Тематическое моделирование текстов на естественном языке

Антон Коршунов, Андрей Гомзин
{korshunov, gomzin}@ispras.ru

Аннотация. Тематическое моделирование — способ построения модели коллекции текстовых документов, которая определяет, к каким темам относится каждый из документов. Переход из пространства терминов в пространство найденных тематик помогает разрешать синонимию и полисемию терминов, а также эффективнее решать такие задачи, как тематический поиск, классификация, суммаризация и аннотация коллекций документов и новостных потоков. Наибольшее применение в современных приложениях находят подходы, основанные на Байесовских сетях — ориентированных графических вероятностных моделях, позволяющих учитывать авторство документов, связи между словами, темами, документами и авторами, а также другие типы сущностей и метаданных. В статье приведён сравнительный обзор различных моделей, описаны способы оценивания их параметров и качества результатов, а также приведены примеры открытых программных реализаций.

Ключевые слова: тематическое моделирование; тематический поиск; классификация документов; графические вероятностные модели; Байесовские сети; скрытое размещение Дирихле; уменьшение размерности; анализ текста; извлечение информации; машинное обучение.

1. Введение

В 1958 году Герхард Лисовски и Леонард Рост завершили работу по составлению каталога религиозных текстов на иврите, призванных помочь учёным определить значения терминов, которые были давно утрачены [20]. Путём кропотливой ручной работы они собрали воедино все возможные контексты, в которых появлялся каждый из терминов. Следующей задачей было научиться игнорировать несущественные различия в формах слов и выделять те различия, которые влияют на семантику. Замыслом авторов было дать возможность исследователям языка проанализировать различные отрывки и понять семантику каждого термина в его контексте.

Трудности, с которыми столкнулись Лисовски и Рост полвека назад, часто возникают и сегодня при автоматическом анализе текстов. Одна и та же концепция может выражаться любым количеством различных терминов (*синонимия*), тогда как один термин часто имеет разные смыслы в различных контекстах (*полисемия*). Таким образом, необходимы способы различать

варианты представления одной концепции и определять конкретный смысл многозначных терминов. Кроме того, нужно уметь представлять данные в доступной для человека форме, чтобы дать возможность понять неизвестный ему смысл термина. Теоретически обоснованным и активно развивающимся направлением в анализе текстов на естественном языке, призванным решать перечисленные задачи, является тематическое моделирование коллекций текстовых документов.

Построение тематической модели может рассматриваться как задача одновременной кластеризации документов и слов по одному и тому же множеству кластеров, называемых темами. В терминах кластерного анализа *тема* — это результат би-кластеризации, то есть одновременной кластеризации и слов, и документов по их семантической близости. Обычно выполняется *нечёткая кластеризация*, то есть документ может принадлежать нескольким темам в различной степени. Таким образом, сжатое семантическое описание слова или документа представляет собой вероятностное распределение на множестве тем. Процесс нахождения этих распределений и называется *тематическим моделированием* [27].

Как правило, количество тем, встречающихся в документах, меньше количества различных слов во всем наборе. Поэтому скрытые переменные — темы — позволяют представить документ в виде вектора в пространстве скрытых (*латентных*) тем вместо представления в пространстве слов. В результате документ имеет меньшее число компонент, что позволяет быстрее и эффективнее его обрабатывать. Таким образом, тематическое моделирование тесно связано с другим классом задач, известным как *уменьшение размерности* данных [14]. Кроме того, найденные темы могут использоваться для семантического анализа текстов.

Задача извлечения скрытых тем из коллекции текстовых документов имеет множество применений. Помимо кластеризации и классификации документов, найденные темы могут применяться для определения релевантности документа заданной теме или запросу, определения тематического сходства документа с другими документами и их фрагментами, построения тематических профилей авторов, разбиения документа на тематически однородные фрагменты и т.д.

В силу своей универсальности и расширяемости, современные способы тематического моделирования находят применение в широком спектре приложений [5, 21, 22, 27]:

- кластеризация, классификация, ранжирование, аннотирование и суммаризация отчётов, научных публикаций, переписки, блогов, студенческих работ и т.д.;
- тематический поиск документов и связанных с ними объектов: рисунков, авторов, организаций, журналов, конференций;

- фильтрация спама;
- рубрикация коллекций изображений, видео, музыки;
- поиск генетических паттернов в различных популяциях и определение пропорции этих паттернов у конкретного индивидуума;
- коллаборативная фильтрация в сервисах рекомендаций;
- построение тематических профилей пользователей форумов, блогов и социальных сетей для поиска тематических сообществ и определения наиболее активных их участников;
- анализ новостных потоков и сообщений из социальных сетей для определения актуальных событий реального мира и реакции пользователей на них.

Иными словами, тематическое моделирование позволяет автоматически систематизировать и реферировать электронные архивы такого масштаба, который человек не в силах обработать.

Дальнейшее изложение строится следующим образом. В разделе 2 вводится понятие векторного представления документов и описываются ранние подходы к поиску тем, основанные на кластеризации. Раздел 3 содержит описание метода латентно-семантического индексирования (LSI), который рассматривает исходный набор данных как матрицу «документ-термин» и использует матричные разложения для извлечения скрытых тем. В разделе 4 рассмотрено применение Байесовских сетей для тематического моделирования текстов на примерах вероятностного латентно-семантического индексирования (PLSI) и скрытого размещения Дирихле (LDA). Здесь же коротко описаны способы оценивания оптимальных значений параметров моделей для обучающего и тестового набора документов. Примеры более сложных вероятностных моделей, позволяющих устранить ограничения и недостатки первых подходов, приведены в разделе 5. В разделе 6 описаны основные способы оценки качества результатов тематического моделирования. Наконец, раздел 7 содержит примеры программных реализаций алгоритмов тематического моделирования.

2. Кластеризация и классификация документов

Задача *определения и отслеживания тем (Topic Detection and Tracking, TDT)* возникла в 1996-1997 годах. В работе [1] понятие темы тесно связано с понятием события: *тема* — это событие или действие вместе со всеми непосредственно связанными событиями и действиями. Задача заключается в извлечении событий из потока информации.

Для представления документов принято пользоваться *векторной моделью (Vector Space Model, VSM)*, в которой каждому слову сопоставляется вес в

соответствии с выбранной весовой функцией. Располагая таким представлением для всех документов, можно, например, находить расстояние между точками пространства и тем самым решать задачу подобия документов — чем ближе расположены точки, тем больше похожи соответствующие документы.

Классическим методом назначения весов словам является *TF-IDF*:

$$TFIDF(t,d,D) = TF(t,d) \times IDF(t,D) \quad (1)$$

TF *(term frequency)* — нормализованная частота слова в тексте:

$$TF(t,d) = \frac{freq(t,d)}{\max_{w \in D} freq(w,d)} \quad (2)$$

Здесь $freq(t,D)$ — число вхождений слова *t* в документе *d*.

IDF *(inverse document frequency)* — обратная частота документов:

$$IDF(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \quad (3)$$

Здесь в числителе — количество документов в наборе, а в знаменателе — количество документов, в которых встречается слово *t*.

