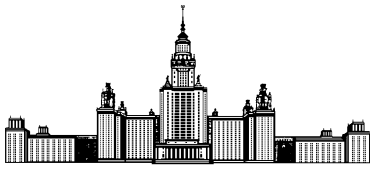
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

Дипломная работа

«Методы построения иерархических тематических моделей коллекции текстовых документов»

Выполнил:

студент 5 курса 517 группы

Гаврилюк Кирилл Артурович

Научный руководитель:

д.ф-м.н., доцент

Воронцов Константин Вячеславович

Москва, 2013

Содержание

Аннотация

В работе рассматривается задача тематического моделирования коллекции текстовых документов. Дается краткий обзор существующих иерархических моделей, а также методов оценки их качества.

На основаниии рассмотренных моделей предложена иерархическая тематическая модель, способная учитывать привязки документов к уже существующему классификатору документов. С использованием данной модели решается задача категоризации документов. Исследуется влияние различных свойств модели на качество категоризации. Показывается, что с помощью предложенной модели удается достичь сопоставимых результатов в сравнении с классическим подходом к иерархической категоризации документов.

Также исследуется вопрос о взаимно однозначном сопоставлении категорий и тем. Показывается, что представление категорий в виде смеси тем, дают лучшее значение качества категоризации.

1 Введение

Тематическое моделирование является одним из современных, активно развивающихся приложений машинного обучения к анализу текстов. Тематическая модель коллекции текстовых документов определяет к каким темам относится каждый документ и какие слова (термины) образуют каждую тему. Число тем в большинстве приложений заранее неизвестно и является одним из важнейших параметров модели.

Важной характеристикой тематических моделей является семантическая значимость восстановленных тематических профилей по коллекции документов. В работе [22] был предложен метод для автоматического определения названия темы для плоских тематических моделей. Если тема имеет семантическую значимость, то ее название является понятным и репрезентативным для человека. Если же семантическая значимость низкая, то определение названия темы будет затруднено и для человека получившееся название будет не репрезентативным. В случае же иерархических моделей ситуация становиться еще сложней, так как помимо восстановленных тем, между темами в иерархии возникают связи типа «тема-подтема». Понимание содержательного значения такой связи между темами является ключевым вопросом в построении иерархических моделей. В современных иерархических моделях связь «тема-подтема» слабо учитывается в вероятностной модели порождения коллекции документов.

Также большинство методов построений тематичеких моделей в качестве входных данных используют не размеченные документы. То есть о документах известно только то, какие слова в этом документе встречаются, а также число их вхождений в документ. При таком представлении документов никак не учитывается возможная привязка документов к уже существующим каталогам и классификаторам. Каталоги и классификаторы считаются наиболее удобным способом хранения, поиска и извлечения информации из существующих баз знаний. В связи с этим возникает задача не просто построения каталога по коллекции неразмеченных документов, но построении модели, которая определяла бы профиль документа по уже существующим каталогам.

В данной работе рассматривается иерархическая тематическая модель, в которой темы взаимно однозначно соответствуют категориям. Показывается, что с помощью профилей документов, построенных по такой модели, можно добиться сопоставимых результатов с классическим подходом для иерархической категоризации документов. Также исследуется подход, при котором категория представляется в виде смеси тем. При таком подходе удается добиться еще лучших результатов.

2 Классические вероятностные тематические модели

Существует ряд «классических» вероятностых тематических моделей, такие как PLSA [13] (Probabilistic Latent Semantic Analysis) и LDA [4] (Latent Dirichlet Allocation), которые используют схожие вероятностные предположения для построения тематических моделей по коллекции текстовых документов. Обозначим через D — коллекцию текстовых документов, а через W — множество или словарь терминов. Каждый документ $d \in D$ будем представлять в виде последовательности терминов $(w_1, \ldots, w_{n_d})$ из W, где n_d — длина документа.

Основные вероятностные предположения:

1. Предполагается, что существует конечное множество тем T, и коллекция документов порождается дискретным распределением $p(d, w, t)$ на $D \times W \times T$, где $t \in T$. Переменные d и w являются наблюдаемыми величинами, а переменная t — скрытой. Построить тематичекую модель означает найти множество тем T, условные распределения $p(w|t) \equiv \varphi_{wt}$ для каждой темы $t \in T$ и $p(t|d) \equiv \theta_{td}$ для каждого документа $d \in D$.

2. Предполагается, что распределение вероятностей терминов $p(w|d, t)$ зависит только от темы t, но не от документа d:

$$p(w|d, t) = p(w|t).$$

3. Предполагается, что для выявления тематики достаточно знать, какие термины встречаются в каких документах, но не важен ни порядок терминов в документах, ни порядок документов в коллекции. Соотвественно документ $d \in D$ рассматривается как выборка терминов, порождаемых случайно и независимо из распределения:

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

2.1 PLSA

Вероятностная тематическая модель появления пары «документ-термин» (d, w) согласно определению условной вероятности, формуле полной вероятности и предположению об условной независимости описывается следующим образом:

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

где распределения $p(w|t), p(t|d)$ являются мультиномиальными распределениями. Алгоритм порождения коллекции документов согласно данной вероятностной модели описан в алгоритме 1.

Algorithm 1 Алгоритм порождения коллекции текстовых документов с помощью вероятностной модели PLSA.

Вход: Распределения $p(w|t), p(t|d)$;

Выход: Выборка пар $(d_i, w_i), i = 1, \ldots, n$;

1: **для всех** $d \in D$
2: Задать длину n_d документа d.
3: **для всех** $i = 1, \ldots, n_d$
4: Выбрать случайную тему t из распределения $p(t|d)$;
5: Выбрать случайный термин w из распределения $p(w|t)$;
6: Добавить в выборку пару (d, w);

Вероятностные распределения, связанные с наблюдаемыми переменными w и d можно оценивать с помощью частотных оценок:

$$\hat{p}(d, w) = \frac{n_{dw}}{n}, \qquad \hat{p}(d) = \frac{n_d}{n}, \qquad \hat{p}(w) = \frac{n_w}{n}, \qquad \hat{p}(w|d) = \frac{n_{dw}}{n_d}$$

n_{dw} — число вхождений термина w в документ d;

$n_d = \sum_{w \in W} n_{dw}$ — длина документа d в терминах;

$n_w = \sum_{d \in D} n_{dw}$ — число вхождений термина w во все документы коллекции;

$n = \sum_{d \in D} \sum_{w \in d} n_{dw}$ — длина коллекции в терминах.

Вероятностные распределения, связанные с скрытой переменной t можно оценивать следующим образом:

$$\hat{p}(t) = \frac{n_t}{n}, \qquad \hat{p}(w|t) = \frac{n_{wt}}{n_t}, \qquad \hat{p}(t|d) = \frac{n_{dt}}{n_d}, \qquad \hat{p}(t|d, w) = \frac{n_{dwt}}{n_{dw}}$$

n_{dwt} — число троек, в которых термин w документа d связан с темой t;

$n_{dt} = \sum_{w \in W} n_{dwt}$ — число троек, в которых термин документа d связан с темой t;

$n_{wt} = \sum_{d \in D} n_{dwt}$ — число троек, в которых термин w связан с темой t;

$n_t = \sum_{d \in D} \sum_{w \in d} n_{dwt}$ — число троек, связанных с темой t.

Обозначим через $\Phi = (\varphi_{wt})_{W \times T}$ — матрицу тем, через $\Theta = (\theta_{td})_{T \times D}$ — матрицу документов. Тогда для оценивания параметров Φ, Θ тематической модели по коллекции документов D можно использовать принцип максимума правдоподобия выборки:

$$p(D; \Phi, \Theta) = C \prod_{d \in D} \prod_{w \in d} p(d, w)^{n_{dw}} = C \prod_{d \in D} \prod_{w \in d} p(w|d)^{n_{dw}} p(d)^{n_{dw}} \to \max_{\Phi, \Theta},$$

где C — нормировачный множитель мультиномиального распределения, зависящий только от чисел n_{dw}. Заметим также, что множитель $p(d)$ не зависит от матриц Φ, Θ, то есть не влияет на решение задачи оптимизации. Отбросим C и $p(d)$ и прологарифмируем правдоподобие, тогда получим следующую задачу оптимизации:

$$L(D; \Phi, \Theta) = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln p(w|d) = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \varphi_{wt} \theta_{td} \to \max_{\Phi, \Theta},$$

при ограничениях неотрицательности $\theta_{td} \geq 0, \varphi_{wt} \geq 0$ и нормировки

$$\sum_{w \in W} \varphi_{wt} = 1, \sum_{w \in W} \theta_{td} = 1$$

Распишем решение данной задачи с помощью лагранжиана с учетом ограничений нормировки, проигнориров ограничения неотрицательности:

$$L(\Phi, \Theta) = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \varphi_{wt} \theta_{td} - \sum_{t \in T} \lambda_t \left(\sum_{w \in W} \varphi_{wt} - 1 \right) - \sum_{d \in D} \mu_d \left(\sum_{t \in T} \theta_{td} - 1 \right)$$

Продифференцировав лагранжиан по φ_{wt} и приравняв получившиеся значение к 0, получим:

$$\lambda_t = \sum_{d \in D} n_{dw} \frac{\theta_{td}}{p(w|d)}$$

Умножим обе части равенства на φ_{wt} и просуммируем по всем $w \in W$. Тогда, учитывая условия нормировки, получим:

$$\lambda_t = \sum_{w \in W} \sum_{d \in D} n_{dw} \frac{\varphi_{wt} \theta_{td}}{p(w|d)}$$

Обозначим через H_{dwt} следующую величину:

$$H_{dwt} = \frac{\varphi_{wt} \theta_{td}}{p(w|d)} = \frac{\varphi_{wt} \theta_{td}}{\sum\limits_{s \in T} \varphi_{ws} \theta_{sd}}$$

Тогда выражение для λ_t примет следующий вид:

$$\lambda_t = \sum_{w \in W} \sum_{d \in D} n_{dw} H_{dwt}$$

Снова умножим первоначальное выражение для λ_t на φ_{wt}. Выразим φ_{wt}, считая выражение для λ_t известным:

$$\varphi_{wt} = \frac{\sum\limits_{d \in D} n_{dw} H_{dwt}}{\sum\limits_{w' \in W} \sum\limits_{d \in D} n_{dw'} H_{dw't}}$$

Введем дополнительные обозначения:

$$\hat{n}_{wt} = \sum_{d \in D} n_{dw} H_{dwt}, \qquad \hat{n}_t = \sum_{w \in W} \hat{n}_{wt}$$

Тогда:

$$\varphi_{wt} = \frac{\hat{n}_{wt}}{\hat{n}_t}$$

Проделав аналогичные действия для θ_{td} получаем:

$$\theta_{td} = \frac{\hat{n}_{dt}}{\hat{n}_d}$$

$$\hat{n}_{td} = \sum_{w \in d} n_{dw} H_{dwt}, \qquad \hat{n}_d = \sum_{t \in T} \hat{n}_{wt}$$

При этом H_{dwt} имеют естественную вероятностную интерпретацию:

$$H_{dwt} = \frac{p(w|t)p(t|d)}{p(w|d)} = p(t|d, w)$$

Считая фиксированными значения H_{dwt} мы можем найти новые значения φ_{wt} и θ_{td}. И, наоборот, считая известными значения φ_{wt} и θ_{td} легко пересчитываются значения H_{dwt}. Таким образом мы на самом деле пришли к известному оптимизационному алгоритму — EM - алгоритму [10]. Алгоритм состоит из двух шагов — E (*expactation*) и M (*maximization*). В нашем случае на E-шаге вычисляется значения H_{dwt}, на M-шаге вычисляются новые значения φ_{wt} и θ_{td}.

Заметим также, что если начальные приближения θ_{td} и φ_{wt} положительны, то и после каждой итерации они будут оставаться положительными, несмотря на то, что ограничение неотрицательности было проигнорировано в ходе решения.

Вариант реализации описанного выше итерационного процесса представлен в алгоритме 2. Отличие состоит в том, что E-шаг встроен в M-шаг для уменьшения затрат памяти для хранения трехмерной матрицы H_{dwt}. Так как значения H_{dwt} используются только в момент просмотра документа d, то для всех остальных документов $d' \in D$ значения H_{dwt} не нужны. Стоит также отметить, что $\hat{n}_d = n_d$ для всех $d \in D$.

На больших коллекциях алгоритм 2 может сходиться очень медленно. Причина в том, что за однократный проход по всем документам коллекции оценки распределений терминов в темах $\varphi_{wt} = \hat{n}_{wt}/\hat{n}_t$ уточняются огромное число раз и успевают сойтись, в

Algorithm 2 PLSA-EM: EM-алгоритм для модели PLSA.

Вход: коллекция документов D, число тем $|T|$, начальные приближения Θ, Φ;

Выход: распределения Θ и Φ;

пока Θ и Φ не сойдутся
  обнулить $\hat{n}_{wt}$, $\hat{n}_{dt}$, $\hat{n}_t$ для всех $d \in D,\ w \in W,\ t \in T$
  для всех $d \in D,\ w \in d$
    $Z = \sum_{t \in T} \varphi_{wt}\theta_{td}$;
    для всех $t \in T$ таких, что $\varphi_{wt}\theta_{td} > 0$
      увеличить $\hat{n}_{wt}$, $\hat{n}_{dt}$, $\hat{n}_t$ на $\delta = n_{dw}\varphi_{wt}\theta_{td}/Z$;
  $\varphi_{wt} = \hat{n}_{wt}/\hat{n}_t$ для всех $w \in W,\ t \in T$
  $\theta_{td} = \hat{n}_{dt}/n_d$ для всех $d \in D,\ t \in T$

то же время распределения тем в документах θ_d проходят лишь одну итерацию. На начальных итерациях, пока распределения θ_d не сошлись, вычислительный ресурс тратится впустую на достижение сходимости φ_t к приближениям, далёким от оптимальных. Причина проблемы в том, что параметры θ_{td} привязаны к отдельным документам $d \in D$, а параметры φ_{wt} — ко всей коллекции.

Решить данную проблему можно путем реорганизации итерационного процесса. Проход каждого документа $d \in D$ производится несколько раз подряд. На каждом проходе документа выполняется E-шаг и обновляется распределение θ_d. Обновление распределений φ_t производится после каждого прохода коллекции. Алгоритм использующий данные идеи представлен в алгоритме 3. В результате распределения φ_t и θ_d сходятся более согласованно. Рассмотренные подходы к методу нахождения распределений θ_d и φ_t применимы и в случае иерархической тематической модели. Результаты представлены в разделе 7.

2.2 LDA

Одним из желательных свойств тематической модели является разреженность распределениея слов по темам и тем по документам. То есть каждая тема t описывается скорее всего небольшим числом терминов w, в типичных случаях порядка десятков или нескольких сотен. Термины, характерные для других тем, имеют малую вероятность встретиться в документе, относящегося к теме t. Также и документы относятся к небольшому числу тем, порядка несколько десятков.

В работе [4] для этого предлагается задавать априорные распределения для $p(w|t)$ и $p(t|d)$. Априорные распределения удобно выбирать из семейства распределений Дирихле,

Algorithm 3 PLSA-BatchEM: пакетный EM-алгоритм для модели PLSA.

