нескольких запусков различаются сильнее.
- «Рекомендуемые» значения параметров γ в HDP и τ в ARTM дают примерно равное число тем $|T| \approx 60$

Сравнение ARTM и HDP по времени вычислений

Сравнение времени одного прохода коллекции (sec)

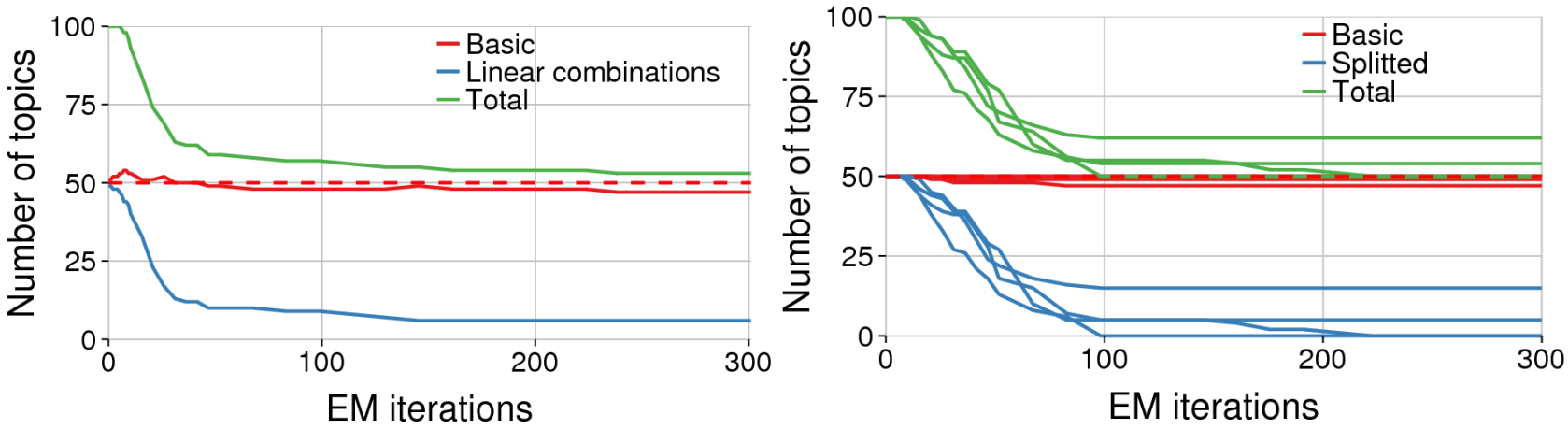


- ARTM в 100 раз быстрее!

Vorontsov K. V., Potapenko A. A., Plavin A. V. Additive regularization of topic models for topic selection and sparse factorization // SLDS 2015, Royal Holloway, University of London, UK. pp. 193–202.

Удаление линейно зависимых и расщеплённых тем

Добавили 50 линейных комбинаций тем в модельную Φ.
Расщепили 50 тем, каждую на две подтемы в модельной Φ.



- Удаляются линейно зависимые и расщеплённые темы
- Остаются более различные темы исходной модели.

Vorontsov K. V., Potapenko A. A., Plavin A. V. Additive regularization of topic models for topic selection and sparse factorization // SLDS 2015, Royal Holloway, University of London, UK. pp. 193–202.

Список классов регуляризаторов

- `artm.SmoothSparsePhiRegularizer`
  — сглаживание или разреживание Φ
- `artm.SmoothSparseThetaRegularizer`
  — сглаживание или разреживание Θ
- `artm.DecorrelatorPhiRegularizer`
  — декоррелятор Φ
- `artm.SpecifiedSparsePhiRegularizer`
  — разреживание Φ с заданной величиной
- `artm.ImproveCoherencePhiRegularizer`
  — повышение когерентности
- `artm.SmoothPtdwRegularizer`
  — сглаживание распределений p_{tdw}
- `artm.TopicSelectionThetaRegularizer`
  — разреживание распределения $p(t)$ для отбора тем

Регуляризатор сглаживания/разреживания матрицы Φ

M-шаг для обобщённой KL-дивергенции:

$$\phi_{wt} = \underset{w \in W^m}{\text{norm}} \left(n_{wt} + \tau \beta_w f(\phi_{wt}) \right)$$

Параметры регуляризатора (все опциональные):

- `name` — имя регуляризатора, строка
- `tau` — коэффициент регуляризации τ, вещественное число
- `topic_names` — список имён тем t, список строк
- `class_ids` — список имён модальностей m, список строк
- `dictionary_name` — имя словаря значений (β_w), строка
- `kl_function_info` — функция $f(x) = x\mu'(x)$,
  по умолчанию $f(x) = 1$ соответствует KL-дивергенции

Параметры name, tau, kl_function_info

Имя регуляризатора используется для его идентификации.