В зависимости от решаемой задачи используются различные модификации TF-IDF. Например, в одном из решений, описанных в [1] используются следующие веса:

$$w(t,D) = (1 + \log_2 TF(t,D)) \times \frac{IDF(t)}{\left\|\vec{d}\right\|} \quad (4)$$

Здесь $\left\|\vec{d}\right\|$ - норма вектора, представляющего документ *D*.

В более поздних исследованиях [4] использовались следующие модификации TF-IDF:

$$TF' = \frac{TF}{TF + 0.5 + 1.5\frac{len_d}{len_{avg}}} \quad (5)$$

Здесь len_d — длина документа *d*, len_{avg} — средняя длина документа.

$$IDF' = \frac{\log(IDF)}{\log(N+1)} \quad (6)$$

Для сравнения векторов документов в [1] и [4] применялись такие метрики, как косинус, дивергенция Кульбака-Лейблера и другие методы (взвешенная сумма компонентов документа, простые языковые модели). Всего существует более 70 способов расчёта похожести векторов [29].

В [1] рассматривается два типа задач: обнаружение событий из набора данных за определенный период времени и обнаружение событий в режиме реального времени.

Первый тип задач заключается в разбиении исходных данных на группы, соответствующие событиям, а также в определении, описывает ли текстовый документ из набора какое-либо событие. Основной идеей всех решений было использование алгоритмов кластеризации [2, 30] (*инкрементальная кластеризация, метод K-средних* и др). При этом предполагается, что каждый кластер содержит документы, описывающие какое-либо событие.

Задача второго типа — для нового документа определить, описывает ли он событие, которое уже встречалось в исходных данных. Для отслеживания событий использовались алгоритмы классификации [3] (*метод k-ближайших соседей*, *решающие деревья* и др). Классификация производилась с использованием двух классов: *YES* — документ описывает событие, *NO* — не описывает.

Таким образом, в ранних исследованиях тема отождествлялась с событием. В реальной жизни тема может описывать иные сущности, а не только события. Поэтому в более поздних работах задачи определения событий и тем стали различаться. Еще один недостаток описанных методов в том, что анализируемые документы относятся только к одной теме или событию. Однако один документ может затрагивать несколько тем. К тому же, векторное представление документов не позволяет разрешать синонимию и полисемию терминов (см. раздел 1).

Для решения перечисленных проблем было предложено рассматривать набор векторов терминов из документов как общую терм-документную матрицу и применять к ней особые разложения (метод LSI).

3. Латентно-семантическое индексирование

80-е годы прошлого столетия ознаменовались активным развитием систем информационного поиска по коллекциям документов разнообразной природы. Следуя принципу «от простого к сложному», первыми были реализованы подходы, основанные на поиске точных совпадений частей документов с запросами пользователей. Довольно скоро, однако, стало очевидно различие между *релевантностью* (соответствием) документа запросу и точным

совпадением их частей. Зачастую документы, релевантные запросу с точки зрения пользователя, не содержали терминов из запроса и поэтому не отображались в результатах поиска (проблема синонимии). С другой стороны, большое количество документов, слабо или вовсе не соответствующих запросу, показывались пользователю только потому, что содержали термины из запроса (проблема полисемии).

Самым простым решением этих проблем кажется добавление к запросу *уточняющих* терминов для более точного описания интересующего контекста. Однако предположение о том, что индекс поисковой системы содержит все возможные уточняющие термины, на практике выполняется довольно редко.

В 1988 г. Dumais et al [36] предложили метод *латентно-семантического индексирования (latent semantic indexing, LSI)*, призванный повысить эффективность работы информационно-поисковых систем путём проецирования документов и терминов в пространство более низкой размерности, которое содержит *семантические концепции* исходного набора документов.

Основная идея метода состоит в оценивании корреляции терминов путём анализа их совместной встречаемости в документах. К примеру, в коллекции всего 100 документов, содержащих термины «*доступ*» и/или «*поиск*». При этом только 95 из них содержат оба термина вместе. Логично предположить, что отсутствие термина «*поиск*» в документе с термином «*доступ*» ошибочно и возвращать данный документ по запросу, содержащему только термин «*доступ*». Разумеется, подобные выводы можно делать не только из простой попарной корреляции терминов.

С другой стороны, анализируя корреляцию терминов в запросе, можно более точно определять интересующий пользователя смысл основного термина и повышать позиции документов, соответствующих этому смыслу, в результатах поиска.

Таким образом, при латентно-семантическом индексировании документов задача состоит в том, чтобы спроецировать часто встречающиеся вместе термины в одно и то же измерение семантического пространства, которое имеет пониженную размерность по сравнению с оригинальной *терм-документной матрицей*, которая обычно довольно разрежена. Элементы этой матрицы содержат веса терминов в документах, назначенные с помощью выбранной весовой функции (см. раздел 2). В качестве примера можно рассмотреть самый простой вариант такой матрицы, в которой вес термина равен 1, если он встретился в документе (независимо от количества появлений), и 0 если не встретился (рис. 1).

	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
voyage	1	0	0	1	1	0
trip	0	0	0	1	0	1

Рис. 1. Терм-документная матрица.

Наиболее распространенный вариант LSI основан на использовании разложения терм-документной матрицы по сингулярным значениям — так называемом *сингулярном разложении (Singular Value Decomposition, SVD)*, которое хорошо зарекомендовало себя в факторном анализе.

Согласно *теореме о сингулярном разложении*, любая вещественная прямоугольная матрица может быть разложена на произведение трех матриц:

$$A = TSD^T, \quad (7)$$

где матрицы T и D — ортогональные, а S — диагональная матрица, элементы на диагонали которой называются *сингулярными значениями* матрицы A.

Такое разложение обладает замечательной особенностью: если в матрице S оставить только k наибольших сингулярных значений, а в матрицах T и D — только соответствующие этим значениям столбцы, то произведение получившихся матриц S, T и D будет наилучшим приближением исходной матрицы A к матрице $\hat{A}$ ранга k.

Если в качестве матрицы A взять терм-документную матрицу, то матрица $\hat{A}$, содержащая только k первых линейно независимых компонент A, отражает основную структуру различных зависимостей, присутствующих в исходной матрице. Структура зависимостей определяется весовыми функциями терминов.

Таким образом, каждый термин и документ представляются при помощи векторов в общем семантическом пространстве размерности k. Близость между любой комбинацией терминов и/или документов легко вычисляется при помощи скалярного произведения векторов. Для задач информационного поиска запрос пользователя рассматривается как набор терминов, который

проецируется в семантическое пространство, после чего полученное представление сравнивается с представлениями документов в коллекции.

Как правило, выбор k зависит от поставленной задачи и подбирается эмпирически. Если выбранное значение k слишком велико, то метод теряет свою мощность и приближается по характеристикам к стандартным векторным методам. Слишком маленькое значение k не позволяет улавливать различия между похожими терминами или документами.

В качестве альтернативы SVD в тематическом моделировании [37] также применяется *неотрицательная матричная факторизация (non-negative matrix factorization, NMF)*. Данный способ разложения матриц накладывает ограничение на результирующие матрицы (факторы): все их элементы должны быть положительными либо нулевыми [38].