Вход: коллекция документов D, число тем $|T|$;

Выход: распределения Θ и Φ;

инициализировать φ_{wt} для всех $w \in W$, $t \in T$;
пока Φ не сойдутся
  $\hat{n}_{wt} = 0$; $\hat{n}_t := 0$ для всех $w \in W,\ t \in T$;
  для всех $d \in D$
    инициализировать θ_{td} для всех $t \in T$;
    пока θ_d не сойдётся
      $Z_w = \sum_{t \in T} \varphi_{wt}\theta_{td}$ для всех $w \in d$;
      $\theta_{td} = \frac{1}{n_d} \sum_{w \in d} n_{dw}\varphi_{wt}\theta_{td}/Z_w$ для всех $t \in T$;
    увеличить $\hat{n}_{wt}$, $\hat{n}_t$ на $n_{dw}\varphi_{wt}\theta_{td}/Z_w$ для всех $w \in d,\ t \in T$;
  $\varphi_{wt} = \hat{n}_{wt}/\hat{n}_t$ для всех $w \in W,\ t \in T$;

так как оно является сопряженным к мультиномиальному распределению, что позволяет аналитически представлять апостериорное распределение. Однако при этом распределения не получаются в чистом виде разреженными: вероятность концентрируется лишь в небольшой части элементов, остальные имеют очень маленькие, но не нулевые вероятности.

Распределение Дирихле порождает нормированные случайные векторы, разреженностью которых управляет векторный параметр той же размерности.

Пусть $\theta_d \sim Dir(\alpha)$, то есть векторы документов $\theta_d \in R^{|T|}$ генерируется из распределения Дирихле с параметром $\alpha = (\alpha_t)_T$:

$$Dir(\theta_d; \alpha) = \frac{\Gamma(\alpha_0)}{\prod\limits_{t \in T} \Gamma(\alpha_t)} \prod_{t \in T} \theta_{td}^{\alpha_t - 1},\ \alpha_t > 0,\ \alpha_0 = \sum_{t \in T} \alpha_t,\ \theta_{td} > 0,\ \sum_{t \in T} \theta_{td} = 1,$$

где $\Gamma()$ — гамма-функция.

Пусть также $\varphi_t \sim Dir(\beta)$, то есть векторы тем $\varphi_t \in R^{|W|}$ генерируется из распределения Дирихле с параметром $\beta = (\beta_w)_W$:

$$Dir(\varphi_t; \beta) = \frac{\Gamma(\beta_0)}{\prod\limits_{w \in W} \Gamma(\beta_W)} \prod_{W \in W} \varphi_{wt}^{\beta_w - 1},\ \beta_w > 0,\ \beta_0 = \sum_{w \in W} \beta_w,\ \varphi_{wt} > 0,\ \sum_{w \in W} \varphi_{wt} = 1,$$

Алгоритм порождения коллекции документов в вероятностной модели LDA представлен в алгоритме 4.

Для отыскания матриц Φ, Θ воспользуемся методом максимизации совместного прав-

Algorithm 4 Алгоритм порождения коллекции текстовых документов с помощью вероятностной модели LDA.

Вход: Параметры α, β, количество тем T;

Выход: Выборка пар $(d_i, w_i), i = 1, \ldots, n$;

для всех $t \in T$

  Сгенерировать вектор тем $\varphi_t \sim Dir(\beta)$;

для всех $d \in D$

  Задать длину n_d документа d;

  Сгенерировать вектор документа $\theta_d \sim Dir(\alpha)$;

  для всех $i = 1, \ldots, n_d$

    Выбрать случайную тему t из распределения θ_d;

    Выбрать случайный термин w из распределения φ_t;

    Добавить в выборку пару (d, w);

доподобия $p(D, \Phi, \Theta; \alpha, \beta)$ при фиксированных значениях параметров α и β:

$$p(D, \Phi, \Theta; \alpha, \beta) = p(D|\Phi, \Theta)p(\Theta|\alpha)p(\Phi|\beta) \to \max_{\Phi,\Theta}$$

Эту задачу принято называть задачей максимизации апостериорной вероятности. В предположении о независимости распределений столбцов матриц Φ, Θ задача оптимизации можно записать в следующем виде:

$$\prod_{d\in D}\prod_{w\in d} p(d,w)^{n_{dw}} \prod_{d\in D} Dir(\theta_d;\alpha) \prod_{t\in T} Dir(\varphi_t;\beta) \to \max_{\Phi,\Theta}$$

Прологарифмировав произведение и отбросив константные слагаемые, не зависящие от матриц Φ, Θ, получим:

$$L(D, \Phi, \Theta; \alpha, \beta) = \sum_{d\in D}\sum_{w\in d} n_{dw} \ln \sum_{t\in T} \varphi_{wt}\theta_{td} +$$

$$+ \sum_{d\in D}\sum_{t\in T} (\alpha_t - 1) \ln \theta_{td} + \sum_{w\in W}\sum_{t\in T} (\beta_w - 1) \ln \varphi_{wt} \to \max_{\Phi,\Theta}$$

Для решения этой задачи также можно воспользоваться EM-алгоритмом.

Особой популярностью в последнее время получили методы поиска приближенного апостериорного распределения, так называемые методы байесовского вывода. Для вероятностной тематической модели LDA предложены алгоритмы на основе сэмплирования Гиббса [4, 28, 37] и вариационный подход [14, 31] для оценивания вероятностных распределений $p(w|t)$ и $p(t|d)$. При этом формулы пересчета для величин θ_{td} и φ_{wt} выглядят

следующим образом:

$$\theta_{td} = \frac{\hat{n}_{dt} + \alpha_t}{\hat{n}_d + \alpha_0}$$
$$\varphi_{wt} = \frac{\hat{n}_{wt} + \beta_w}{\hat{n}_t + \beta_0}$$

3 HDP

Одним из существенных недостатков тематических моделей PLSA и LDA является необходимость задания количества тем. То есть ни PLSA, ни LDA не способны автоматические определять число тем в коллекции документов. Для нахождения значение количества тем в документе часто многократно запускают данные модели с разным значением числа тем и по некоторым критериям качества определяют оптимальное число тем в коллекции. Однако также существует алгоритм, использующий иерархические процессы Дирихле (Hierarchical Dirichlet Processes, HDP) [32], для автоматического определения числа тем в коллекции документов. В данном разделе дано описание HDP и определение процессов Дирихле.

Другим значимым недостатком PLSA, LDA, а также и HDP, является то, что в данных моделях задается так называемая «плоская» тематическая модель. Достаточно естественной является ситуация, когда темы коллекции документов образуют некоторую иерархию. Например, тема «Машинное обучение» может разбиваться на две подтемы «Обучение без учителя» и «Обучение с учителем», а последняя в свою очередь может разбиваться, например, на «Классификацию» и «Регрессию». Также и эти темы в свою очередь могут разбиваться на подтемы и т.д. Описание иерархических вероятностных тематических моделей дается в разделе 5, критерии качества иерархических тематических моделей описаны в разделе 6.

3.1 Процесс Дирихле

Пусть (Ω, B, G_0) — вероятностное пространство. Пусть также α — положительное число.

Определение 3.1 *Процессом Дирихле с параметрами $\alpha \in R_+$ и G_0 ($DP(\alpha, G_0)$) называется вероятностное распределение G такое, что для любого конечного измеримого разбиения Ω на $A_1, \ldots, A_l$ вектор $(G(A_1), \ldots, G(A_l))$ распределен согласно распределению Дирихле с вектором параметров $(\alpha G_0(A_1), \ldots, \alpha G_0(A_l))$:*

$$(G(A_1), \ldots, G(A_l)) \sim Dir(\alpha G_0(A_1), \ldots, \alpha G_0(A_l)).$$

При этом G_0 называется базовым распределением для G.

В работе [11] было показано, что с вероятностью единица процесс Дирихле имеет дискретное распределение, даже в случае, если базовое распределение является непрерывным. Подход к конструктивному описанию процесса Дирихле дается в работе [25] и имеет название $stick - breaking$. Рассмотрим следующие независимо и одинаково рапределенные случайные величины:

$$p'_k \sim Beta(1, \alpha), \quad \psi_k \sim G_0, \quad k = 1, 2, \ldots$$

Тогда вероятностное распределение G, определенное следующим образом, имеет распределение процесса Дирихле $DP(\alpha, G_0)$:

$$G = \sum_{k=1}^{\infty} p_k \delta_{\psi_k}, \qquad p_k = p'_k \prod_{i=1}^{k-1} (1 - p'_i),$$

где δ_ψ является вероятностным распределением, сконцентрированное в точке ψ, а $Beta(a, b),\ a > 0, b > 0$ является бета-распределением. Плотность бета-распределения $Beta(a, b)$ выглядит следующим образом:

$$f(x) = \frac{1}{B(a, b)} x^{a-1}(1 - x)^{b-1},$$

где $B(a, b)$ — бета-функция:

$$B(a, b) = \int_0^1 x^{a-1}(1 - x)^{b-1} dx.$$

При этом $\sum\limits_{k=1}^{\infty} p_k = 1$ с вероятностью единица.

Важное свойство процесса Дирихле проявляется, если использовать для понимания схему урн Пойя. Такой подход предложен в работе [3]. Стоит отметить, что аналогия со схемой урн Пойя не позволяет конструктивно построить вид вероятностного распределения $G \sim DP(\alpha, G_0)$ как это делается с помощью $stick - breaking$, но позволяет выразить распределение случайных величин, которые сгенерированы из распределения G. Пусть $\tau_1, \ldots, \tau_{m-1}$ — случайные величины, сгенерированные из распределения G. Обозначим через $\psi_k, k = 1, \ldots, K$ — все уникальные значения, которые принимают величины $\tau_1, \ldots, \tau_{m-1}$. Пусть также n_k — число $\tau_i, i = 1, \ldots, m-1$, которые принимают значенияе ψ_k. Тогда случайная величина τ_m при условии $\tau_1, \ldots, \tau_{m-1}$ распределена согласно следующему распределению:

$$\tau_m | \tau_1, \ldots, \tau_{m-1} \sim \sum_{k=1}^{K} \frac{n_k}{m - 1 + \alpha} \delta_{\psi_k} + \frac{\alpha}{m - 1 + \alpha} G_0$$

Аналогия с урновой схемой следующая. Пусть в урне находится некоторая жидкость. При этом жидкость состоит из цветовых компонент. Объемы компонент распределены согласно базовому распределению G_0. При этом общий объем жидкости равен α. Из урны равновероятно извлекается единичный объем жидкости с одним цветом. После этого этот объем жидкости возвращается обратно в урну с дополнительным единичным объемом жидкости того же цвета. Соотвественно вероятность вытащить жидкость какого-то цвета пропорцианальна количеству раз, когда вытаскивалась жидкость этого же цвета. Вероятность вытащить жидкость нового цвета пропорцианальна α. Благодаря рассмотрению процессов Дирихле через схему урн Пойя становится понятно кластерное свойство процесса Дирихле. Чем меньше значение α, тем меньше цветов будет представлено в урне (будет небольшое число кластеров). Чем больше α, тем больше больше будет и цветов в урне (больше кластеров).

3.2 HDP

Идея использования процессов Дирихле для описание смеси распределений тем с неизвестным количеством тем заключается в следующем. Пусть есть базовое для всей коллекции документов распределение по тематикам. Обозначим его G_0. Тогда для каждого документа генерируется с помощью процессов Дирихле векторы документов $\theta_d \sim DP(\alpha, G_0)$. Тогда вектора документов θ_d будут представлять собой вероятностные распределения на схожих наборах тематик, которые представлены в G_0. При генерации тем для каждого термина в документе будут выбираться либо тема, уже выбранная для каких-то слов этого же документа, либо же будет сгенерирована новая тема. То есть θ_d в общем случае являются бесконечномерными. Соответственно с помощью процессов Дирихле удается избежать априорного задания числа тем для коллекции документов. При этом зачастую в качестве тем для документов будут выбираться темы, которые уже есть в коллекции, тем самым достигается общий для всех документов набор тематик в коллекции.

Однако вместо задания числа тем возникает необходимость задания базового распределения G_0. Один из вариантов определения G_0 является его определение также с помощью процессов Дирихле. Пусть $G_0 \sim DP(\gamma, H)$, где H – некоторое вероятностное распределение. Получаем тем самым иерархическую байесовскую систему, которое получила название иерахические процессы Дирихле (HDP). При этом в качестве распределения H принято брать само распределение Дирихле. Алгоритм порождения коллекции документов описан в алгоритме 5.

Algorithm 5 Алгоритм порождения коллекции текстовых документов с помощью вероятностной модели HDP.

Вход: Параметры $\alpha, \beta, \gamma, \eta$;

Выход: Выборка пар $(d_i, w_i), i = 1, \ldots, n$;

Сгенерировать $H \sim Dir(\eta)$;
Сгенерировать $G_0 \sim DP(\gamma, H)$;
для всех $d \in D$
    Задать длину n_d документа d;
    Сгенерировать векторы документов $\theta_d \sim DP(\alpha, G_0)$;
    для всех $i = 1, \ldots, n_d$
        Выбрать случайную тему t из распределения θ_d;
        если Тема t является новой **то**
            Сгенерировать вектор темы $\varphi_t \sim Dir(\beta)$;
        Выбрать случайный термин w из распределения φ_t;
        Добавить в выборку пару (d, w);

Существуют алгоритмы вывода приближенного апостериорного распределения в данной модели на основе сэмплирования Гиббса [12, 32] и на основе вариационного метода [33]. Более подробное описание HDP дано в работах [32, 34].

4 Оценки качества плоских тематических моделей

4.1 Перплексия и правдоподобие

Оценивание качества тематических моделей является нетривиальной задачей. В отличие от классических задач машинного обучения, таких как классификация и регрессия, в вероятностном тематическом моделировании отсутствуют понятия «ошибки». Также применение стандартных критериев качеств для задач кластеризации, таких как среднее внутрикластерное и среднее внекластерное расстояние плохо применимы, так как тематическая модель осуществляет «мягкую» кластеризаию.

Поэтому критерии качества оценивания тематических моделей все еще остаются темой активных исследований. Наиболее распростроненной оценкой качества является перплексия, которая активно применяется для оценивания качества языковых моделей в компьютерной лингвистике. Перплексия зачастую измеряется на контрольной выборке и име-

ет следующий вид:

$$Perplexity(D, p) = \exp\left(-\frac{1}{n}\sum_{d\in D}\sum_{w\in d} n_{dw}\ln p(w|d)\right)$$

Чем меньше величина перплексии, тем лучше построенная вероятностная модель p способна объснять появление терминов w в еще не рассмотренном документе d. Перплексия обратна среднему геометрическому правдободию слов. Общее правдоподобия контрольных документов имеет следующий вид:

$$Likelihood(D, p) = \prod_{d\in D}\prod_{w\in d} p(w|d)$$

Однако для контрольной выборки неизвестны значения $p(w|d)$. Для подсчета перплексии и правдоподобия документ разделяют на две половины. По первой половине оценивают профили документов θ_d, при этом профили тем φ_t считаются фиксированными. По второй половине документа оценивается собственно перплексия или правдоподобие. Правильное разделение документов на две части для получения несмещенных оценок перплексии и правдоподобия остается открытым вопросом. В работе [36] дается обзор методов для приближенной оценки правдободия контрольной выборки документов. Перплексия и правдоподобие могут быть хорошимим оценками для сравнения различных моделей между собой и для выбора параметров модели, таких как, например, количетсво тем. Однако они плохо подходят для измерения качества внутреннего представления тематической модели.