Пример создания регуляризатора:

```
model.regularizer.add(artm.SmoothSparsePhiRegularizer(name='SSPR'))
model.regularizer['SSPR'].tau = -1
model.regularizer['SSPR'].kl_function_info =\
    KlFunctionInfo(function_type='pol', power_value=-1)
```

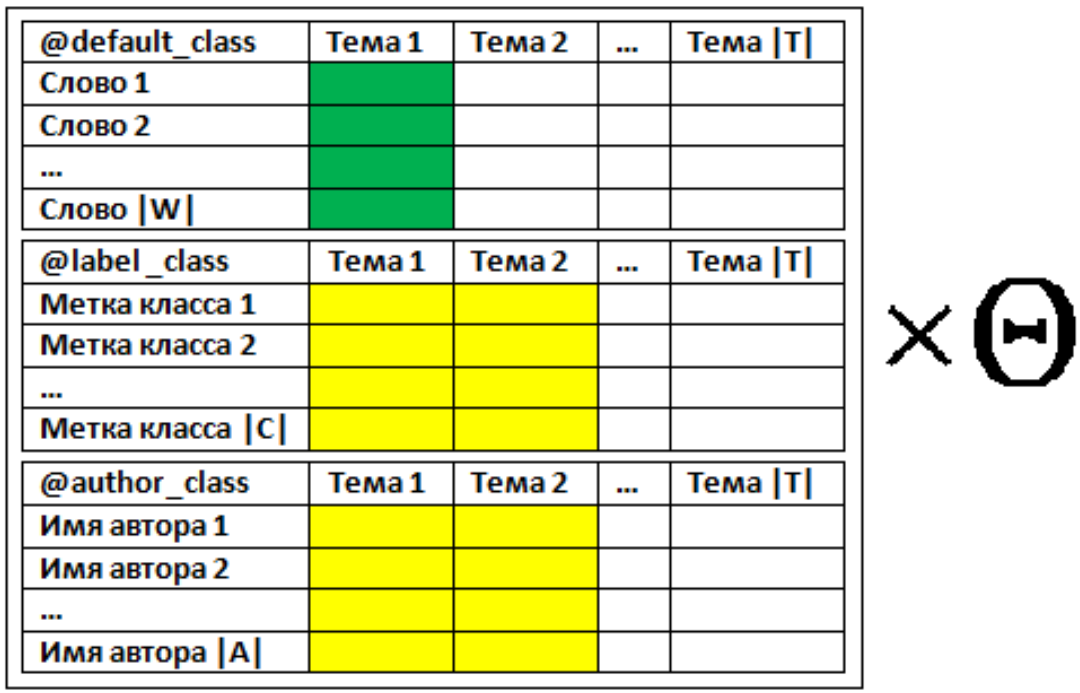
`kl_function_info`— объект класса `KlFunctionInfo`

В данном случае задана функция $f(x) = \frac{1}{x}$, которая приводит к поощрению более редких слов при $\beta_w > 0$:

$$\phi_{wt} = \underset{w \in W^m}{\text{norm}} \left(n_{wt} + \tau \beta_w \frac{1}{\phi_{wt}} \right)$$

Параметры topic_names и class_ids

```
model.regularizer['SSPR'].topic_names= ['Тема_1']
model.regularizer['SSPR'].class_ids= ['@default_class']
model.regularizer['SSPR'].topic_names= ['Тема_1','Тема_2']
model.regularizer['SSPR'].class_ids= ['@label_class','@author_class']
```



Регуляризатор сглаживания/разреживания Θ

M-шаг для обобщённой KL-дивергенции:

$$\theta_{td} = \underset{t \in T}{\text{norm}}\left(n_{td} + \tau\alpha_i f(\theta_{td})\right)$$

Параметры регуляризатора (все опциональные):

- `name` — имя регуляризатора, строка
- `tau` — коэффициент регуляризации τ, вещественное число
- `topic_names` — список имён тем, список строк
- `alpha_iter` — массив значений (α_i),
  на каждую i-ю итерацию по документу, список чисел
- `kl_function_info` — функция $f(x) = x\mu'(x)$,
  по умолчанию $f(x) = 1$ соответствует KL-дивергенции

Регуляризатор декорреляции Ф

Формула M-шага для декоррелирования столбцов Ф:

$$\phi_{wt} = \underset{w \in W^m}{\text{norm}} \left(n_{wt} - \tau \phi_{wt} \sum_{s \in T \setminus t} \phi_{ws} \right).$$

Повышает различность тем как столбцов матрицы Ф.

- `name` — имя регуляризатора, строка
- `tau` — коэффициент регуляризации τ, вещественное число
- `topic_names` — список имён тем t, список строк
- `class_ids` — список имён модальностей m, список строк

Параметр dictionary_name

Регуляризатор сглаживания/разреживания матрицы Φ:

$$\phi_{wt} = \underset{w \in W^m}{\text{norm}} \left(n_{wt} + \tau \beta_w f(\phi_{wt}) \right)$$

β_w позволяет регуляризовать по-разному отдельные слова.
Значение β_w хранится в словаре, своё для каждого слова.