Следующим этапом развития тематического моделирования стали подходы, позволяющие моделировать *вероятности* скрытых тем в документах и терминов в темах (см. раздел 4). В результатах работы LSI эти вероятности для каждой темы и документа распределены равномерно, что не соответствует характеристикам реальных коллекций документов. В отличие от так называемых *дискриминативных* подходов (к которым относится LSI), в вероятностных подходах сначала задаётся модель, а затем с помощью терм-документной матрицы оцениваются её скрытые параметры, которые затем могут быть использованы для генерации моделируемых распределений. Из этого следуют преимущества вероятностного моделирования документов:

- результаты работы представляются в терминах теории вероятностей и поэтому могут быть с минимальными затратами встроены в другие вероятностные модели и проанализированы стандартными статистическими методами;
- новые порции входных данных не требуют повторного обучения модели;
- использование Байесовского непараметрического моделирования позволяет избежать подбора входных параметров, что делает такие модели более гибкими;
- вероятностные модели могут быть с лёгкостью расширены путём добавления переменных, а также новых связей между наблюдаемыми и скрытыми переменными.

4. Вероятностные тематические модели

Вероятностное тематическое моделирование — это набор алгоритмов, позволяющих анализировать слова в больших наборах документов и извлекать из них темы, связи между темами и изменение их во времени [5]. При этом

документ рассматривается как набор слов, порядок которых не имеет значения. Для каждого документа определено распределение θ_d его слов по темам, т.е. вероятность для каждой темы встретить ее в данном документе, причём . В документе на рис. 2 это распределение изображено справа в виде гистограммы.

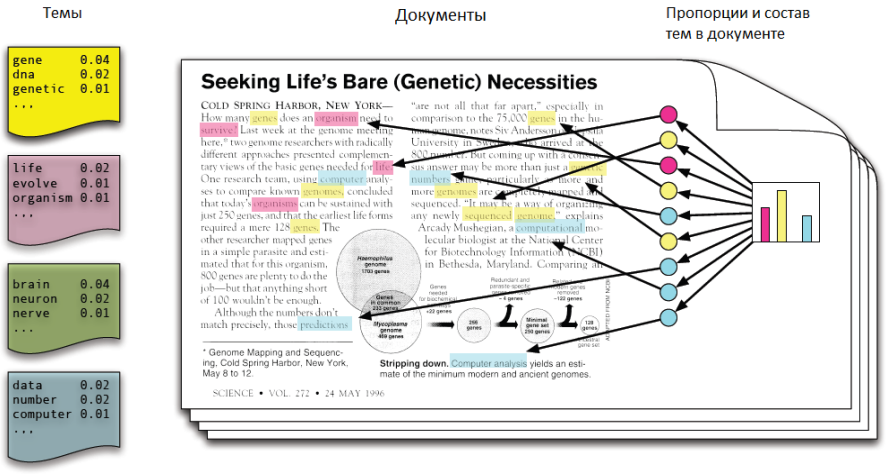

Рис. 2. Интуитивное представление тем и тематических моделей.

Тема представляется в виде распределения φ_t слов из фиксированного словаря, т.е. каждое слово входит в тему с некоторой вероятностью , причём . На рис. 2 распределения φ_t изображены слева.

Вероятностные модели являются *генеративными (порождающими)*, то есть их можно использовать для генерации документов. Описание модели, как правило, начинается со способа генерации документов — *генеративного процесса*. Однако основной целью тематического моделирования является не генерация, а извлечение тем из имеющегося набора документов. То есть нужно определить скрытые структуры (темы, распределение тем в каждом документе и т.д.) при данных словах в документах. Другими словами, это задача, обратная генерации: выяснить, с помощью каких скрытых структур вероятнее всего могли бы быть сгенерированы исходные документы.

Задача *оценивания модели* заключается в том, чтобы найти значения параметров модели, при которых наблюдаемая обучающая выборка максимально правдоподобна. Задача *вывода по модели* состоит в определении значений скрытых переменных (например, скрытых вероятностей тем) для нового документа, изначально не входившего в состав обучающей выборки. Поскольку эти задачи отличаются лишь исходными данными и поэтому часто решаются похожими способами, то далее будет использоваться общий термин *оценивание параметров модели* для обеих задач.

Вводные обзоры по вероятностному тематическому моделированию даны в [5, 26, 27]. В них рассмотрены многочисленные модификации ранних моделей, используемые для современных приложений информационного поиска. Текущий и последующие разделы призваны дополнить предыдущие обзоры описанием недавно предложенных подходов к построению тематических моделей, а также оцениванию их параметров и качества результатов.

4.1. Графические модели. Плоское графическое представление

Вероятностные тематические модели по своей природе являются графическими и поэтому часто представляются в виде интуитивно понятного и наглядного *плоского графического представления* [28]. Каждая случайная величина обозначается кругом. Наблюдаемые величины (значения которых известны) закрашиваются, скрытые (значения которых надо найти) остаются незакрашенными. Стрелка (направленное ребро) из первой вершины во вторую обозначает условную зависимость второй величины от первой.

Зависимость величины Y от величины X означает, что совместную вероятность $P(X,Y)$ можно представить в виде $P(Y|X)P(X)$. Если X и Y независимы, то $P(X,Y) = P(X)P(Y)$.

Прямоугольник, включающий в себя некоторый подграф с указанным в правом нижнем углу числом N, обозначает совокупность N экземпляров данного подграфа. Прямоугольники могут быть вложенными.

Пример модели представлен на рис. 3. Здесь изображены величины и зависимости. Величины c и w — наблюдаемые, остальные — скрытые. В модели присутствует C величин α, D величин φ, MD величин c и θ, ND величин w, y, z. Стрелками показаны зависимости: например, z зависит от соответствующей y и от всех θ.

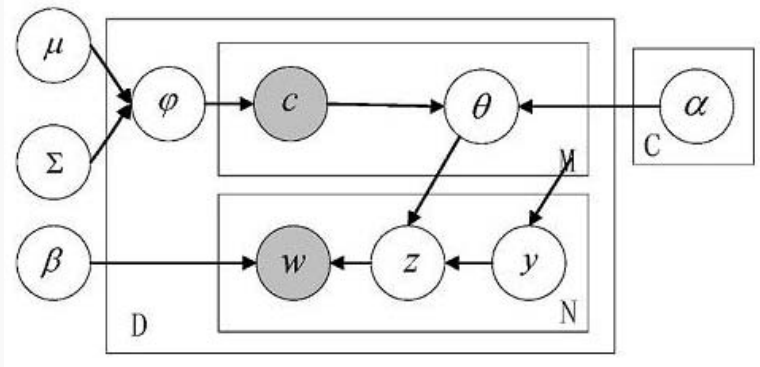

Рис. 3. Пример графической вероятностной модели.

4.2. Вероятностное латентно-семантическое индексирование

Одной из первых вероятностных тематических моделей является *вероятностное латентно-семантическое индексирование (Probabilistic Latent Semantic Indexing, PLSI)*, предложенное Томасом Хоффманом в 1999 году [6, 7].

В основе PLSI лежит т.н. *аспектная модель*, которая связывает скрытые переменные тем $z \in Z = \{z_1, \dots, z_k\}$ с каждой наблюдаемой переменной — словом или документом. Таким образом, каждый документ может относиться к нескольким темам с некоторой вероятностью, что является отличительной особенностью этой модели по сравнению с подходами, не позволяющими вероятностного моделирования.