Одним из важных качеств тематической модели является интерпретируемость тем с точки зрения человека. В работе [8] было показано, что модели, достигающие лучшего значения перплексии зачастую содержат менее интерпретируемые профили тем и документов. Для этого был поставлен эксперимент с участием добровольцев для оценки качества построенных профилей. Для измерения качества профилей тем добровольцам предлагалось шесть терминов, перемешанных в случайном порядке. Необходимо было исключить из этих шести терминов один термин, который плохо семантически соотносился с остальными пятью терминами. Идея эксперимента следующая: если пять терминов семантически близки друг к другу, то исключение лишнего шестого термина, при условии его семантического дальности от остальных терминов, будет простой задачей. Если же шестой термин близок к остальным пяти или эти пять будут далеки друг от друга, то испытуемые будут исключать не только этот шестой термин, но и другие термины. Тем самым показывая, что эти пять терминов не образуют семантически единую группу. Для эксперимента брались наиболее вероятные пять терминов из профиля случайно выбранной темы, а в качестве шестого слова использовался термин, с маленькой вероятностью

в данной теме, но с большой вероятность в другой теме. Аналогичным образом исследовалось качество построенных профилей документов. Для этого добровольцам предоставлялась информация о названии документа, а также некоторый отрывок из него. Также добровольцам предлагались четыре темы, представленые в виде списка восьми наиболее вероятных слов в каждой из них. При этом три темы имели высокое значение вероятности в профиле документа, а четвертая низкое. Испытуемым необходимо было найти лишнюю тему.

После этой работы начали активно проводиться исследования по поиску альтернативных перплексии и правдободобию способов измерения качества тематических моделей.

4.2 Ранжирование профилей тем

В работах [2, 18, 20, 21] предлагается автоматически ранжировать темы по их качеству. В [2] в роли метрики качества используется взвешенное расстояние профиля темы до заведомо «плохих» профилей. В роли «плохих» профилей рассматривается равномерное распределение на словаре (в силу закона Ципфа только небольшая часть терминов должна иметь высокое значение вероятности), эмпирическое распределение терминов в коллекции (предполагается, что тема должна быть уникальной, а не представляться в виде смеси других тем), равномерное распределение темы в документах (предполагается, что тема должна относиться только к небольшему числу документов). В качестве расстояния могут использоваться дивергенция Кульбака-Лейбнера, косинусная мера и коэффициент корреляции. В работах [20, 21] предлагается использовать внешние источники, такие как Wikipedia, Google, Medline, Wordnet для оценки качества построенных тем. При этом качество темы оценивается как среднее или медиана расстояний первых 10 самых вероятных терминов в теме между собой. Наиболее хорошие результаты показала метрика PMI (pointwise mutual information).

$$\text{PMI-score}(\mathbf{w}) = \sum_{i,j \in \{1,\ldots,10\}, i<j} \text{PMI}(w_i, w_j)$$

$$\text{PMI}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)},$$

где $\mathbf{w} = (w_1, w_2, \ldots, w_{10})$ - первые 10 наиболее вероятных термина в профиле темы. Где $p(w_i, w_j)$ — совместная встречаемость пары терминов в w_i, w_j в некотором внешнем источнике. Эта метрика позволяет оценить, насколько точно в профиле тем термины согласованы друг с другом относительно общего употребления.

В [18] предлагается мера, оценивающая насколько точно и полно в профиле тем учитывается совместное встречаемость терминов в коллекции документов. Выглядит она следующим образом:

$$C(t,(w)) = \sum_{m=2}^{M} \sum_{i=1}^{m-1} \log \frac{D(w_m, w_i) + 1}{D(w_i)},$$

где $\mathbf{w} = (w_1, w_2, \ldots, w_M)$ - первые M наиболее вероятных термина в профиле темы, $D(w)$ - количество документов, содержащих термин w, $D(w, v)$ - количество документов, содержащих термины w и v.

4.3 Критерии, проверяющие гипотезу условной независимости

Гипотеза условной независимости $p(w|d,t) = p(w|t)$ чрезвычайно важна для вероятностных тематических моделей. Для её проверки не требуется выделять контрольную выборку, что является преимуществом данного типа критериев. Вероятностные распределения оцениваются следующим образом:

$$p(w|d,t) = \frac{n_{dwt}}{n_{dt}}$$

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

Рассмотрим статистические тесты, проверяющие нулевую гипотезу о том, что различия между этими распределениями незначимы, точнее, что выборка с эмпирическим распределением $p(w|d,t)$ могла быть получена из генеральной совокупности с распределением $p(w|t)$. Ожидаемое число вхождений термина w в документ d, связанных с темой t:

$$E_{dwt} = n_{dt} p(w|t)$$

Критерий X^2 Пирсона основан на вычислении статистики хи-квадрат, которая является естественной мерой различия двух распределений:

$$X^2_{dt} = \sum_{w \in W_{dt}} \frac{(E_{dwt} - n_{dwt})^2}{E_{dwt}} = n_{dt} \sum_{w \in W_{dt}} \frac{(p(w|t) - p(w|d,t))^2}{p(w|t)},$$

где $W_{dt} = \{w \in W : E_{dwt} > 0\}$.

Условием применимости асимптотики χ^2_k считается наличие достаточного числа наблюдений во всей выборке, $n_{dt} > 50$, а также достаточного ожидаемого числа наблюдений каждого термина $E_{dwt} > 5$. Второе требование в типичном случае не выполняется для большинства терминов w, так как распределение $p(w|t)$, как правило, разрежено, более того, мощность словаря W_{dt} может превышать длину документа n_{dt}. Таким образом, в

нашем случае критерий Пирсона применять нельзя. Для случая разреженных распределений больше подходят статистики G^2 и D^2.

Статистика G^2 определяется через дивергенцию Кульбака–Лейблера:

$$G^2_{dt} = 2 \sum_{w \in W_{dt}} n_{dwt} \ln \frac{n_{dwt}}{E_{dwt}}$$

Статистика D^2 — это поправка к статистике X^2 для случая разреженных распределений:

$$D^2_{dt} = \sum_{w \in W_{dt}} \frac{(E_{dwt} - n_{dwt})^2 - n_{dwt}}{E_{dwt}}$$

Эта статистика имеет асимптотически нормальное распределение.

В работе [19] предлагается еще один критерий для проверки условной независимости, основанный на дивергенции Кульбака-Лейблера:

$$KL_t = KL(p(d,w|t)||p(d|t)p(w|t)) = \sum_{d,w} \frac{n_{dwt}}{n_t} \ln \frac{n_{dwt}}{E_{dwt}}$$

.

4.4 Устойчивость модели

В работах [29, 35] был предложен критерий так называемой устойчивости вероятностной тематической модели. Если на выходе алгоритма построения тематической модели получаются значимые профили тем и профили документов, то эти же профили также должны появляться, если запускать алгоритм с другим начальным приближением. Для измерения соответствия профилей друг другу (устойчивости) каждому профилю, получившемуся при одной начальной инициализации, ставится в соответствие профиль, получившийся при другой инициализации. Для этого строится матрица попарных расстояний между профилями и с помощью алгоритма Манкреса (модификация Венгерского алгоритма для задачи о назначениях) находятся соответствующие друг другу пары профилей, минимизирующие общую сумму расстояний. Чем меньше эта сумма, тем более устойчивой является модель. Те профили, для которых при различных начальных инициализациях находятся близкие по расстоянию профили считаются устойчивыми, что может говорить о правильном их восстановлении по коллекции документов.

5 Методы построения иерархических вероятностных тематических моделей

В современных исследованиях по иерархическим тематическим моделям можно выделить два основных подхода для построения иерархии тем документов. Первый подход предполагает, что существование иерархии заложено в порождающую модель коллекции документов. То есть с использованием иерархии тем происходит генерация документов. Например, в hLDA [6] предполагается, что все темы документа расположены на одном из путей иерархии, при этом иерархия представляет собой дерево с явно выделенным корнем. Каждый термин документа принадлежит одной из тем этого пути. В модифицированном варианте данного подхода nHDP [7] уже для каждого термина в документе выбирается один из путей в иерархии (дереве). При этом каждый документ имеет некоторое распределение на путях и соответственно путь для каждого термина выбирается согласно распределению на путях для всего документа. Такое изменение позволяет документу относится к заведомо более широкому набору тем, чем в hLDA.

Второй подход используют «плоскую» тематическую модель для построения иерархии. При этом не предполагается, что иерархия тематик заложена в порождающую модель коллекции документов. Методы для построения иерархии зачастую используют методы, схожие с иерархической кластеризацией. Данный подход активно использует свойства HDP и LDA с различным значением параметров для распределения Дирихле.

5.1 hLDA

Модель hLDA (hierarchical LDA) описана в работах [5, 6]. Иерархия тем коллекции документов представляется в виде дерева, у которого более абстрактные темы располагаются ближе к корню, а более специфичные — ближе к листьям. В данном алгоритме не накладывается никаких ограничений на значение максимальной глубины дерева, на значения максимального количества потомков у вершины. При этом каждая вершина дерева представляет собой тему. Также вершина дерева содержит вероятности перехода в дочерние вершины. В силу того, что дерево не ограничено по глубине и по количеству потомков у вершин, то в качестве вероятностей перехода удобно использовать процессы Дирихле. Для каждой вершины задается порядок потомков по тому, как они появляются в дереве. Пусть z_{il} — вершины дерева, где l — глубина вершины, i — порядковый номер как потомка. Пусть z_{00} — корень дерева. Пусть также T — иерархия тем, а распределения

переходов в вершине z_{il} обозначаются через $G_{z_{il}}$. Так как порождаемое дерево является деревом бесконечной глубины, то для алгоритмической реализации вводится ограничение на максимальную глубину дерева $max\_depth \geq 0$, где $max\_depth = 0$ означает, что в порожденном дереве только одна вершина. Порождающий алгоритм для коллекции документов представлен в алгоритме 6.

Algorithm 6 Алгоритм порождения коллекции текстовых документов с помощью вероятностной модели hLDA.

Вход: Параметры $\alpha, \beta, \gamma, \eta$, макимальная глубина дерева max_depth, семейство дискретных распределений $Discrete$;

Выход: Выборка пар $(d_i, w_i), i = 1, \ldots, n$;

1: Сгенерировать $H \sim Dir(\eta)$;
2: Сгенерировать $G_0 \sim DP(\gamma, H)$;
3: Сгенерировать $G_{z_{00}} \sim DP(\alpha, G_0)$;
4: Сгенерировать вектор тем $\varphi_{z_{00}} \sim Dir(\beta)$;
5: **для всех** $d \in D$
6: Задать длину n_d документа d;
7: Сгенерировать путь $path_d = GenPath(T, G_0, depth, \alpha, \beta)$ в дереве T;
8: Выбрать распределение θ_d на вершинах пути $path_d$ из некторого семейства дискретных распределений;
9: **для всех** $i = 1, \ldots, n_d$
10: Выбрать случайную вершину пути c_k из распределения θ_d;
11: Выбрать случайный термин w из распределения φ_{c_k};
12: Добавить в выборку пару (d, w);

Авторы метода отмечают важную особенность построенной иерархии тем — внутренние вершины не представляют собой обобщение своих потомков. Скорее они задают общую терминологию для документов, сгенерированных по пути в дереве. То есть в вершине предка большие значения вероятности имеют термины, общеупотребительные для всех его потомков. У потомков же такие слова имеют маленькую вероятность. Выбор семейства дискретного распределения для генерации θ_d зависит от априорного предположения о тематиках тем. Если предполагается, что в документах часто встречаются общеупотребительные слова, то тогда веротности для первых вершин в пути должны быть большими. Если же наоборот, считается что в документах употребляются более специфичные слова, то тогда вероятности для последних вершин в пути должны быть большими. В статье [5]

Algorithm 7 Процедура генерации пути $GenPath(T, G_0, depth, \alpha, \beta)$.

Вход: Дерево T, априорное распределение G_0, максимальная глубина дерева max_depth, параметры α, β;

Выход: Путь в дереве $path$;

1: $j = 0$;
2: $c_j = z_{00}$;
3: **для** $j \leq max\_depth$
4: Сгенерировать потомка вершины c_j из распределения G_{c_j}. Обозначим его z_{ij+1};
5: **если** z_{ij+1} является новым потомком **то**
6: Сгенерировать $G_{z_{ij+1}} \sim DP(\alpha, G_0)$;
7: Сгенерировать вектор тем $\varphi_{z_{ij+1}} \sim Dir(\beta)$;
8: $c_{j+1} = z_{ij+1}$;
9: $j++$;
10: $path = (c_0, c_1, \ldots, c_{max\_depth})$;

описан вывод в данной модели на основе сэмплирования Гиббса, вариационный вывод описан в [6].

5.2 nHDP

Описание иерархической модели nHDP (nested hierarchical Dirichlet process) представлен в работе [7]. nHDP является обобщением hLDA: вместо выбора одного пути в иерархии для всех терминов в документе, в nHDP предлагается выбирать свой путь для каждого термина в документе. Соответсвенно nHDP позволяет словам документа принадлежать заведомо большему набору тем, чем в hLDA. Алгоритм генерации коллекции текстовых документов разбит на два шага: сначала генерируетя априорная иерархия для всей коллекции документов, а потом для каждого документа генерируется своя иерархия на основе априорной иерархии. Это делается с помощью генерации в вершинах дерева специфичных для документа вероятностей перехода в потомки $G_{z_{il}}^{(d)} \sim DP(\pi, G_{z_{il}})$. При этом предполагается, что априорная иерархия имеет бесконечную глубину и бесконечное число потомков у вершины. Соответственно для определения глубины иерархии для каждого документа в nHDP предлагается принимать решение о дальнейшей генерации новой вершины в пути стохастически. То есть для каждой вершины z_{il} иерархии дополнительно генерируется случайная величина $U_{z_{il}}^{(d)}$, принимающая значение на отрезке $[0; 1]$. Тогда с вероятностью $U_{z_{il}}^{(d)}$ процесс построения пути для слова прекращается в вершине z_{il}.

Алгоритм генерации коллекции текстовых документов описан в алгоритме 8.

Algorithm 8 Алгоритм порождения коллекции текстовых документов с помощью вероятностной модели nHDP.