Библиотека генерирует словарь, в котором
$\beta_w = \frac{n_w}{n}$ — относительная частота слова w в коллекции.

Словарь можно редактировать и записывать значения β_w, лучше отвечающие поставленной задаче.

Если словарь отсутствует или β_w для слова w не найдено, то используется $\beta_w = 1$.

Работа со словарями

Словарь создаётся по батчам вызовом

```
model.gather_dictionary(dictionary_name, dictionary_path)
```

В этом случае словарь создаётся в ядре библиотеки, и к нему можно обращаться по имени `dictionary_name`.

Чтобы не создавать словарь каждый раз заново, его можно сохранить на диск, а потом загружать обратно:

```
model.save_dictionary(dictionary_name, dictionary_path)
model.load_dictionary(dictionary_name, dictionary_path)
```

Работа со словарями

Словарь можно редактировать вручную: надо выгрузить содержимое в виде текстового файла, редактировать (например, менять β_w), и загрузить обратно:

```
model.save_text_dictionary(dictionary_name, new_dictionary_path)
model.load_text_dictionary(new_dictionary_name, new_dictionary_path)
```

Отредактированный файл можно сохранить на диск:

```
model.save_dictionary(new_dictionary_name, new_dictionary_path)
```

ВАЖНО: не удаляйте словарь, созданный библиотекой, он полезен для разных метрик (например, перплексии). Создавайте новые словари с другими именами.

Количество загружаемых в ядро словарей не ограничено.

Список классов метрик качества

- `artm.PerplexityScore` — перплексия
- `artm.SparsityPhiScore` — разреженность Φ
- `artm.SparsityThetaScore` — разреженность Θ
- `artm.TopicKernelScore` — ядровые слова тем и когерентности
- `artm.TopTokensScore` — топовые слова тем и когерентности
- `artm.TopicMassPhiScore` — «массы» n_t тем, посчитанные по Φ

Информационные функции (не метрики качества!)

- `artm.ThetaSnippetScore` — сниппет матрицы Θ
- `artm.ItemsProcessedScore` — число обработанных документов

Перплексия

Перплексия коллекции D:

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

- `name` — имя метрики, строка
- `topic_names` — список имён тем, список строк
- `class_ids` — список имён модальностей, список строк
- `dictionary_name` — имя словаря, строка
- `use_unigram_document_model` — флаг использования униграммной модели документа/коллекции

Перплексия, поправка в случае $p(w|d) = 0$

Перплексия коллекции D:

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

Если $p(w|d) = 0$, то используется униграммная модель документа $p(w|d) = \frac{n_{dw}}{n_d}$ или коллекции $p(w|d) = \frac{n_w}{n}$.

Второй вариант лучше, т.к. не занижает перплексию, но он требует словаря значений β_w, $w \in W$.

Поэтому для подсчёта перплексии требуется:

1. загрузить в модель словарь, сгенерированный библиотекой;
2. подключить словарь к метрике через `dictionary_name`;
3. задать `use_unigram_document_model = False`.

Разреженность Φ и топ-слова

Параметры метрики разреженности Φ:

- `name` — имя метрики, строка
- `topic_names` — список имён тем, список строк
- `class_id` — имя модальности, строка
- `eps` — константа толерантности, число

Параметры метрики топовых (наиболее вероятных) слов Φ:

- `name` — имя метрики, строка
- `topic_names` — список имён тем, список строк
- `class_id` — имя модальности, строка
- `num_tokens` — число топовых слов, целое число
- `dictionary_name` — имя словаря, строка

Пример использования метрик

```
model.scores.add(artm.PerplexityScore(name='PSScore',
                                      dictionary_name='dictionary'))
model.scores.add(artm.SparsityPhiScore(name='SPScore'))
model.scores.add(artm.TopTokensScore(name='TTScore', num_tokens=20))
model.fit_offline(num_collection_passes=10)
```

`model.score_tracker['SPScore'].value`
— вернётся список значений на каждой итерации

`model.score_tracker['SPScore'].last_value`
— вернётся финальное значение

`model.score_tracker['TTScore'].last_topic_info['Тема_1'].tokens`
— вернётся финальный топ-20 слов для «темы 1».

`model.score_tracker['TTScore'].last_topic_info['Тема_1'].weights`
— а так можно посмотреть соответствующие им значения ϕ_{wt}.

Резюме

- Разреживание, сглаживание и декоррелирование — «джентльменский набор» регуляризаторов.
- Регуляризатор отбора тем удаляет зависимые темы. Оптимального числа тем вообще не существует!
- Решение задач анализа текстов *в стиле ARTM* — это построение моделей с заданными свойствами путём включения нужного набора регуляризаторов.
- Коэффициенты регуляризации пока подбираем вручную, их автоматическая настройка — в стадии разработки.