Генеративный процесс следующий:

1. Выбрать документ d согласно распределению $p(d)$
2. Выбрать тему $i \in \{1, \dots, k\}$ на основе распределения $\theta_{di} = p(z = i|d)$
3. Выбрать слово v — значение переменной w на основе распределения $\varphi_{iv} = p(w = v|z = i)$

Совместная вероятностная модель над документами и словами определена следующим образом:

$$P(d,w) = P(d)\sum_{z \in Z} P(w \mid z)P(z \mid d) \tag{8}$$

Также модель может быть представлена в виде:

$$P(d,w) = \sum_{z \in Z} P(z)P(w \mid z)P(d \mid z) \tag{9}$$

Представление (8) является асимметричным, представление (9) — симметричным (рис. 4).

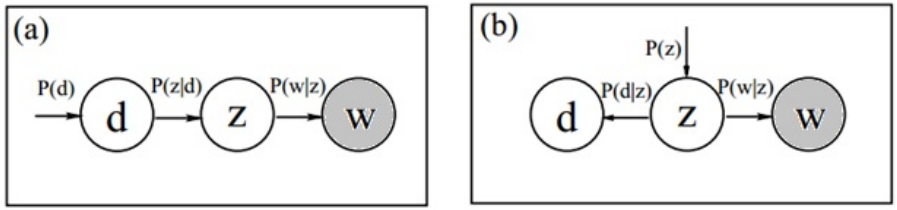

Рис. 4. Графическое представление модели вероятностного латентно-семантического индексирования с асимметричной (a) и симметричной (b) параметризацией.

4.2.1 Оценивание параметров модели вероятностного латентно-семантического индексирования

Для определения оптимальных значений скрытых параметров модели используется стандартная процедура оценки максимального правдоподобия — *EM-алгоритм (Expectation Maximization)* [8]. Он применяется к симметричному представлению модели.

На E-шаге алгоритма оценивается вероятность *P(z|d,w)*:

$$P(z \mid d,w) = \frac{P(z)[P(d \mid z)P(w \mid z)]^{\beta}}{\sum_{z' \in Z} P(z')[P(d \mid z')P(w \mid z')]^{\beta}} \tag{10}$$

где $\beta<1$ — задаваемый параметр [6, 7].

На M-шаге алгоритма вычисляются:

$$P(w \mid z) = \frac{\sum_d n(d,w)P(z \mid d,w)}{\sum_{d,w'} n(d,w')P(z \mid d,w')} \tag{11}$$

$$P(d \mid z) = \frac{\sum_w n(d,w)P(z \mid d,w)}{\sum_{d',w} n(d',w)P(z \mid d',w)} \tag{12}$$

$$P(z) = \frac{\sum_{d,w} n(d,w)P(z \mid d,w)}{\sum_{d,w} n(d,w)} \tag{13}$$

Несмотря на очевидные преимущества перед более ранними подходами, модель PLSI не лишена недостатков. Во-первых, она содержит большое число параметров, которое растет в линейной зависимости от числа документов. Как следствие, модель склонна к переобучению и неприменима к большим наборам данных. Во-вторых, невозможно вычислить вероятность документа, которого нет в наборе данных. В-третьих, отсутствует какая-либо закономерность при генерации документов из сочетания полученных тем. Данные недостатки устранены в модели LDA.

4.3. Скрытое размещение Дирихле

Скрытое размещение Дирихле (Latent Dirichlet Allocation, LDA) — генеративная графическая вероятностная модель, предложенная Дэвидом Блеем и соавторами в 2003 году [9]. Процесс генерации документа похож на генеративный процесс в PLSI. Каждый документ генерируется независимо:

1. Случайно выбрать для документа его распределение по темам θ_d

2. Для каждого слова в документе:
    a. Случайно выбрать тему из распределения θ_d, полученного на 1-м шаге
    b. Случайно выбрать слово из распределения слов в выбранной теме φ_t

Схема модели скрытого размещения Дирихле изображена на рис. 5.

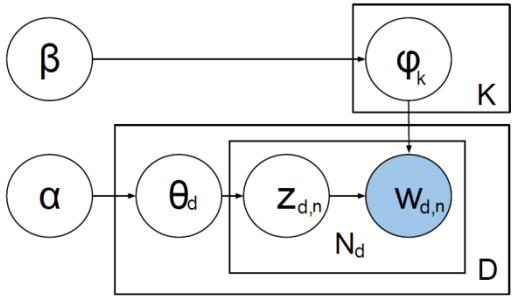

Рис. 5. Графическое представление модели скрытого размещения Дирихле.

Имеется набор из D документов. Каждый документ состоит из N_d слов, w_{dn} соответствует наблюдаемым переменным — словам в документе. Это единственные наблюдаемые переменные в модели, остальные переменные — скрытые. Переменная z_{dn} принимает значение темы, выбранной на шаге 2a для слова w_{dn}. Для каждого документа d переменная θ_d представляет собой распределение тем в этом документе.

В классической модели LDA количество тем фиксировано изначально и задаётся в явном виде параметром K. φ_k — распределение слов в теме k. Можно подобрать оптимальное значение K, варьируя его и измеряя способность модели предсказывать неизвестные данные, например, документы из тестовой выборки (см. раздел 6.1). Однако для того чтобы определять оптимальное количество тем в документе автоматически, были предложены более совершенные способы (см. раздел 5.2).

В модели LDA предполагается, что параметры θ_d и φ_t распределены следующим образом: $\theta \sim Dir(\alpha)$, $\varphi \sim Dir(\beta)$, где α и β — задаваемые вектора-параметры (т.н. *гиперпараметры*) распределения *Дирихле*. Из Байесовской теории вероятностей известно, что распределение Дирихле является *сопряжённым априорным* к *мультиномиальному* распределению, которое обычно используется для моделирования текстов. Знание сопряжённых семейств распределений существенно упрощает вычисление апостериорных вероятностей при оценке параметров модели (см. раздел 4.3.1).

Как правило, все компоненты параметров α и β распределения Дирихле берутся равными, поскольку отсутствует априорная информация о распределении слов в темах и тем в документах. Предложены подходы, позволяющие восстановить оптимальные значения гиперпараметров модели по обучающей выборке [10, 24]. На практике, как правило, используются значения, наиболее характерные для конкретных данных. К примеру, значение параметра, близкое к нулю, позволяет после оценивания параметров модели получить мультиномиальное распределение, в котором в котором бо́льшая часть плотности вероятности сосредоточена на небольшом наборе значений. Это хорошо соотносится со *степенным* распределением, которое часто наблюдается в текстах на естественном языке.