Вход: Параметры $\alpha, \beta, \gamma, \eta, \lambda_1, \lambda_2, \pi$;

Выход: Выборка пар $(d_i, w_i), i = 1, \ldots, n$;

1: Сгенерировать $H \sim Dir(\eta)$;
2: Сгенерировать $G_0 \sim DP(\gamma, H)$;
3: Сгенерировать априорную иерархию тем T с помощью алгоритма 6 с достаточно большой глубиной иерархии;
4: **для всех** $d \in D$
5: Задать длину n_d документа d;
6: Сгенерировать T_d путем генерации $G_{z_{il}}^{(d)} \sim DP(\pi, _{z_{il}})$, $U_{z_{il}}^{(d)} \sim Beta(\lambda_1, \lambda_2)$ для всех вершин z_{il} иерархии T.
7: **для всех** $i = 1, \ldots, n_d$
8: Сгенерировать тему $t = GenTopic(T_d)$;
9: Выбрать случайный термин w из распределения φ_t;
10: Добавить в выборку пару (d, w);

Вывод в данной модели основан на стохастическом вариационным методе. Данный алгоритм позволяет эффективно обрабатывать большие коллекции документов.

5.3 hHDP

Алгоритм описан в работе [38]. Алгоритм базируется на использовании алгоритма HDP. На выходе алгоритма HDP образуется три матрицы: исходная матрица документы-термины, матрица темы-термины, документы-темы. При этом для HDP нет существенной разницы какую именно матрицу он использует, поэтому возникает идея строить иерархию с помощью HDP на матрицах темы-термины или документы-темы. Отсюда возникают два алгоритма, описанные ниже. Как и раньше обозначим через D — множество документов, через W — множестов терминов, T — множество тем. Тогда исходная коллекция документов представляется в виде матрицы документы-термины. Обозначим ее Ω. Как и раньше матрица темы-термины обозначается Φ, а документы-темы — Θ. Алгоритм hvHDP построения иерархии тем представлен в алгоритме 10. Алгоритм htHDP построения иерархии тем представлен в алгоритме 11. Алгоритм HDP параметризуется параметрами $\alpha, \gamma, Dir(\eta)$,

Algorithm 9 Процедура генерации темы $GenTopic(T)$.

Вход: Иерархия T;

Выход: Тема t;

1: $c = z_{00}$;

2: $stop \sim B(1, U_{z_{00}})$ — сгенировать величину $stop$ из биномального распределения $B(1, U_{z_{00}})$;

3: **пока** $stop \neq 1$

4: Сгенерировать потомка вершины из распределения G_c. Обозначим его z_{il};

5: $c = z_{il}$;

6: В качестве темы t взять тему, соответствующую вершине c;

как было описано выше:

$$H \sim Dir(\eta),\ G_0 \sim DP(\gamma, H),\ G_d \sim DP(\alpha, G_0)$$

Algorithm 10 Алгоритм построения иерархии тем по коллекции текстовых документов hvHDP.

Вход: Параметры α, γ, η;

Выход: Иерархия тем;

1: Применить алгоритм $HDP(Dir(\eta), \gamma, \alpha)$ к матрице Ω;

2: Обозначим через T_{cur} получившиеся множество тем, а через Φ_{cur} — получившуюся матрицу темы-термины;

3: **пока** $|T_{cur}| > 1$

4: Применить $HDP(Dir(\eta), \gamma, \alpha)$ к матрице Φ_{cur}. На выходе получаем новое множество тем T_{new} и новую матрицу Φ_{new};

5: Присваиваем $T_{cur} = T_{new},\ \Phi_{cur} = \Phi_{new}$;

На выходе hvHDP и алгоритма htHDP получаем ацикликлический граф с одной вершиной на первом уровне, в котором вершины каждого уровня соединены со всеми вершинами следующего уровня. Однако в hvHDP каждая вершина представляет собой тему — распределение на терминах. При этом для всех внутренних вершин имеем также распределения по темам, которые находятся на уровне ниже. То есть все ребра графа имеют вес, равный вероятности соответствующей подтемы в супертеме. В модели htHDP только листовые вершины являются темами. При этом каждая внутренняя вершина является «супертемой», которая представляет собой распределение по подтемам, находящихся на

Algorithm 11 Алгоритм построения иерархии тем по коллекции текстовых документов htHDP.

Вход: Параметры α, γ, η;

Выход: Иерархия тем;

1: Применить алгоритм $HDP(Dir(\eta), \gamma, \alpha)$ к матрице Ω;
2: Обозначим через T_{cur} получившиеся множество тем, а через Θ_{cur} — получившуюся матрицу документы-темы;
3: **пока** $|T_{cur}| > 1$
4: Применить $HDP(Dir(\eta), \gamma, \alpha)$ к матрице Θ_{cur}. На выходе получаем новое множество тем T_{new} и новую матрицу Θ_{new};
5: Присваиваем $T_{cur} = T_{new},\ \Theta_{cur} = \Theta_{new}$;

уровень ниже. То есть все ребра графа имеют вес, равный вероятности соответствующей подтемы в супертеме.

В обоих алгоритмах получаем для каждой супертемы рапределение по подтемам. Соответственно, мы можем удалять «слабые связи» в графе, то есть удалять ребра с весом ниже заданного порога. Тем самым мы можем получать несбалансированный граф, который может лучше соответствовать истинной иерархии тематик в коллекции. При этом значение порога предлагается определять по критерию качества, описанному в разделе 6.

Иерархия, построенная с помощью hvHDP, является более интерпретируемой, чем htHDP, так как каждая вершина представляет собой тему, а также содержит распредление по подтемам. А в же htHDP внутренние вершины не являются темами.

В качестве основного недостатка стоит отметить, что иерархия строится как некоторая иерархическая надстройка над плоской моделью. То есть существование связи «тема–подтема» не заложена в вероятностную модель порождения коллекции текстовых документов.

5.4 HDP-C

Иерархическая тематическая модель описана в работе [26]. Алгоритм использует свойства параметра распределения Дирихле $Dir(\alpha_1, \ldots, \alpha_n)$. Рассмотрим случай, когда $\alpha_1 = \cdots = \alpha_n = \eta$. При значении параметра $\eta = 1$ получаем равномерное распределение точек $(n-1)$-мерного симплекса. Если $\eta < 1$, то получается более концентрированное распределение, если же значение параметра больше 1, то получается более равномерное распределение. Алгоритм построения HDP-C представлен в алгоритме 12. В связи со свой-

ством распределения Дирихле возникла идея построить два уровня тем — первый уровень отвечает за более абстрактные темы коллекции текстовых документов, а второй — за более специфичные тем. Чтобы установить связи между темы двух уровней применяется аналог иерархической кластеризации. Алгоритм построения HDP-C представлен в алгоритме 12.

Algorithm 12 Алгоритм построения иерархии тем по коллекции текстовых документов HDP-C.

Вход: параметры $\alpha, \gamma, D(p,q)$ — метрика близости распределений p и q;

Выход: иерархия тем;

1: Строим темы нижнего уровня с помощью $HDP(Dir(\eta), \gamma.\alpha)$ с параметром $\eta = 0.125$, получившиеся темы $M = \{t_{l1}, \ldots, t_{l|M|}\}$;
2: Строим темы верхнего уровня с помощью $HDP(Dir(\eta), \gamma.\alpha)$ с параметром $\eta = 1$, получившиеся темы $N = \{t_{u1}, \ldots, t_{u|N|}\}$;
3: **для всех** $i = 1, \ldots, |M|$
4: $t_{uj} = Argmin_{t_{uj} \in N} D(t_{li}, t_{uj})$;
5: $t_{uj}.ChildList.add(t_{li})$ — добавляем в список подтем t_{uj} новую подтему t_{li};
6: $t_{uj}.nchild++$ — увеличиваем число подтем темы t_{uj};
7: **для всех** $i = 1, \ldots, |N|$
8: **пока** $t_{ui}.nchild > 3$
9: Находим наиболее близкую пару потомков (t_x, t_y) в $t_{ui}.ChildList$;
10: Сливаем (t_x, t_y) в новую тему t_m;
11: $t_{ui}.ChildList.remove(t_x)$ — удаляем из списка подтем t_{ui} старую подтему t_x;
12: $t_{ui}.ChildList.remove(t_y)$ — удаляем из списка подтем t_{ui} старую подтему t_y;
13: $t_{ui}.ChildList.add(t_m)$ — добавляем в список подтем t_{ui} новую подтему t_m;
14: $t_{uj}.nchild--$

При этом в описании метода не говорится, как происходит слияние тем.

6 Оценки качества иерархических тематических моделей

Большинство иерархических тематических моделей являются достаточно естественным расширением соответствующих плоских тематических моделей. Поэтому во многих статьях оценка качества иерархических моделей производилась путем сравнения с плоскими моделями или hLDA (которая была одной из первых иерархических моделей) с

помощью стандартных методов, а именно перплексии и правдоподобия. Также широко используются визуальная оценка качества построенных иерархий на предмет адекватности с точки зрения человека связей тема-подтема. В то же время, для оценки качества получившихся распределений для тем можно применять те же критерии, что описаны в разделе 4.

Один из возможных подходов к оцениванию качества иерархических тематических моделей описан в работе [39]. Оценка качества построенной иерархии предполагает наличие эталонной иерархии для заданной коллекции документов. Процесс оценки условно можно разделить на три этапа. На первом этапе восстанавливаются для заданной эталонной иерархии профили тем, если они изначально не доступны. На втором этапе происходит сопоставление тем из построенной и эталонной иерархии друг с другом. На третьем этапе происходит собственно оценка, состоящая в агрегировании разниц между профилями эталонных и соответствующих им оцененных тем.

Метод оценивания может быть описан следующим образом, при условии, что распределения на терминах для вершин эталонной иерархии уже заданы:

1. Сопоставить темы построенной иерархии $L = \{t_{l1}, \ldots, t_{l|L|}\}$ с темами эталонной иерархии $G = \{t_{g1}, \ldots, t_{g|G|}\}$. Сопоставлене происходит путем поиска наиболее близких пар тем из двух иерархий по одной из метрик близости для распределений. Назовем ее SimDist, значение метрики должно находиться в отрезке $[0; 1]$. При этом значение 0 соответствует равным распределениям. После сопоставления получаем M пар $m = (t_{li}, t_{gj})$ и им соответствующие значения метрики $SimDist_m$. При этом $M = \min(|L|, |G|)$.

2. Для каждой пары $m = (t_{li}, t_{gj})$ строим множества $CS(t_{li})$ и $CS(t_{gj})$, где множество $CS(t)$ — множество всех потомков и родителей вершины t. То есть $CS(t)$ — совокупность всех вершин на всех путях в иерархии, проходящих через вершину t.

3. Для каждой пары $m = (t_{li}, t_{gj})$ вычисляем

$$PCP_m = \frac{|CS(t_{li}) \cap CS(t_{gj})|}{|CS(t_{li})|}$$

$$PCR_m = \frac{|CS(t_{li}) \cap CS(t_{gj})|}{|CS(t_{gj})|},$$

где под $CS(t_{li}) \cap CS(t_{gj})$ понимается количество соответствий между темами в $CS(t_{li})$ и темами в $CS(t_{gj})$.

4. Вычисляем значения точности P, полноты R и F-меры F:

$$P = \frac{1}{|M|}\sum_{m=1}^{|M|}(1 - SimDist_m)PCP_m$$

$$R = \frac{1}{|M|}\sum_{m=1}^{|M|}(1 - SimDist_m)PCR_m$$

$$F = \frac{2PR}{P+R}$$

Соответственно, чем значения точности, полноты, F-меры больше, тем больше похожа построенная иерархии на эталонную.

Другой метод оценки качества иерархических моделей при наличии эталонный иерархии и кластеризованных документов по вершинам иерархии описан в статье [17]. Для оценки качества иерархии используется мера качества кластеризации $FScore$. Пусть дан некоторый эталонный кластер C_r размера n_r и построенный кластер S_i размера n_i. Пусть также n_{ri} документа кластера S_i принадлежат C_r, тогда $FScore$ для этой пары кластеров определяется следующим образом:

$$F(C_r, S_i) = \frac{2R(C_r, S_i)P(C_r, S_i)}{R(C_r, S_i) + P(C_r, S_i)}$$

где $P(C_r, S_i) = n_{ri}/n_i, R(C_r, S_i) = n_{ri}/n_r$. Для каждой эталонного кластера вершины определяется значение F как максимум из всех кластеров в вершинах, построенного дерева T:

$$F(C_r) = \max_{S_i \in T} F(C_r, S_i)$$

Тогда общее качество кластеризации определяется следующим образом:

$$FScore = \sum_{r=1}^{c}\frac{n_r}{n}F(C_r),$$

где c — общее число эталонных кластеров.

7 Иерархическая модель с частичным обучением для категоризации документов

7.1 Мотивация

В данной разделе рассматривается задача построения иерахической тематической модели по уже заданному иерархическому классификатору для автоматической классификации документов. Таким образом получившаяся иерархическая тематическая модель должна предоставлять как понятную интерпретацию профилей тем, так и связей в иерархии

вида «тема-подтема». Иерархическую модель, учитывающую существующий каталогизатор, предлагается строить схожим образом, что и плоскую модель PLSA, описанную в разделе 2.1.

7.2 Гипотеза о существовании тематического дерева.

Рассмотрим дерево с множеством вершин V и корнем $t_0 \in V$. Вершины дерева соответствуют темам. Каждой теме $t \in V$ соответствует множество её подтем — дочерних вершин в дереве $S_t \subset V$. Каждое ребро дерева соответствует паре «тема–подтема» (t, s), $s \in S_t$. Если $S_t = \emptyset$, то тема t называется терминальной или листом тематического дерева. Для каждой вершины t в дереве V существует только одна родительская вершина, следовательно, только один путь $(t_0, \ldots, t)$ от корня дерева t_0 до темы t.

В плоской модели PLSA предполагалось, что каждое вхождение термина w в документ d связано только с одной темой t. Теперь примем за аксиому другие предположения:

1. Если пара (d, w) связана с темой t, то она связана и со всеми темами выше вершины t на пути до корня t_0;

2. Если пара (d, w) не связана с темой t, то она не связана и со всеми подтемами в поддереве ниже вершины t.

Вероятностная интерпретация отношения «тема–подтема». Каждому ребру тематического дерева (t, s) соответствует условная вероятность $p(s|t)$ того, что термин документа, связанный с темой t, связан также с подтемой $s \in S_t$:

$$p(s|t) = \frac{p(t,s)}{p(t)} = \frac{p(s)}{p(t)}$$

Если рассматривать коллекцию документов как выборку троек (d, w, t), то частотной оценкой этой условной вероятности будет $\hat{p}(s|t) = n_s/n_t$ — доля троек, связанных с подтемой s, среди всех троек, связанных с темой t.

Условные вероятности подтем удовлетворяют ограничениям нормировки, которые допускают две эквивалентные записи:

$$\sum_{s \in S_t} p(s|t) = 1$$

$$\sum_{s \in S_t} p(s) = p(t)$$

Обозначим через T множество тем, соответствующих терминальным вершинам дерева V. Условие нормировки:

$$\sum_{t \in T} p(t) = 1$$

выполняется именно для этого множества, а не для всего множества тем в дереве V. Условие нормировки останется справедливым, если заменить любое из множеств S_t его родительской темой t, а также если делать такие замены многократно в произвольном порядке. В частности, для корневой темы $p(t_0) = 1$.