4.3.1 Оценивание параметров модели скрытого размещения Дирихле

Генеративный процесс LDA соответствует следующему совместному распределению наблюдаемых и скрытых переменных [5]:

$$P(\varphi_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} P(\varphi_i) \prod_{d=1}^{D} P(\theta_d) \left(\prod_{n=1}^{N} P(z_{dn} \mid \theta_d) p(w_{dn} \mid \varphi_{1:K}, z_{dn}) \right) \qquad (14)$$

Для определения оптимальных значений скрытых переменных модели нужно найти так называемое апостериорное распределение, т.е. условное распределение:

$$P(\varphi_{1:K}, \theta_{1:D}, z_{1:D} \mid w_{1:D}) = \frac{P(\varphi_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{P(w_{1:D})} \qquad (15)$$

Числитель дроби (15) легко вычисляется согласно (14), знаменатель представляет собой маргинальную вероятность наблюдаемых переменных, т.е. вероятность наблюдать w при всех возможных параметрах модели. Теоретически он вычисляется как сумма вероятностей совместного распределения по всем значениям скрытых переменных. Но число всевозможных назначений тем z словам w экспоненциально зависит от размера документа, поэтому на практике используются другие методы для оценивания (15).

Алгоритмы оценивания (15) делятся на две категории: на основе сэмплирования и вариационные методы. Алгоритмы первой группы пытаются собрать конечную выборку переменных, чтобы приблизить апостериорное распределение (15) эмпирическим распределением. Как правило, алгоритм принадлежит классу *методов Монте-Карло для марковских цепей (Markov Chain Monte Carlo, MCMC)*. Примером такого алгоритма является *сэмплирование по Гиббсу* [10], которое состоит в том, чтобы на каждом шаге фиксировать все переменные, кроме одной, и выбирать оставшуюся переменную согласно распределению вероятности этой переменной при

условии всех остальных (эта вероятность выводится в [10]). Недостаток этих методов в том, что они недетерминированы, так как выборки берутся случайно. Кроме того, марковская цепь может неопределённо долго сходиться к нужному распределению.

Методы второй группы — вариационные алгоритмы, детерминированная альтернатива методам на основе сэмплирования. Такие алгоритмы сначала задают параметризованное семейство распределений над скрытыми переменными, а затем с помощью EM-алгоритма ищут распределение из этого семейства, наиболее близкое к апостериорному распределению (15). Таким образом, задача оценивания параметров сводится к задаче оптимизации. При этом сознательно игнорируются некоторые зависимости между переменными. На рис. 6 приведен пример модели LDA для вариационного оценивания.

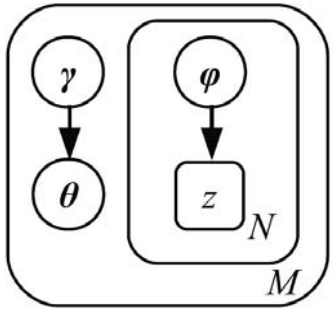

Рис. 6. Вариационная модель для оценивания параметров скрытого размещения Дирихле.

5. Второе поколение вероятностных тематических моделей

Модели, описанные ранее, применимы лишь к задачам, для которых верны следующие предположения об исходных данных [5]:

1. Последовательность слов в документе не имеет значения
2. Последовательность документов не имеет значения
3. Количество тем известно и не меняется

Если же какие-либо из данных условий не удовлетворяют поставленной задаче, то требуются более сложные модели. Например, при генерации документов, понятных человеку, последовательность слов в генерируемом документе имеет большое значение. Второе предположение может быть неверным при анализе тем в документах из большого временного промежутка. Например, тема, описывающая какую-нибудь научную область, может иметь разное распределение и состав в разные промежутки времени: с течением

времени из-за смены приоритетов и терминологии какие-то термины начинают встречаться чаще, а какие-то реже. Третье предположение работает только в том случае, если в задаче априорно известно количество тем в документах, что на практике выполняется редко.

В данном разделе рассмотрены модификации LDA и другие модели, позволяющие снять некоторые из перечисленных ограничений и расширить область применения тематического моделирования. Обзоры разнообразных подходов, предложенных за 10 лет активного развития тематического моделирования, приведены в [5, 26, 27].

5.1. Иерархическое скрытое размещение Дирихле

Одним из недостатков скрытого размещения Дирихле является тенденция к извлечению слишком общих тем для заданного набора документов. В случае, когда некоторая концепция имеет ряд аспектов (смыслов), которые часто употребляются совместно с основной концепцией, в результатах работы LDA с большой вероятностью будет присутствовать тема, которая включает как основной смысл концепции, так и все её аспекты. Часто необходимо, чтобы в отдельные темы была выделена не только основная концепция, но и различные её аспекты. В таких случаях используются иерархические тематические модели, позволяющие моделировать иерархию тем — от более общих до узких.

Модель иерархического скрытого размещения Дирихле (Hierarchical Latent Dirichlet Allocation, hLDA), описанная в работе [11], основана на вложенном процессе китайского ресторана (Nested Chinese Restaurant Process, nCRP). Логично сначала рассмотреть стандартный процесс китайского ресторана (Chinese Restaurant Process, CRP), который генерирует распределение M объектов (клиентов) по неограниченному числу разделов (столов).

Пусть некоторый китайский ресторан имеет неограниченное (счетное) количество столов. В него по очереди заходят M клиентов. Первый клиент садится за первый стол. Очередной клиент с номером m выбирает стол согласно распределению:

$$p(\textit{занятый стол } i \mid \textit{клиенты } \overline{1,\mathrm{m}-1}) = \frac{m_i}{\gamma + m - 1}$$
$$p(\textit{первый свободный стол} \mid \textit{клиенты } \overline{1,\mathrm{m}-1}) = \frac{\gamma}{\gamma + m - 1} \qquad (16)$$

Здесь γ - так называемый *концентрационный параметр* процесса.

Таким образом, если клиент садится за занятый стол, то с большей вероятностью он занимает стол с большим количеством клиентов, с каждым клиентом уменьшается вероятность занять новый стол. Причем распределение

объектов (клиентов) по разделам (столам) получается такое же, как из процесса Дирихле (см. раздел 5.2).

Процесс китайского ресторана можно расширить до вложенного процесса китайского ресторана [11]. Пусть в городе имеется бесконечное (счетное) число ресторанов. Каждый стол в ресторане содержит ссылку на другой ресторан. Пусть имеется один корневой ресторан, и в каждый ресторан ведет только одна ссылка. Таким образом, получается древовидная структура ресторанов.

Клиент прибывает в город на L дней. В первый вечер посещает корневой ресторан, выбирая стол согласно (16). На следующий день он идет в ресторан, определенный выбранным в корневом ресторане столом, снова выбирает стол согласно (16) и так далее. Каждый день клиент посещает один из ресторанов. Таким образом, он посетит L ресторанов. После того, как город посетят M клиентов, коллекция их путей по ресторанам будет представлять конечное поддерево глубины L бесконечного дерева ресторанов.

Полученное дерево может быть использовано для моделирования иерархии тем. В модели иерархического скрытого размещения Дирихле [11] каждому ресторану из процесса китайского ресторана соответствует тема. Генеративный процесс следующий:

1. Пусть c_1 — корневой ресторан
2. Для каждого уровня дерева $l \in \{2, \dots, L\}$:
    a. Выбрать стол в ресторане c_{l-1} согласно (16). Установить c_l — ресторан, на который ссылается выбранный стол
3. Случайно выбрать для документа его распределение по L темам $\theta_d \sim Dir(\alpha)$
4. Для каждого слова в документе:
    a. Случайно выбрать $z \in \{1, \dots, L\}$ согласно распределению θ_d
    b. Случайно выбрать слово из распределения слов в теме, соответствующему ресторану c_z

Схема модели hLDA изображена на рис. 7. Здесь T — дерево ресторанов, получаемое с помощью вложенного процесса китайского ресторана, $c_1, c_2, \dots, c_L$ — путь по ресторанам, причем значение c_i зависит от $c_1, c_2, \dots, c_{i-1}$. Способ оценивания параметров модели описан в [11].

Описанная модификация расширяет модель LDA, добавляя возможность существования неограниченного количества тем. Однако количество тем, описывающих один документ, по-прежнему постоянно и равно L.

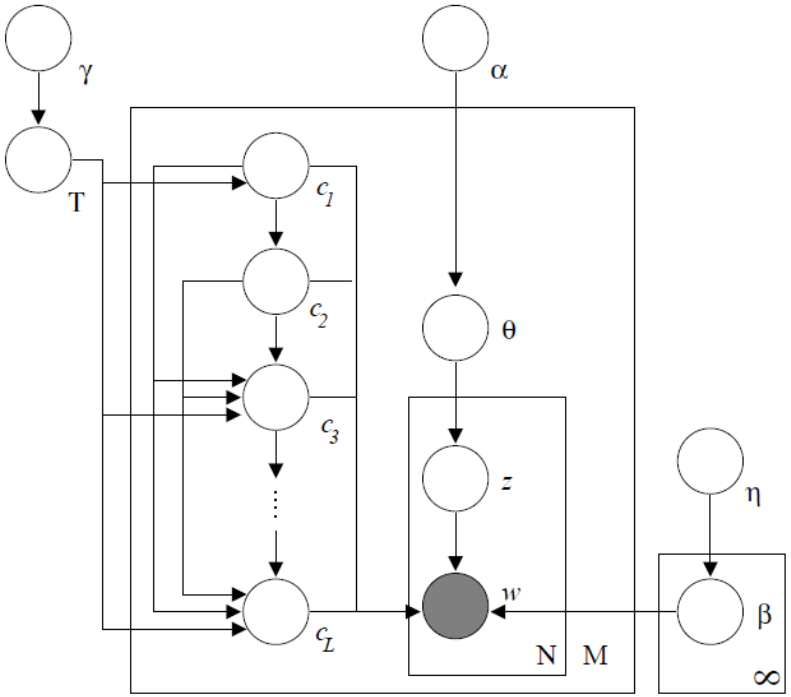

Рис. 7. Графическое представление модели иерархического скрытого размещения Дирихле.

5.2. Непараметрические модели

Зачастую невозможно предсказать, сколько тем встретится в наборе данных, например, при анализе потока неизвестных документов, новостей и т.д. Поэтому в этих случаях используются непараметрические модели.

Иерархический процесс Дирихле (Hierarchical Dirichlet Process, HDP) является Байесовской непараметрической моделью [12, 13], которая может быть использована для тематического моделирования с потенциально бесконечным числом тем.

Случайный процесс $G \sim DP(\alpha, H)$ называется *процессом Дирихле* с *базовым распределением* H и *параметром концентрации* α, если для произвольного конечного разбиения $A_1, A_2, \dots, A_r$ вероятностного пространства над H выполняется:

$$(G(A_1), G(A_2), \dots, G(A_r)) \sim Dir(\alpha H(A_1), \alpha H(A_2), \dots, \alpha H(A_r)) \qquad (17)$$

Здесь $G(Ai)$ и $H(Ai)$ — маргинальные вероятности G и H над A_i.

Процесс Дирихле может быть представлен как процесс «*ломания палки*». Интуитивная интерпретация процесса предполагает, что имеется палка длины 1. Сначала ломают ее в точке β_1 (из бета-распределения (18) с параметром α) и получают пропорцию π_1. Затем повторяют процесс для оставшейся $1-\beta_1$ части палки, получая π_2, π_3 и т.д. Пропорция π_k кластера k определена следующим образом:

$$\beta_k \sim Beta(1,\alpha)$$
$$\pi_k = \beta_k \prod_{l=1}^{k-1}(1-\beta_l) \qquad (18)$$

Иерархический процесс Дирихле можно понимать как процесс Дирихле над процессом Дирихле. Иначе говоря, при генерации случайного элемента из иерархического процесса Дирихле сначала выбирается, из какого процесса Дирихле верхнего уровня следует генерировать элемент, после чего к выбранному процессу Дирихле применяется стандартная схема генерации элемента. Он моделирует документы, в которых имеются некоторые темы, общие для всего набора данных, а также более специфичные темы. В работе [13] предложен способ онлайнового вариационного оценивания параметров процесса Дирихле.

5.3. Модели, учитывающие временной фактор

С течением времени темы могут появляться, изменяться и исчезать. Для решения задач, где нужно не только найти темы, но и проследить их динамику, используются темпоральные тематические модели.

В данном разделе рассмотрены две модели: в первой время делится на отрезки, каждый отрезок рассматривается как атомарная единица, во второй время рассматривается как непрерывная величина.

Динамическая тематическая модель, предложенная в [15], отслеживает эволюцию тем в последовательно организованном корпусе документов. В этой статье документы группируются по годам (корпус содержит документы за 100 лет), и документы каждого года генерируются согласно темам, произошедшим от тем из прошлого года.

Схема предложенной модели представлена на рис. 8. Для каждого интервала времени имеются свои распределения тем в документе θ и распределения слов в темах β. Причем эти распределения зависят от соответствующих распределений в предыдущем временном отрезке. Генеративный процесс следующий:

1. Сгенерировать темы $\beta_t|\beta_{t-1}\sim\mathcal{N}(\beta_{t-1},\sigma^2 I)$

2. Сгенерировать $\alpha_t|\alpha_{t-1}\sim\mathcal{N}(\alpha_{t-1},\sigma^2 I)$
3. Для каждого документа:
   a. Сгенерировать распределение $\theta\sim\mathcal{N}(\alpha_t,a^2 I)$
   b. Для каждого слова сгенерировать тему из θ и слово из соответствующего β

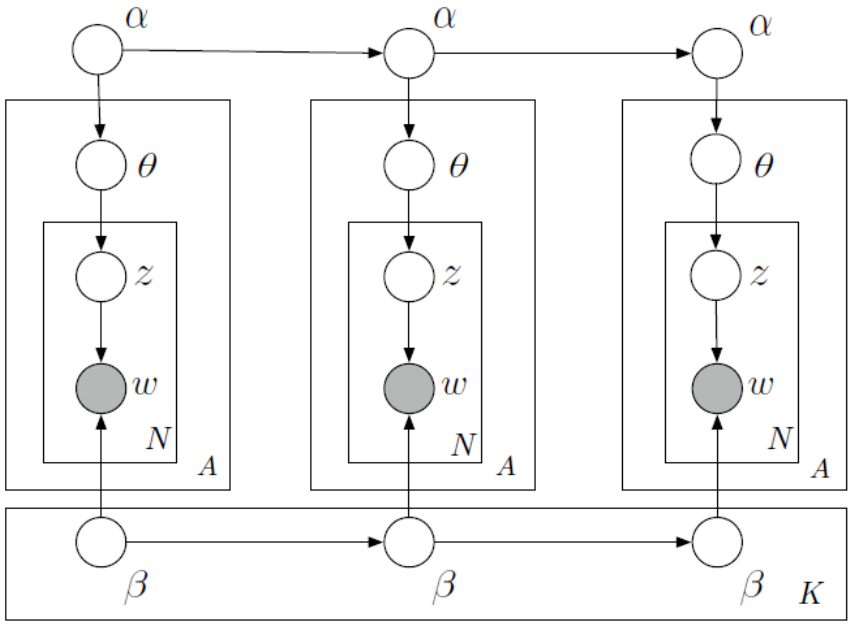

Рис. 8. Графическое представление динамической тематической модели.