При разделении темы t на подтемы $s \in S_t$ условные распределения для подтем $\varphi_{ws} = p(w|s)$ и $\theta_{sd} = p(s|d)$ должны удовлетворять требованиям нормировки:

$$\sum_{w \in W} \varphi_{ws} = 1, \qquad s \in S_t;$$

$$\sum_{s \in S_t} \theta_{sd} = \theta_{td}, \qquad d \in D$$

Распределения $p(s|w) = \varphi_{ws}\frac{p(s)}{p(w)}$ и $p(d|s) = \theta_{sd}\frac{p(d)}{p(s)}$ также должны быть нормированы, откуда следуют ещё две серии тождеств:

$$\sum_{s \in S_t} \varphi_{ws} p(s) = \varphi_{wt} p(t), \qquad w \in W;$$

$$\sum_{d \in D} \theta_{sd} p(d) = p(s), \qquad s \in S_t$$

7.3 Иерархическая модель для категоризации текстов

Рассмотрим случай, когда структура тематического дерева $\{S_t \colon t \in V\}$ фиксирована. Чтобы каждая тема $t \in T$ имела интерпретацию, к ней привязывается множество документов $D_t \subset D$ и множество терминов $W_t \subset W$. Одно из этих множеств может быть пустым. Таким образом, ставится задача частичного обучения (semi-supervised learning) иерархической тематической модели.

Распределение $\theta_{td} = p(t|d)$, полученное в результате тематического моделирования, непосредственно решает задачу категоризации — документ d относится к тем темам (категориям), для которых вероятность θ_{td} превышает заданный порог. Для решения задач категоризации лучше подходят разреженные модели, в которых малые вероятности θ_{td} обнуляются в процессе построения модели. Обнуление или игнорирование малых вероятностей в уже построенной модели может приводить к менее адекватным результатам.

Иерархический алгоритм 13 основан на пакетном алгоритме 3. Основное отличие PLSA-HEM в том, что для вычисления распределения θ_{td} тем t в документе d производится спуск по дереву от корня к терминальным вершинам.

Модификация формул М-шага для иерархической модели. Пусть $T \subset V$ — подмножество вершин дерева, удовлетворяющее условию нормировки $\sum\limits_{t\in T} p(t) = 1$, и для всех тем $t \in T$ известны значения параметров φ_{wt}, θ_{td}. Сначала T состоит из единственной корневой вершины t_0, для которой $\varphi_{wt_0} = p(w)$ и $\theta_{t_0 d} = 1$. Спуск по дереву — это итерационный процесс, на каждом шаге которого выбирается некоторое подмножество тем $R \subseteq T$, и для каждой вершины $s \in S$ из множества всех их подтем $S = \bigcup\limits_{t\in R} S_t$, вычисляются значения параметров φ_{ws}, θ_{sd}. После этого множество T заменяется на $(T\backslash R) \cup S$ и начинается следующий шаг. Спуск продолжается, пока T не совпадёт с множеством терминальных вершин дерева.

Рассмотрим вероятностную модель

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

при ограничениях нормировки

$$\sum_{w\in W} \varphi_{ws} = 1, \qquad s \in S_t;$$

$$\sum_{s\in S_t} \theta_{sd} = \theta_{td}, \qquad d \in D, \qquad t \in T.$$

Параметры φ_{wt} и θ_{td} для всех тем $t \in T\backslash R$ будем считать фиксированными. Обозначим через σ_{dw} фиксированную часть вероятностной тематической модели:

$$\sigma_{dw} = \sum_{t\in T\backslash R} \varphi_{wt}\theta_{td}, \qquad d \in D,\ w \in d.$$

Задача оценивания параметров φ_{ws}, θ_{sd}, $s \in S$ сводится к максимизации логарифма правдоподобия, аналогично задаче для плоской модели PLSA, но оптимизируется только часть параметров $\Phi_S = (\varphi_{ws})_{W\times S}$ и $\Theta_S = (\theta_{sd})_{S\times D}$, связанных с темами из S:

$$L(D; \Theta_S, \Phi_S) = \sum_{d\in D}\sum_{w\in d} n_{dw} \ln\left(\sigma_{dw} + \sum_{s\in S} \varphi_{ws}\theta_{sd}\right) \to \max_{\Phi_S,\Theta_S};$$

$$\sum_{w\in W} \varphi_{ws} = 1, \qquad s \in S;$$

$$\sum_{s\in S_t} \theta_{sd} = \theta_{td}, \qquad t \in R,\ \ d \in D.$$

Для решения оптимизационной задачи нужно записать лагранжиан, приравнять к нулю его производные по переменным φ_{ws} и θ_{sd}, из полученных уравнений исключить

двойственные переменные и выразить φ_{ws} и θ_{sd} через H_{dws}:

$$H_{dws} = \frac{\varphi_{ws}\theta_{sd}}{\sigma_{dw} + \sum\limits_{s' \in S} \varphi_{ws'}\theta_{s'd}}, \qquad d \in D,\ \ w \in d,\ \ s \in S;$$

$$\varphi_{ws} = \frac{\sum\limits_{d \in D} n_{dw} H_{dws}}{\sum\limits_{w' \in W} \sum\limits_{d \in D} n_{dw'} H_{dw's}}, \qquad w \in W,\ \ s \in S;$$

$$\theta_{sd} = \theta_{td} \frac{\sum\limits_{w \in d} n_{dw} H_{dws}}{\sum\limits_{s' \in S_t} \sum\limits_{w' \in d} n_{dw'} H_{dw's'}}, \qquad d \in D,\ \ s \in S_t,\ \ t \in R;$$

В более компактной записи можно записать следующим образом:

$$\varphi_{ws} = \frac{\hat{n}_{ws}}{\hat{n}_s}, \qquad \hat{n}_s = \sum_{w \in W} \hat{n}_{ws}, \qquad \hat{n}_{ws} = \sum_{d \in D} n_{dw} H_{dws};$$

$$\theta_{sd} = \theta_{td} \frac{\hat{n}_{ds}}{\hat{n}_{dt}}, \qquad \hat{n}_{dt} = \sum_{s \in S_t} \hat{n}_{ds}, \qquad \hat{n}_{ds} = \sum_{w \in d} n_{dw} H_{dws}.$$

Таким образом, формулы M-шага и E-шага для иерархического алгоритма лишь немногим отличаются от обычного PLSA-EM.

7.4 Документы во внутренних вершинах.

В некоторых приложениях важно, чтобы документы и термины могли относиться не только к терминальным вершинам, но и к любым внутренним вершинам тематического дерева. В частности, это могут быть документы, относящиеся сразу к нескольким подтемам, либо новые документы, которые пока не выделились в отдельную подтему.

Для каждой внутренней вершины $t \in V \setminus T$ создаётся выделенная терминальная вершина — подтема $s_0 \in S_t$. Если документ или термин попадает в s_0, то считается, что он остался в теме t. Алгоритм 13 остается фактически без изменений.

В терминах кластеризации выделенная подтема s_0 — это специальный «фоновый» кластер, к которому относится всё, что не удалось с уверенностью отнести к другим кластерам — подтемам темы t.

8 Категоризация документов

В последние 20 лет объем поступающей с каждым годом информации растет экспоненциально. Особенно это касается текстовый информации, которая широко представлена новостными сообщениями, записями в блогах, короткими сообщениями в социальных сетях. Поэтому становится особенно важным хранить информацию в структурированном

Algorithm 13 PLSA-batchHEM: иерархический пакетный EM-алгоритм.

Вход: коллекция документов D; параметры λ и μ,

множество тем V и структура тематического дерева $\{S_t \colon t \in V\}$,

привязки терминов и документов к темам φ^0_{wt} и θ^0_{td};

Выход: распределения Θ и Φ;

инициализировать φ_{wt} с учётом φ^0_{wt} для всех $w \in W$, $t \in V$

пока Φ не сойдутся

  $\hat{n}_{wt} = 0$; $\hat{n}_t = 0$ для всех $w \in W$, $t \in V$

  для всех $d \in D$

    инициализировать θ_{td} с учётом θ^0_{td} для всех $t \in V$

    $T = \{t_0\}$; $R = \{t_0\}$; $\theta_{t_0 d} = 1$

    пока множество подтем $S := \bigcup\limits_{t \in R} S_t$ не пусто

      $\sigma_{dw} = \sum\limits_{t \in T \setminus R} \varphi_{wt}\theta_{td}$ для всех $w \in d$

      пока θ_{sd} не сойдутся для всех $s \in S$

        $Z_w = \sigma_{dw} + \sum\limits_{s \in S} \varphi_{ws}\theta_{sd}$ для всех $w \in d$

        $n_s = \sum\limits_{w \in d} n_{dw}\varphi_{ws}\theta_{sd}/Z_w$ для всех $w \in d$

        $n := \sum\limits_{s \in S} n_s$

        $\theta_{sd} := \mu\theta^0_{sd} + (1-\mu)\theta_{td} n_s/n$ для всех $s \in S_t$, $t \in R$

      увеличить $\hat{n}_{ws}$, $\hat{n}_s$ на $n_{dw}\varphi_{ws}\theta_{sd}/Z_w$ для всех $w \in d$, $s \in S$

      $T = (T \setminus R) \cup S$; $R = S$

  $\varphi_{wt} := \lambda\varphi^0_{wt} + (1-\lambda)\hat{n}_{wt}/\hat{n}_t$ для всех $w \in W$, $t \in V$

виде, таком как каталоги, в котором удобно искать нужную и актуальную информацию в короткие сроки. Также важным становиться автоматическая каталогизация новых текстовых документов согласно существующим каталогам. В последние время активно развиваются автоматической категоризации документов на основе методов машинного обучения [15, 16, 24]. Вероятностные тематические модели также оказались полезны для автоматической категоризации документов [23, 27, 30].

В данной работе основным предметом исследования является иерархическая категоризация документов, когда некоторые категории могут вложенными в другие категории. Формально задача иерархической категоризации документов ставится следующим образом. Пусть $D = \{d_1, \ldots, d_n\}$ — множество документов, $C = \{c_1, \ldots, c_m\}$ — множество категорий. Задача категоризации заключается в построении функции $f : D \times C \to \{0, 1\}$. Если $f(d_i, c_j) = 1$, то это означает, что документ d_i относится к категории c_j. Если же

$f(d_i, c_j) = 0$, то документ d_i не относится к категории c_j. Также известна информация о вложенности категорий друг в друга. То есть задана функция $g : C \times C \to \{0, 1\}$, такая что, если $g(c_k, c_j) = 1$, то категория c_k вложена в c_j, если $g(c_k, c_j) = 0$, то c_k не вложена в c_j. При этом выполнено условие: если $g(c_k, c_j) = 1$ и $g(c_l, c_j) = 1$, то либо $g(c_k, c_l) = 1$, либо $g(c_l, c_k) = 1$. Таким образом категории образуют иерархию, причем эта иерархия представляется в виде дерева. Задача состоит в том, чтобы по набору значений пар (d_i, c_j), $d_i \in D, c_j \in C$ восстановить функцию f на всем пространстве $D \times C$ при заданной функции вложенности g.

8.1 Оценка качества

Качество категоризации обычно измеряется в привычных терминах для информационного поиска точности (precision) и полноты (recall) [24]. Введем следующие обозначения. Пусть $TP_j(true\ positive)$ — количество правильно категоризированных документов для категории c_j, $FP_j(false\ positive)$ — количество неправильно категоризированных документов для категории c_j. Пусть также FN_j (false negative) — количество документов, не отнесенных к c_j, но которые в дейсвительности принадлежат c_j. Точность для рубрики c_j тогда измеряется по следующей формуле:

$$P_j = \frac{TP_j}{TP_j + FP_j}.$$

Полнота:

$$R_j = \frac{TP_j}{TP_j + FN_j}.$$

Зачастую дополнительно к этим двум метрикам используется так называемая F-мера, как единая метрика, объединяющая и точность, и полноту. Обычно F-мера вычисляется следующим образом:

$$F_j = \frac{2P_jR_j}{P_j + R_j}.$$

При этом общее качество по всем категориям приняты два типа усреднения: микроусреднение и макроусреднение. Микроусреднение:

$$P_{micro} = \frac{TP}{TP + FP} = \frac{\sum\limits_{c_j \in C} TP_j}{\sum\limits_{c_j \in C} TP_j + FP_j};$$

$$R_{micro} = \frac{TP}{TP + FN} = \frac{\sum\limits_{c_j \in C} TP_j}{\sum\limits_{c_j \in C} TP_j + FN_j}.$$

Макроусреднение:

$$P_{macro} = \frac{\sum\limits_{c_j \in C} P_j}{|C|};$$

$$R_{macro} = \frac{\sum\limits_{c_j \in C} R_j}{|C|}.$$

Эти два типа могут давать различные результаты, особенно если категории имеют различный размер по количеству документов к ним относящихся. В некоторых приложениях предпочтительно микроусреднение, так как есть категории с малым количеством документов и они сильней влияют на макроусреденение, чем на микроусреднение. Также в дальнейшем будем использовать значения точности и полности не только для категорий, но также отдельно для различных уровней иерархии тем.

8.2 Описание данных

В качестве исходных данных использовалась коллекция русскоязычных документов. Документам приписывались категории из УДК (универсальный десятичный классификатор). При этом коллекция состояла из 1320 документов с разметкой по первым двум уровням УДК. Каждый документ был подвержен стандартным процедурам лемматизации, отбрасывания стоп-слов и отбрасывания редких слов. Под лемматизацией понимается приведение каждого слова в документе к его нормальной форме. В русском языке нормальными формами считаются: для существительных — именительный падеж, единственное число; для прилагательных — именительный падеж, единственное число, мужской род; для глаголов, причастий, деепричастий — глагол в инфинитиве. В качестве стоп-слов брались предлоги, союзы, числительные, местоимения. Также отбрасывались слова, которые встеречались в документе один раз и те слова, которые встречались менее, чем в 20 документах коллекции. В итоге был создан словарь размером в 52927 слов.

Иерархия категорий представляет собой двухуровневое дерево. На первом уровне располагаются 9 категорий, на втором 59. Соответственно всего 68 категорий. Для каждого документа предоставлялось множество категорий, к которым он относится. При этом это множество является корректным, в том смысле, что оно образует связное поддерево категорий. Выборка документов была случайным образом разделена на обучающую и контрольную выборки. В обучающую выборку вошли 1054 документа, в контрольную соотвественно 266 документов. На рис. 1 представлены гистограмма распределений документов по категориям для всей выборки целиком, а также для обучающей и контрольной выборок. Красным цветом обозначены категории первого уровня, синим цветом категории

второго уровня. Из графиков видно, что в целом, все три выборки имеют одинаковое распределение по категориям. На рис. 2 представлено распределение количества категорий по документам.