В отличие от описанной выше модели, в работе [16] документы не делятся на дискретные группы, т.е. временной фактор непрерывный. Оценивание скрытых параметров модели позволяет найти темы, которые учитывают как одновременную встречаемость слов (как в стандартном LDA), так и локальность слов во времени.

В стандартную модель LDA для каждого документа добавляется наблюдаемая переменная — время t, которое зависит от темы и от переменной ψ, которая служит параметром априорного распределения t.

Процесс генерации аналогичен модели LDA. Основное отличие в том, что при генерации каждого слова генерируется переменная t из бета-распределения ψ. Схема модели изображена на рис. 9.

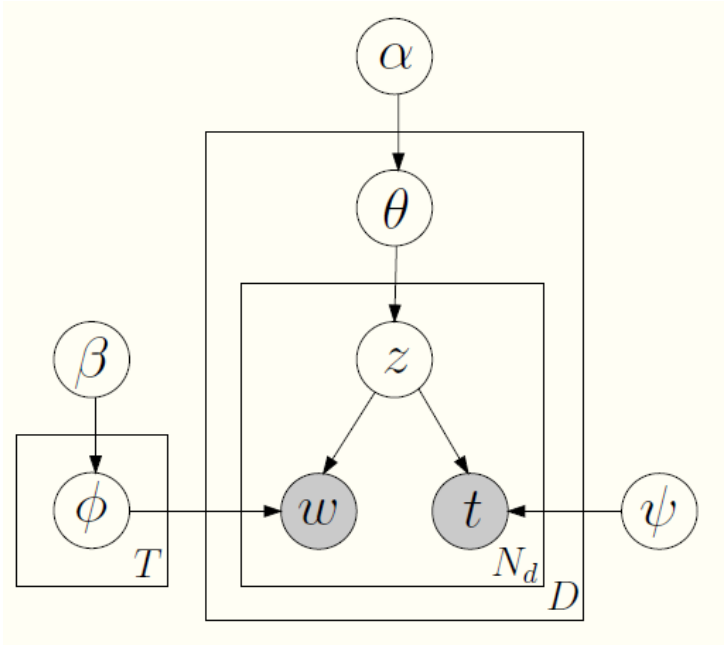

Рис. 9. Графическое представление тематической модели с учетом непрерывного временного фактора.

5.4. Онлайн-модели

В современном мире информация все чаще представляется в виде потоков. Поэтому необходимы алгоритмы, работающие не с фиксированным набором данных, а с данными, обновляющимися в режиме реального времени. Стандартная модель LDA не подходит для задачи поиска тем в режиме реального времени, потому что при появлении в потоке нового документа нужно полностью пересчитывать параметры модели на всех данных.

В работе [17] рассматривается онлайновое оценивание параметров модели LDA, при котором каждый документ обрабатывается один раз. Метод представляет собой модификацию вариационного оценивания. Все документы разбиваются на небольшие порции. Запускается итерационный процесс, на каждом шаге которого обрабатываются только данные из соответствующей порции и вычисляется оптимальный параметр постулируемого апостериорного распределения, задаваемого вариационным алгоритмом. После каждого шага этот параметр обновляется с учетом значения, вычисленного на предыдущей порции. Как правило, параметр вычисляется как взвешенное среднее текущего и предыдущего значений.

Кроме вариационного онлайнового оценивания параметров LDA существуют методы, являющиеся онлайн-модификациями сэмплирования по Гиббсу. В

работе [18] рассматривается алгоритм сэмплирования по Гиббсу, который выбирает значение темы z_i для слова w_i один раз для каждого слова. При этом сначала вычисляются параметры z для первых нескольких документов согласно стандартному алгоритму. А затем запускается итерационный процесс: для каждого последующего слова w_i выбирается назначение темы z_i, при условии значений z для предыдущих слов. Существенной проблемой данного алгоритма является зависимость от качества работы первого этапа — сэмплирования тем для первых документов, — так как все последующие темы выбираются на их основе. В связи с этим также рассматривается модификация алгоритма, в которой после сэмплирования темы для очередного слова для некоторых предыдущих слов производится повторное сэмплирование, то есть z_i выбирается заново.

5.5. Модели, учитывающие пользовательские метки

Многие социальные сервисы позволяют пользователям реагировать на объекты или оставлять дополнительную информацию. Эта информация может быть представлена в разных формах: в виде комментариев, оценок, меток и др. Ее можно использовать для улучшения качества результатов алгоритмов, работающих с содержимым социальных сетей.

В работе [19] рассматривается модификация LDA, которая учитывает метки (или *тэги*), которые пользователи назначают текстовым объектам.

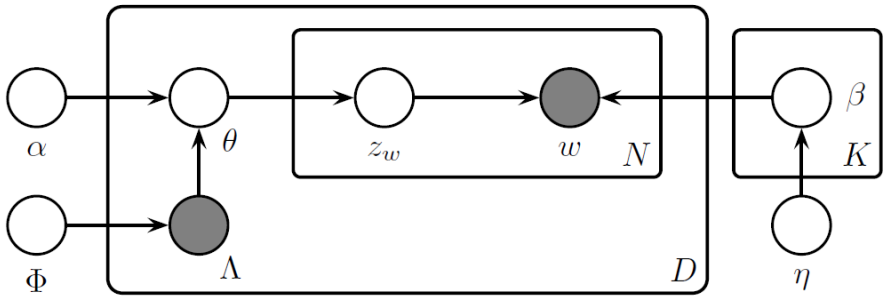

Рис. 10. Графическое представление тематической модели с учетом меток.

Схема предложенной модели представлена на рис. 10. Основная её особенность в том, что для каждого документа появляется наблюдаемая переменная Λ, которая соответствует тэгам пользователей. Λ — это бинарный вектор размерности K, где K — количество различных тэгов на всем наборе данных. При этом i-я компонента Λ равна 1, если документ помечен соответствующим i-м тэгом, 0 — в противном случае. В данной модели темы отождествляются с тэгами, поэтому количество различных тэгов совпадает с

количеством тем K. Распределение тем в документе θ зависит не только от априорного распределения, но и от наблюдаемых меток: в каждом документе вычисляется распределение только по тем темам, которые соответствуют тэгам с $\Lambda_i=1$, для остальных тем вероятность равна 0. Для оценивания скрытых параметров в работе [19] используется сэмплирование по Гиббсу, в котором каждая переменная z_w выбирается только из соответствующего набора тем.