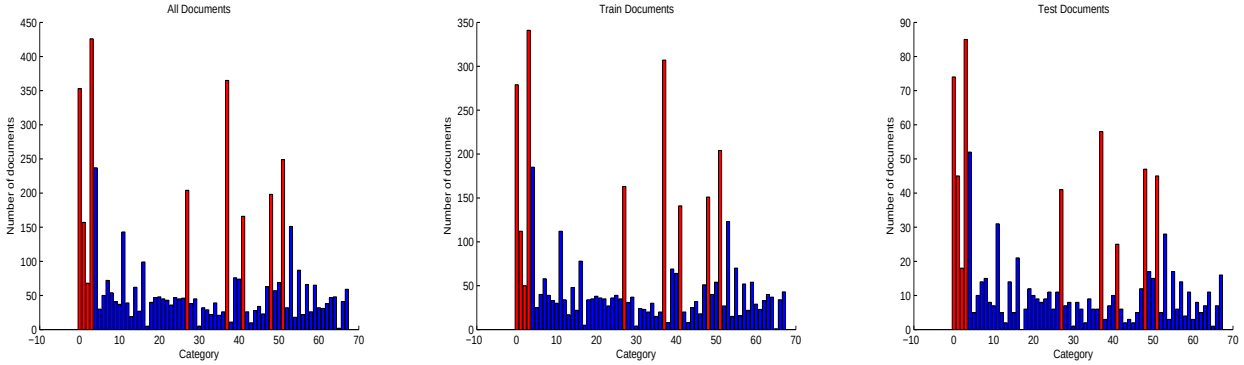


Рис. 1: Распределение документов по категориям для всей выборки (слева), для обучающей (в центре) и для контрольной (справа). Красным цветом обозначены категории первого уровня, синим – второго уровня.

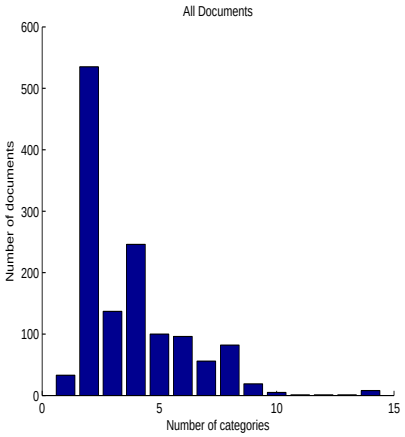


Рис. 2: Распределение количества категорий по документам.

8.3 Классический подход

Для иерархической категоризации текстов часто применяется следующий подход: для каждой пары тема–подтема строится классификатор на два класса [24], принимающий решение об отнесении документа к подтеме данной темы. Каждый классификатор обучается по выборке документов, относящихся к родительской теме, что требует больших затрат времени и памяти. В данной работе приведены результаты такого подхода с использованием в качестве классификаторов наиболее зарекомендовавшего себе в практическом

использовании классификатора SVM (support vector machines) [9]. Для обучения SVM использовалась только обучающая выборка, тестирование производилось по контрольной выборке. При этом в качестве признакового представления документов использовалось три следующих варианта. Первый — документ описывается вектором числа вхождений слов в документ. Назовем этот вариант raw. Второй — документ описывается частотами вхождений слов в документ. Обозначим этот вариант как $normalized$. Третий — документ описывается с помощью tf-idf (term frequency - inverse document frequency):

$$tf(w,d) = \frac{n_{wd}}{n_d}$$

$$idf(w,D) = \log\left(\frac{|D|}{1+|\{d \in D : w \in d\}|}\right)$$

Соответственно tf-idf $= tf(w,d) * idf(w,D)$. При этом в качестве множества D выступает только обучающая выборка. Обозначим этот вариант признакового пространства через $tf - idf$.

Каждый SVM для каждой пары тема–подтема настраивался с помощью 2-кросс-валидации по сетке параметров C, γ. Тестировались линейное и радиальное ядра. Линейное ядро $K_{lin}(u,v) = (u,v)$. Радиальное ядро $K_{rad}(u,v) = exp(-\gamma||u-v||^2)$. Будем обозначать SVM с линейным ядром как SVM_{linear}, а SVM с радиальным ядром как SVM_{radial}. Результаты категоризации контрольной выборки представлены в таблице 1 для всех категорий. В таблицах 2, 3 представлены результаты для категорий первого и второго уровня в отдельности. Категоризация документов происходила следующим образом. Сначала строилась категоризация для первого уровня иерархии категорий. Далее, для тех категорий, которые были предсказаны для документа, строилась категоризации по дочерним подкатегориям. Стоит также отметить, что idf для контрольных документов строился с использованием только обучающей выборки. Из таблицы видно, что наилучшие результаты показывают признаки на основе tf-idf, при этом особенной сильной разницы в результатах между SVM_{linear} и SVM_{radial} не наблюдается.

Модель	SVM_{linear}			SVM_{radial}		
Признаки	raw	normalized	tf-idf	raw	normalized	tf-idf
P_{micro}	0.462	0.683	0.689	0.532	0.703	0.725
R_{micro}	0.536	0.607	0.634	0.467	0.593	0.590
F_{micro}	0.496	0.643	0.661	0.497	0.643	0.650
P_{macro}	0.440	0.673	0.702	0.541	0.738	0.735
R_{macro}	0.520	0.642	0.678	0.526	0.597	0.580
F_{macro}	0.476	0.657	0.689	0.533	0.660	0.648

Таблица 1: Результаты работы иерархического категоризатора на основе SVM для категоризации контрольной выборки для всех категорий.

Модель	SVM_{linear}			SVM_{radial}		
Признаки	raw	normalized	tf-idf	raw	normalized	tf-idf
P_{micro}	0.589	0.766	0.766	0.618	0.749	0.772
R_{micro}	0.626	0.671	0.696	0.550	0.669	0.664
F_{micro}	0.607	0.715	0.730	0.582	0.707	0.714
P_{macro}	0.582	0.781	0.790	0.608	0.783	0.805
R_{macro}	0.615	0.670	0.689	0.575	0.676	0.672
F_{macro}	0.598	0.721	0.736	0.591	0.725	0.732

Таблица 2: Результаты работы иерархического категоризатора на основе SVM для категоризации контрольной выборки для категорий первого уровня.

Модель	SVM_{linear}			SVM_{radial}		
Признаки	raw	normalized	tf-idf	raw	normalized	tf-idf
P_{micro}	0.376	0.619	0.630	0.461	0.662	0.683
R_{micro}	0.466	0.556	0.585	0.400	0.533	0.531
F_{micro}	0.416	0.586	0.607	0.429	0.590	0.597
P_{macro}	0.415	0.655	0.665	0.528	0.730	0.723
R_{macro}	0.504	0.589	0.606	0.517	0.583	0.564
F_{macro}	0.455	0.620	0.634	0.522	0.648	0.633

Таблица 3: Результаты работы иерархического категоризатора на основе SVM для категоризации контрольной выборки для категорий второго уровня.

8.4 Иерархическая тематическая модель

Иерархическая тематическая модель, описанная в разделе 7 кажется более естественным подходом к задаче категоризации документов. Каждой категории поставим в соответствие тему, таким образом, чтобы иерархическая структура категорий совпадала с иерархической структурой соответствующих им тем. Далее будем отождествлять тему t и соответствующую ей категорию c. На выходе тематической модели получается иерархия тем, которые имеют четкую привязанность к категориям. Соответсвенно получается интерперетируемое в виде распределения на терминах (φ_t) представление категории.

Также для каждого документа получаем распределение на тематиках θ_d, а соответственно и на категориях. Так как θ_d является вероятностным распределением, то значение θ_{td} имеют четкую вероятностную интерпретацию принадлежности документа d тематике t, а значит и соответствующей ей категории c. Соответственно категоризацию документа можно пытаться производить с помощью отсечения категорий с маленькой вероятностью и отнесению документа к остальным категориям, имеющих большую вероятность принадлежности к документу.

Привязка иерархической тематической модели к существующей иерархии категорий осуществляется с помощью задания начальных распределений φ^0_{wt} и θ^0_{td} в алгоритме 13. Распределения φ^0_{wt} можно задавать на основе документов, относящихся к соответствующей категории. В данной работе рассматривается вариант, при котором документы конкатенируются в один большой документ, и в качестве φ^0_{wt} берется частоты слов в этом

документе. Обозначим такую инициализацию через $\varphi_{wt}^{0,D}$. Также для инициализации можно использовать слова из алфавитно-предметного указателя (АПУ) для УДК. В данной работе рассматривается способ, при котором выбирается все слова из АПУ, относящиеся к категории. Строится равномерное распределение на всех этих словах, обозначим его $\varphi_{wt}^{0,APU}$. Так как в алгоритме 13 осуществляется поуровневаяя оптимизация распределения θ_d, то θ_{sd}^0 строится как равномерное распределение на тематиках (категориях) соответствующего уровня, к которым относится документ d. Например, пусть документ d относится к трем $\{s_{i1}, s_{i2}, s_{i3}\}$ из девяти категориям на первом уровне иерархии, тогда:

$$\theta_{s_{i1}d}^0 = \frac{1}{3}, \qquad \theta_{s_{i2}d}^0 = \frac{1}{3}, \qquad \theta_{s_{i3}d}^0 = \frac{1}{3}, \qquad \theta_{sd}^0 = 0, s \neq s_{i1}, s_{i2}, s_{i3}$$

В качестве выбора порога отсечения (b) маленьких вероятностей тем рассматривались 4 варианта, при этом профиль документа θ_d предварительно сортировался по убыванию и получался профиль θ_d^{sort}. Обозначим через $\theta_d^{sort}[i]$ — i-ый элемент θ_d^{sort}. Тогда номер i для отсечения вибиралася одним из следующих способов:

1. $\sum\limits_{j=1}^{i} \theta_d^{sort}[j] > b$; Обозначим его как *type* 0;
2. $(\theta_d^{sort}[i] - \theta_d^{sort}[i+1]) > b * \theta_d^{sort}[i]$; Обозначим его как *type* 1;
3. $(\theta_d^{sort}[i] - \theta_d^{sort}[i+1]) > b$; Обозначим его как *type* 2;
4. $\theta_d^{sort}[i]$ меньше, чем кумулятивная сумма θ_d^{sort} до i-ого члена; Обозначим его как *type* 3;

Описанные выше варианта порогового правила в той или иной степени выражают критерий крутого склона для поиска вероятностей для отсечения. Значение порога b выбиралось из отрезка $[0; 1]$ по обучающей выборке. При этом также изучались влияние инициализации и значение параметров λ, μ на результат тематической модели. Начальная инициализация осуществлялась по формуле:

$$\varphi_{wt}^0 = \tau\varphi_{wt}^{0,D} + (1-\tau)\varphi_{wt}^{0,APU}$$

Для параметра τ расмматривались значения $0, 0.5, 1$, для параметра λ – значения $0, 0.33, 0.66$, для параметра μ – значения $0, 0.5, 1$. На рис. 3, 4 показаны графики изменения перплексии для контрольной выборок для $\tau = 0, 1$. Поведение перплексии при $\tau = 0.5$ имеет промежуточные результаты и не представлены в данной работе. Из графиков видно, что при $\tau = 0$ перплексия принимает значения больше, чем при $\tau = 1$. Это значит, что модель при $\tau = 0$ хуже объясняет контрольную выборку. Отсюда можно сделать вывод,

что для начальной инициализации больше подходит использование $\varphi_{wt}^{0,D}$. Также стоит отметить, что чем меньше значение параметра μ, тем ниже перплексия. Связано это с тем, что в этом случае темы менее привязаны к категориям, а значит тематическая модель может более свободно менять распределения тем. Это и объясняет меньшее значение перплексии. На графике 5 представлено зависимость категоризации обучающей выборки в F_{micro} от типа порогового правила и значения параметра b. Видно, что хуже всего работает пороговое правило *type* 2. Остальные типы дают схожие результаты. Для тестирования на контрольной выборке были выбраны следующие значения $b = 0.3$ для *type* 0, $b = 0.2$ для *type* 3, $b = 0.55$ для *type* 1. Результаты для контрольной выборки представлены на графиках 6, 7. Пороговое правило *type* 3 работает лучше, чем все стальные правила. При использовании $\mu = 0$ ухудшается качество категоризации, так как темы начинают терять связь с категориями, с которыми они были изначально связаны. Также стоит отметить, что начальная инициализация дает результат категоризации по F_{micro} равный 0.52 (не представлено на графиках). Тематическая модель позволяет увеличить качество, доводя его значение до 0.57.

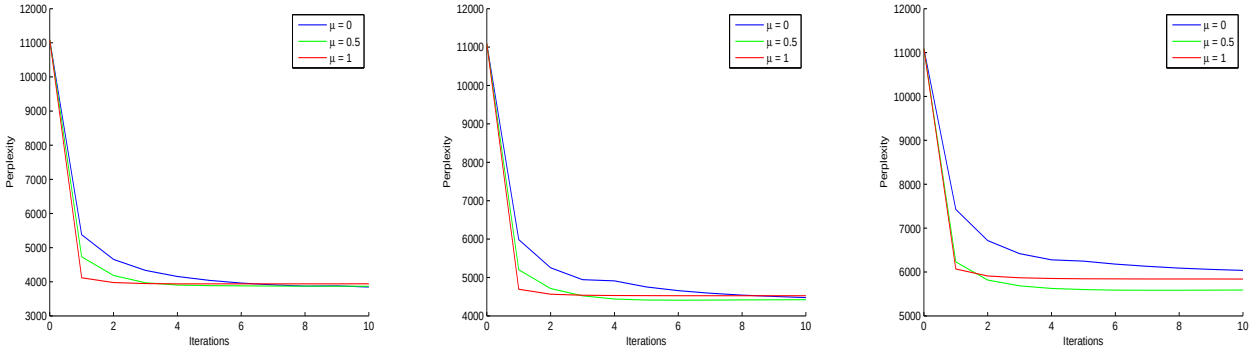


Рис. 3: Изменение перплексии контрольной выборки при $\tau = 0$ и $\lambda = 0, 0.33, 0.66$.

В результате этого эксперимента было выявлено, что лучше работает инициализация с помощью $\varphi_{wt}^{0,D}$. Однако при этом тематическая модель слабо улучшает качество категоризации по сравнению с инициализацией $\varphi_{wt}^{0,D}$. Параметр μ позволяет искать копромисс между качеством тематической модели и качеством категоризации, так как при $\mu = 0$ меньше значение перплексии, а при $\mu = 1$ выше значение качества категоризации. При этом качество категоризации оказывается все же ниже, чем при использовании SVM.

Соответственно, видится два пути улучшения иерархической тематичекой модели: улучшение перплексии и улучшение качества категоризации. Перплексию предлагается улучшать за счет введения разреживания. В работе [1] показано, что разреживание ве-

Algorithm 14 Модификация PLSA-batchHEM.

Вход: коллекция документов D; параметры λ и μ,

множество тем V и структура тематического дерева $\{S_t \colon t \in V\}$,

привязки терминов и документов к темам ϕ^0_{wt} и θ^0_{td};

минимальное количество итераций m_{min};

Выход: распределения Θ и Φ;

инициализировать φ^1_{wt} с учётом φ^0_{wt} для всех $w \in W$, $t \in V$

пока Φ не сойдутся

  m = 1;

  $\hat{n}_{wt} = 0$; $\hat{n}_t = 0$ для всех $w \in W$, $t \in V$

  для всех $d \in D$

    инициализировать θ_{td} с учётом θ^0_{td} для всех $t \in V$

    $T = \{t_0\}$; $R = \{t_0\}$; $\theta_{t_0 d} = 1$

    пока множество подтем $S := \bigcup\limits_{t \in R} S_t$ не пусто

      $\sigma_{dw} = \sum\limits_{t \in T \setminus R} \varphi^m_{wt} \theta_{td}$ для всех $w \in d$

      пока θ_{sd} не сойдутся для всех $s \in S$

        $Z_w = \sigma_{dw} + \sum\limits_{s \in S} \varphi^m_{ws} \theta_{sd}$ для всех $w \in d$

        $n_s = \sum\limits_{w \in d} n_{dw} \varphi^m_{ws} \theta_{sd} / Z_w$ для всех $w \in d$

        $n := \sum\limits_{s \in S} n_s$

        $\theta_{sd} := \mu \theta^0_{sd} + (1 - \mu) \theta_{td} n_s / n$ для всех $s \in S_t$, $t \in R$

        провести разреживание θ_d;

        нормировать θ_d;

      вычислить вес документа g_d как вероятность правильного поддерева для документа d;

      если $m < m_{min}$ **то**

        увеличить $\hat{n}_{ws}$, $\hat{n}_s$ на $n_{dw} \varphi^m_{ws} \theta_{sd} / Z_w$ для всех $w \in d$, $s \in S$

      иначе

        увеличить $\hat{n}_{ws}$, $\hat{n}_s$ на $(1 - g_d) n_{dw} \varphi^m_{ws} \theta_{sd} / Z_w$ для всех $w \in d$, $s \in S$

      $T = (T \setminus R) \cup S$; $R = S$

  $\varphi^{m+1}_{wt} := \lambda \varphi^m_{wt} + (1 - \lambda) \hat{n}_{wt} / \hat{n}_t$ для всех $w \in W$, $t \in V$

  провести разреживание φ_t для всех $t \in V$;

  нормировать φ_t для всех $t \in V$;

  $m = m + 1$

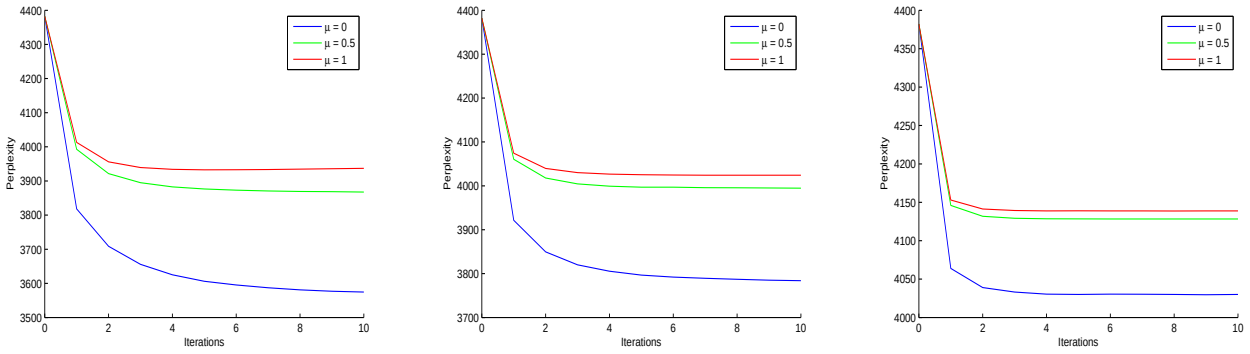


Рис. 4: Изменение перплексии контрольной выборки при $\tau = 1$ и $\lambda = 0, 0.33, 0.66$.

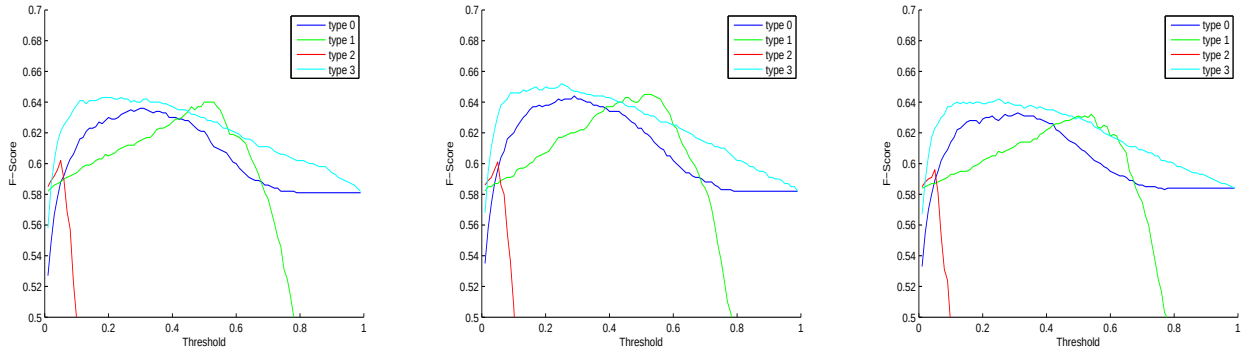


Рис. 5: Зависимость качества категоризации обучающей выборки в F_{micro} от типа порогового правила и значения параметра b при $\tau = 1, \mu = 1$ и $\lambda = 0, 0.33, 0.66$.

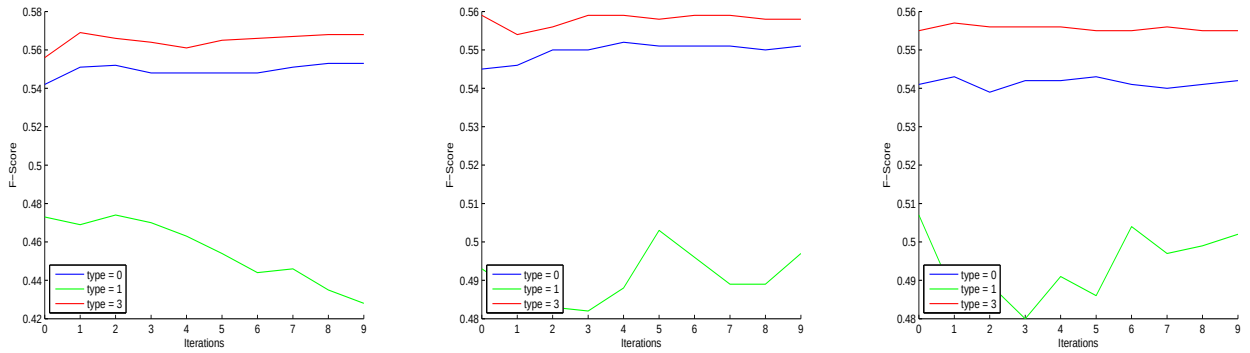


Рис. 6: Зависимость качества категоризации контрольной выборки в F_{micro} от типа порогового правила при $\tau = 1, \mu = 1$ и $\lambda = 0, 0.33, 0.66$.

роятностных распределений φ_t способно заметно уменьшать перплексию для плоских тематических моделей. Качество категоризации предлагается улучшать за счет введения весов документов. Эти две идеи представлены в алгоритме 14. В нем также предлагается

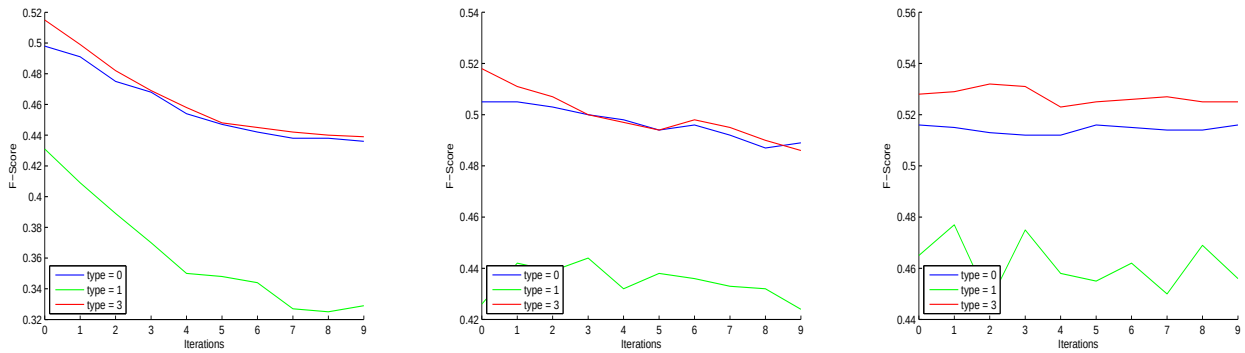


Рис. 7: Зависимость качества категоризации контрольной выборки в F_{micro} от типа порогового правила при $\tau = 1, \mu = 0$ и $\lambda = 0, 0.33, 0.66$.

отказаться от привязки распределений φ_t к начальной инициализации φ_t^0 на каждом проходе коллекции. Вместо этого используется сглаживание с распределением φ_t, полученного с предыдущего прохода по коллекции. Это должно позволить еще сильнее уменьшать перплексию. Веса документов g_d предлагается вычислять как вероятность правильного поддерева категорий документа d, то есть вычислять сумму вероятностей тем в профиле θ_d, которые соответсвуют категориям документа d. Чем выше это значение, тем лучше тематическая модель описывает документ d с точки зрения его категоризации. Хочется, чтобы тематическая модель больше настраивалась на те документы, которые она еще плохо объясняет, поэтому документы учитываются с весом $(1 - g_d)$. Разреживание θ_d осуществлялось путем обнуления вероятностей θ_{td} меньше 0.01. Разреживание ϕ_t происходило путем обнуления хвоста распределения, сумма которого меньше 0.05. Эксперименты проводились отдельно для использования g_d и для использования разреживания φ_t. В обоих случаях использовалось разреживание θ_d. На рис. 8,9 при использования g_d, на рис. 10,11 при разреживании φ_t, начиная с пятой итерации.

Из проведенных экспериментов было выявлено, что использование весов документов улучшает категоризацию контрольной выборки. Однако стоит рассмотреть и второй подход, когда плохо категоризированные документы исключаются из обучающей как шумовые. На рис. 12,13 представлены результаты для качества категоризации и значения перплексии. Фильтрация шумовых объектов производилась каждые 5 итераций и отбрасывались 10% документов с наименьшим весом g_d, что как раз и является индикатором плохой категоризации документа. Оказалось, что фильтрация документов не только не улучшает категоризацию контрольной выборки, но и ухудшает значение перплексии.

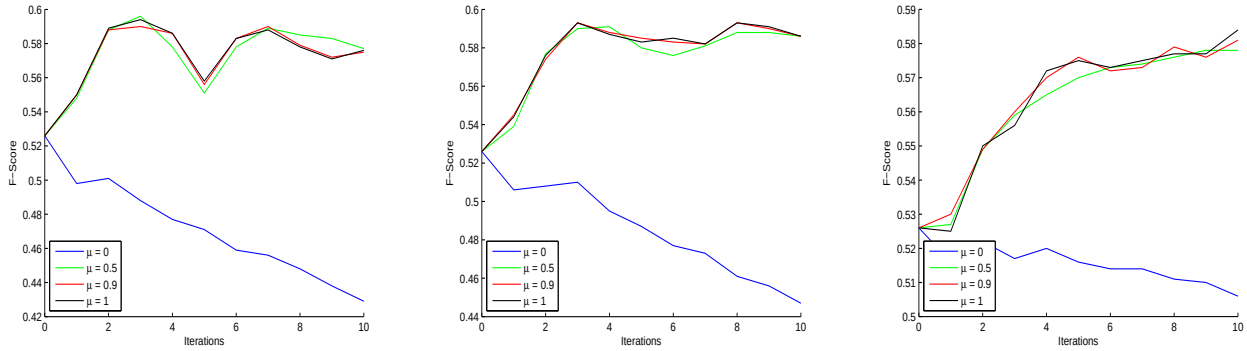


Рис. 8: Зависимость качества категоризации контрольной выборки в F_{micro} от значения μ при $\lambda = 0.2, 0.5, 0.9$ при использовании весов документов.

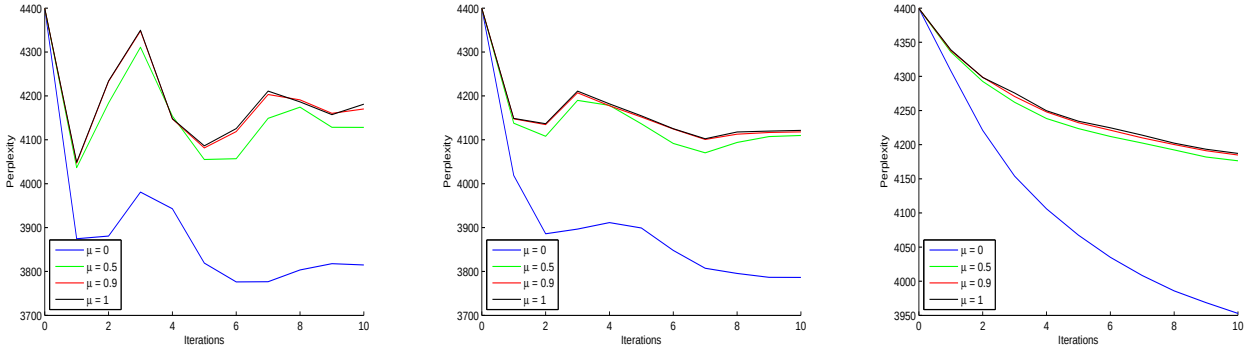


Рис. 9: Зависимость перплексии контрольной выборки от значения μ при $\lambda = 0.2, 0.5, 0.9$. при использовании весов документов.

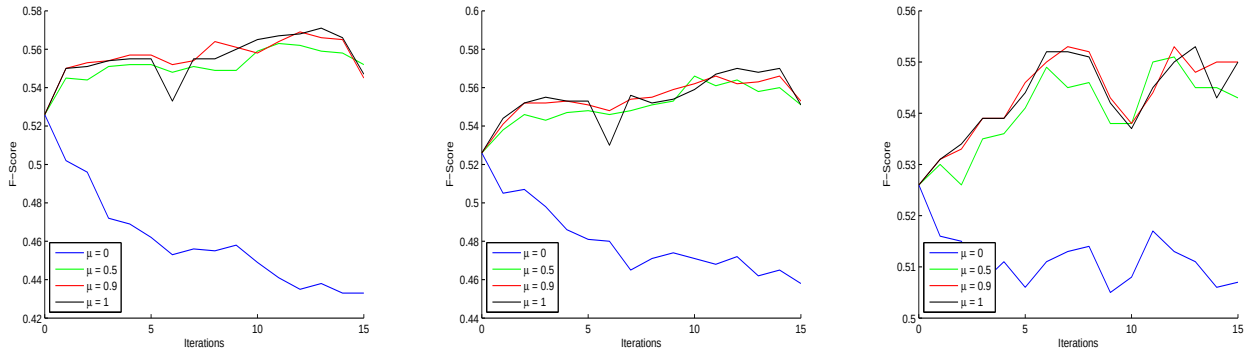


Рис. 10: Зависимость качества категоризации контрольной выборки в F_{micro} от значения μ при $\lambda = 0.2, 0.5, 0.9$ при использовании разреживания φ_t.

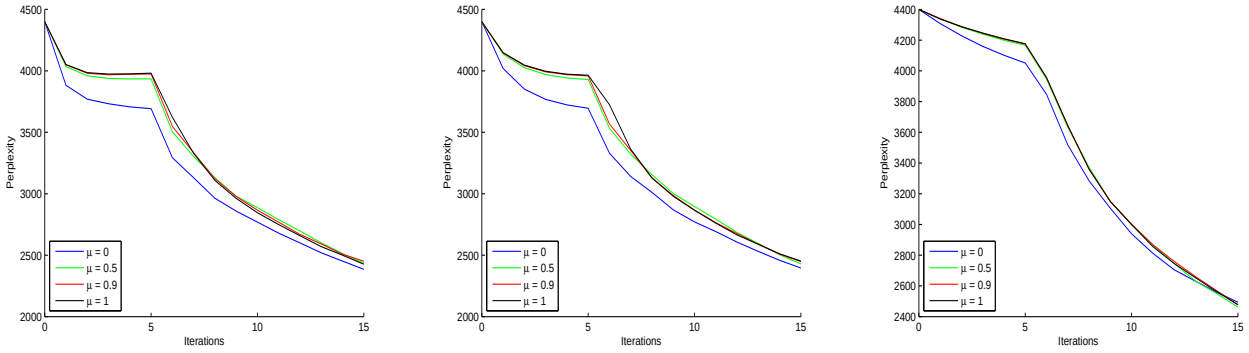


Рис. 11: Зависимость перплексии контрольной выборки от значения μ при $\lambda = 0.2, 0.5, 0.9$. при использовании разреживания φ_t.

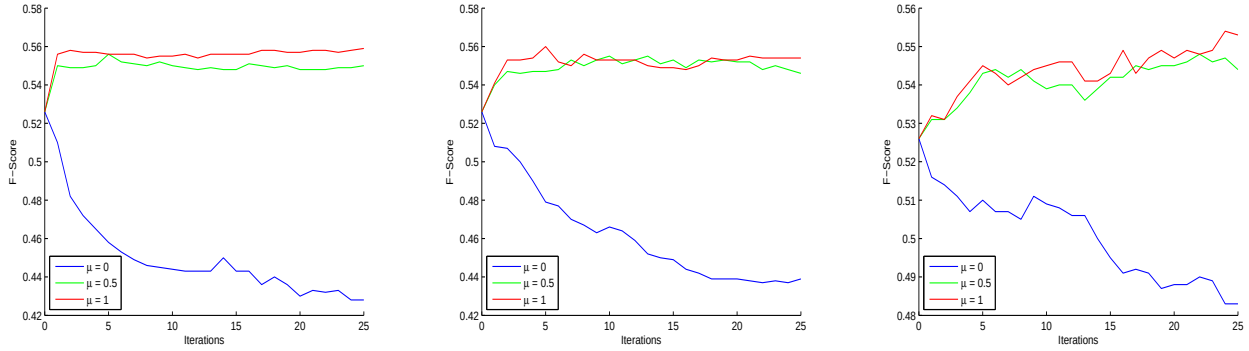


Рис. 12: Зависимость качества категоризации контрольной выборки в F_{micro} от значения μ при $\lambda = 0.2, 0.5, 0.9$ при фильтрации «шумовых» документов.

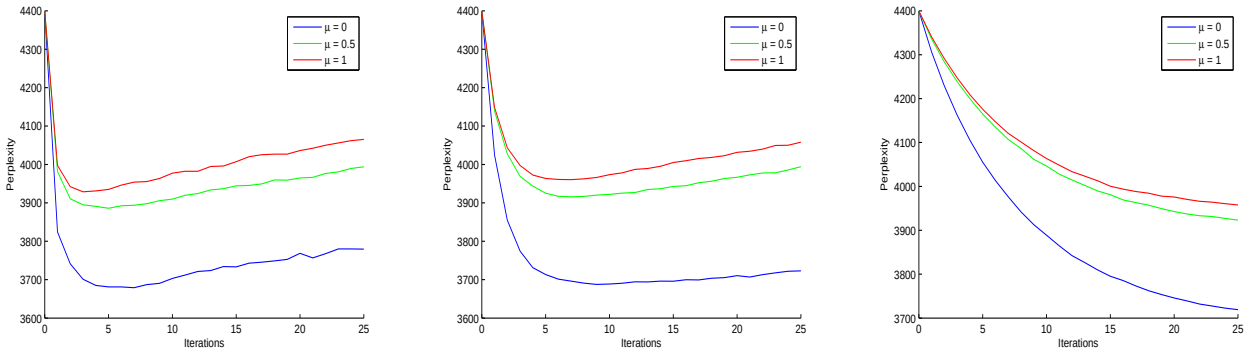


Рис. 13: Зависимость перплексии контрольной выборки от значения μ при $\lambda = 0.2, 0.5, 0.9$. при фильтрации «шумовых» документов.

8.5 Категории как смеси распределений

В описанном выше подходе категория рассматривалась как тема, то есть как распределение на терминах. В данном разделе рассматривается иной подход, при котором категория рассматривается как смесь некоторого числа тем в коллекции документов. При этом возникает вопрос о представлении категорий. При наличии размеченной коллекции документов наиболее естественным способом представления категорий видится представлении в виде конкатенации всех документов, относящихся к этой категории. Именно этот подход и используется в данном разделе. Для получения профилей категорий θ_c используется тематическая модель PLSA, описанная в разделе 2.1, и алгоритм 3. Для построения категорий использовалась только обучающая выборка. На основе полученных профилей тем φ_t также с помощью PLSA строилось признаковое описание документов обучающей и контрольной выборок. Далее по обучающей выборке настраивались соответствующие SVM для иерархической категоризации документов, как описано в разделе 8.3. Оценка качества производилась по контрольной выборке. Для того, чтобы темы сильнее отличались друг от друга дополнительно производилось разреживание φ_t каждые 10 итераций алгоритма 3 путем обнуления хвоста распределения, сумма которого меньше 0.01. При этом обнулялось не более 15% ненулевых элементов распределения φ_t, чтобы сильно не ухудашть тематические профили. Всего производилось 100 итераций алгоритма. На рисунке 14 представлена распределение количества тем по категориям при количестве там равным 68 при использовании разреживания профиля категорий. Обнулялись элементы профиля категории, значение которых меньше 0.01. На рисунке 15 представлено зависимость качества категоризации в F_{micro} от количества темы для линенйного и радиального ядра SVM. Оказалось, что результаты категоризации лучше для радиального ядра и достигают практически тех же результатов, что и с использованием tf-idf. Тем самым удается сократить размерность пространства признаков для представления документов с 52927 до 100 признаков без потери качества. При этом стоит также отметить, что разреживание улучшает качество категоризации для обоих типов ядер.

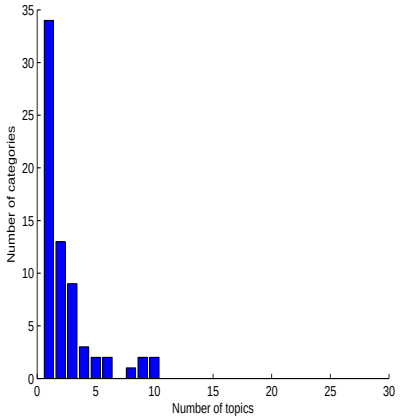


Рис. 14: Распределение количества тем по категориям при разреживании профиля категорий.

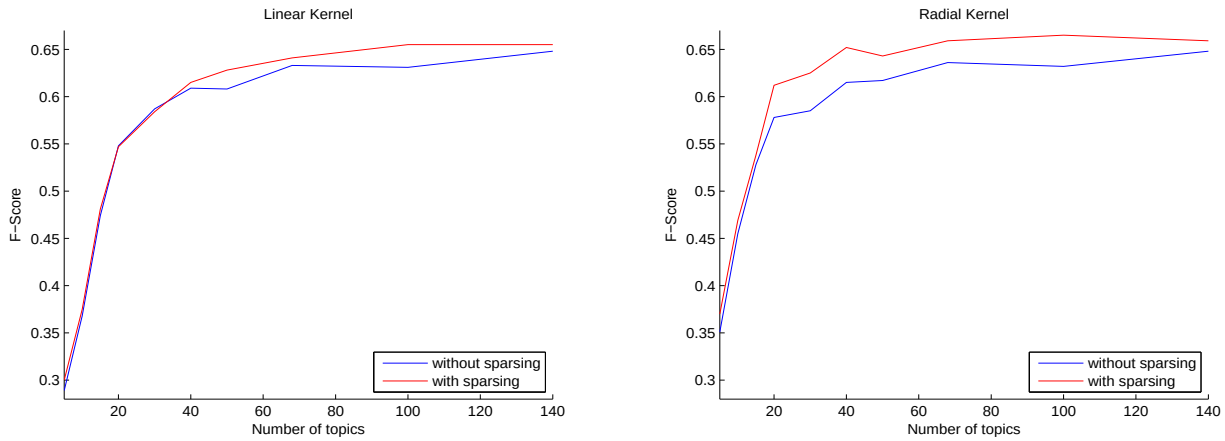


Рис. 15: Зависимость качества категоризации контрольной выборки в F_{micro} от количества «супертем». Слева для линейного ядра, справа для радиального ядра.

9 Заключение

Результатами данной работы являются:

- Разработаны алгоритмы иерархического тематического моделирования для категоризации текстов.
- Показано, что частичное обучение улучшает качество категоризации, но ухудшает перплексию.
- Показано, что разреживание позволяет обнулить до 90% вероятностей $p(w|t)$, не ухудшая качество категоризации.
- Показано, что описание категорий разреженными смесями тем позволяет достичь наилучшего качества категоризации.

10 Список литературы

[1] Воронцов К. В., Потапенко А. А. Регуляризация, робастность и разреженность вероятностных тематических моделей // Компьютерные исследования и моде- лирование. Т. 4, № 4. С. 693–706. — 2012.

[2] AlSumait L., BarbaraD̈, GentleJ̇, Domeniconi C. Topic significance ranking of LDA generative models. // ECML. — 2009.

[3] Blackwell D., MacQueen J. Ferguson Distributions via Polya Urn Schemes // Annals of Statistics, 1. — 1973. — Pp. 353–355.

[4] Blei D. M., Ng A. Y., Jordan M. I. Latent Dirichlet allocation // Journal of Machine Research. — 2003. —Vol. 3. — Pp.993-1022.

[5] Blei D. M., Grifiths T. L., Jordan M. I., Tenenbaum J. Hierarchical topic models and the nested Chinese restaurant process. // Neural Information Processing Systems 16. — 2003.

[6] Blei D. M., Grifiths T. L., Jordan M. I. The Nested Chinese Restaurant Process and Bayesian Nonparametric Inference of Topic Hierarchies. — 2010.

[7] Blei D. M., Paisley J., Wqang C., Jordan M. I. Nested Hierarchical Dirichlet Processes. — 2012.

[8] Chang J., Boyd-Graber J., Gerris S., Wang C., Blei D. Reading tea leaves: How humans interpret topic models. // In Proc. of NIPS. — 2009.

[9] Cortes C., Vapnik V. Support-Vector Networks. // Machine Learning, 20. — 1995.

[10] Dempster A. P., Laird N. M., Rubin D. B. Maximum likelihood from incomplete data via the EM algorithm // J. of the Royal Statistical Society, Series B. —1977. — no. 34, Pp. 1–38.

[11] Ferguson T. Bayesian Analysis of Some Nonparametric Problems // Annals of Statistics, 1(2). — 1973. — Pp. 209–230.

[12] Heinrich G. “Infinite LDA” – Implementing the HDP with minimum code complexity. Technical note TN2011/1. — 2011.

[13] Hofmann T. Probabilistic latent semantic indexing // Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. — New York, NY, USA: ACM, 1999. — Pp.50-57.

[14] Hoffman M., Blei D., Bach F. Online learning for latent Dirichlet allocation. // Neural Information Processing Systems. — 2010.

[15] Ikonomakis M., Kotsiantis S., Tampakas V. Text Classification Using Machine Learning Techniques. — 2005.

[16] Joachims T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. // Proceedings of the European Conference on Machine Learning (ECML), Springer. — 1998.

[17] Mao Xian-Ling et al. SSHLDA: A Semi-Supervised Hierarchical Topic Model. // EMNLP-CoNLL, pp 800-809. — 2012.

[18] Mimno D., Wallach H., Talley E., Leenders M., McCallum A. Optimizing Semantic Coherence in Topic Models. // EMNLP. — 2011.

[19] Mimno D., Blei D. Bayesian Checking for Topic Models. // EMNLP. — 2011.

[20] Newman D., Lau J., Grieser K., Baldwin T. Automatic evaluation of topic coherence. // NAACL HLT. — 2010.

[21] Newman D., Noh Y., Talley E., Karimi S., Baldwin T. Evaluating topic models for digital libraries. // Proc. of JCDL/ICADL. — 2010.

[22] Newman, Lau, Greiser, Baldwin. Automatic Labelling of Topic Models. //ACL-HLT. — 2011.

[23] Rubin T., Holloway A., Smyth P., Steyvers M. Statistical Topic Models for Multi-Label Document Classification. // Machine Learning (Journal). — 2012.

[24] Sebastiani F . Machine learning in automated text categorization // ACM Computing Surveys. — 2002. — Vol. 34, no. 1. Pp. 1–47.

[25] Sethurman J. A Constructive Definition of Dirichlet Priors // Statistica Sinica, 4. — 1994. — Pp. 639–650.

[26] Si J., Li Q., Qian T., Deng X. Hierarchical Clustering on HDP Topics to build a Semantic Tree from Text. — 2012.

[27] Sriurai W. Improving text categorization by using a topic model. // Advanced Computing: An International Journal, Vol.2, No.6. — 2011.

[28] Steyvers M., Griffiths T., Finding scientific topics // Proceedings of the National Academy of Sciences. — 2004. — Vol. 101, no. Suppl. 1. Pp. 5228–5235.

[29] Steyvers M. Griffiths T. Probabilistic topic models. In Landauer T., McNamara D., Dennis S., Kintsch W. Latent Semantic Analysis: A Road to Meaning, pp 427-448. — 2007.

[30] Tasci S., Gungor T. LDA-based keyword selection in text categorization. // ISCIS. — 2009.

[31] Teh Y., Newman N., Welling .M. A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation. // NIPS. — 2006.

[32] Teh Y., Jordan M., Beal M., Blei D. M. Hierarchical Dirichlet processes // Journal of the American Statistical Association, 101(476). — 2007. — Pp. 1566–1581.

[33] Teh Y., Kurihara K., Welling M. Collapsed Variational Inference for HDP. // NIPS. — 2007.

[34] Teh Y. Encyclopedia of Machine Learning, chap. Dirichlet Processes. Springer, — 2010.

[35] de Waal A., Barnard E. Evaluating topic models with stability. — 2008.

[36] Wallach H. M., Murray I., Salakhutdinov R., et al. Evaluation methods for topic models. // ICML. — 2009.

[37] Wang Y. Distributed Gibbs sampling of latent dirichlet allocation: The gritty details. — 2008.

[38] Zavitsanos E., Paliouras G., Vouros G. Non-Parametric Estimation of Topic Hierarchies from Texts with Hierarchical Dirichlet. — 2011.

[39] Zavitsanos E., Paliouras G., Vouros G. Gold Standard Evaluation of Ontology Learning Methods through Ontology Transformation and Alignment. — 2011.