6. Методы оценивания качества результатов

6.1. Обобщающая способность модели

Самым распространённым способом оценивания качества вероятностных тематических моделей является расчёт *перплексии* [31] на тестовом наборе данных D_{test} из M документов:

$$\text{Перплексия}(D_{test}) = \exp\left\{-\frac{\sum_{d=1}^{M}\log p(w_d|\text{модель})}{\sum_{d=1}^{M} N_d}\right\} \qquad (19)$$

Способы расчёта вероятности нового документа w_d при условии известных параметров модели рассмотрены в работе [32].

Если немного изменить способ расчёта перплексии и оценивать вероятности для каждого слова w_{dn} из тестового набора документов:

$$\text{Перплексия}(D_{test}) = \exp\left\{-\sum_{d=1}^{M}\sum_{n=1}^{N_d}\log p(w_{dn}|\text{модель})\right\}, \qquad (20)$$

то полученное значение соответствует полезному размеру словаря модели. Например, значение 100 означает, что набор вероятностей полученной модели эквивалентен случайному выбору каждого слова из словаря размером в 100 слов [14]. Таким образом, меньшее значение перплексии означает, что модель лучше описывает (обобщает) тестовые данные. Кроме того, минимизируя значение этого критерия, можно экспериментально подобрать оптимальное число различных тем в коллекции документов.

Альтернативным подходом является оценивание вероятности второй части документа при условии наличия первой [35]. Для этого каждый документ разделяют на 2 части: первую часть считают обучающими данными, а с помощью второй тестируют качество модели.

6.2. Эффективность приложений

Тестирование моделей на уровне приложений позволяет оценить их применимость к конкретной задаче и данным. К примеру, в работе [21] приведены результаты использования 4 различных тематических моделей в задаче фильтрации спама, а также сформулированы рекомендации для их применения к разным наборам данных. Авторы [25] исследовали

производительность моделей LSI и LDA в сочетании с популярным алгоритмом классификации *метод опорных векторов (Support Vector Machine, SVM)* в применении к классической задаче классификации документов. Продемонстрировано значительное увеличение точности классификации с использованием тематических моделей по сравнению с обычным векторным представлением.

6.3. Интерпретируемость

Приложения, которые предполагают непосредственное взаимодействие пользователя с результатами тематического моделирования, должны также учитывать их *интерпретируемость*. В исследовании [33] было показано, что модели с наименьшей перплексией (см. раздел 6.1) обычно хуже интерпретируются обычными людьми. Однако предложенный авторами метод оценивания интерпретируемости моделей предполагает активное участие пользователей и не применим в общем случае.

Авторами [34] были предложены методы автоматического оценивания *связности* найденных тем с помощью внешних баз знаний (*WordNet, Wikipedia, Google*). Наибольшую согласованность с мнениями экспертов показал метод, основанный на расчёте значения *поточечной взаимной информации (pointwise mutual information, PMI)* для пар терминов (w_i, w_j), составляющих тему, на полном корпусе статей Википедии (~2 млн. статей с ~1 млрд. слов):

$$PMI(w_i, w_j) = \log\frac{p(w_i, w_j)}{p(w_i)p(w_j)} \qquad (21)$$

7. Программные реализации

Некоторые реализации вероятностных тематических моделей представлены в табл. 1. Кроме того, на веб-странице автора LDA и других тематических моделей Дэвида Блея [23] доступны реализации оригинальных методов, описанных им и его коллегами в статьях.

Фреймворки, позволяющие описать произвольную модель и оценить её оптимальные параметры, описаны в табл. 2.

В табл. 3 представлены библиотеки, упрощающие использование методов тематического моделирования с помощью графического интерфейса.

Табл. 1. Свободные реализации вероятностных тематических моделей.

Название	Язык	Алгоритмы оценивания параметров	Ссылка
LDA-C	C	вариационный EM	www.cs.princeton.edu/~blei/lda-c
Mallet	Java	сэмплирование по Гиббсу	mallet.cs.umass.edu/topics.php
GibbsLDA ++	C/C++	сэмплирование по Гиббсу	gibbslda.sourceforge.net
Gensim	Python	сэмплирование по Гиббсу	radimrehurek.com/gensim
Matlab Topic Modeling Toolbox	Matlab	сэмплирование по Гиббсу	psiexp.ss.uci.edu/research/programs_data/toolbox.htm
Stanford Topic Modeling Toolbox	Scala	коллапсированное сэмплирование по Гиббсу, коллапсированная вариационная Байесовская аппроксимация	nlp.stanford.edu/software/tmt
GraphLab	C++	коллапсированное сэмплирование по Гиббсу	docs.graphlab.org/topic_modeling.html
Yahoo LDA	C++	сэмплирование по Гиббсу (MapReduce)	github.com/shravanmn/Yahoo_LDA
Mahout	Java	вариационный EM (MapReduce)	cwiki.apache.org/confluence/display/MAHOUT/Latent+Dirichlet+Allocation

Табл. 2. Фреймворки для вероятностного моделирования.

Название	Язык	Алгоритмы оценивания параметров	Ссылка
PyMC	Pyhton	методы Монте-Карло для марковских цепей	github.com/pymc-devs/pymc
Factorie	Scala	методы Монте-Карло для марковских цепей, вариационный EM	factorie.cs.umass.edu
Open BUGS	C	сэмплирование по Гиббсу	openbugs.info/w

Табл. 3. Библиотеки для тематического моделирования с графическим интерфейсом.

Название	Ссылка
WinBUGS	www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml
Topic Modeling Tool	code.google.com/p/topic-modeling-tool

8. Заключение

В статье были продемонстрированы эволюция и современное состояние тематического моделирования текстов на естественном языке, которое является перспективным инструментом для обработки больших коллекций документов в приложениях информационного поиска и анализа текстов.

Известно, что бо́льшая часть текстов на сегодняшний день создаётся и публикуется пользователями Интернета в свободной форме. Вместе с тем, именно автоматизированный анализ растущих объёмов подобной информации способен дать заинтересованным организациям уникальную возможность для отслеживания актуальных трендов, а также понимания заинтересованности потребителей в тех или иных продуктах, товарах или услугах.

В связи с этим можно выделить следующие перспективные направления развития тематического моделирования:

- Анализ пользовательского контента (блоги, форумы, социальные сети, рецензии, отзывы и др.);
- Использование дополнительных данных: социального профиля и связей пользователей, метаданных документов и связей между ними, внешних энциклопедий и онтологий, статистики просмотра страниц, записей о реакциях пользователей, временных меток и т.д.;
- Разработка методов обучения и вывода по моделям для распределённой обработки больших потоков данных в реальном времени;
- Разработка способов применения тематического моделирования к новым типам данных (изображения, аудио- и видеофайлы, геном, финансовая статистика и т.д.).

Список литературы

[1] James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. *Topic Detection and Tracking Pilot Study. Final Report*. Proceedings of the Broadcast News Transcription and Understanding Workshop (Sponsored by DARPA), Feb. 1998

[2] A.K. Jain, M.N. Murty, P.J. Flynn. *Data Clustering: A Review*; ACM Computing Surveys, Vol. 31, No. 3, September 1999

[3] Fabrizio Sebastiani. *Machine Learning in Automated Text Categorization*, ACM Computing Surveys, Vol. 34, No.1, pp.1-47, 2002.

[4] Allan, J. and Lavrenko, V. and Malin, D. and Swan, R. *Detections, bounds, and timelines: UMass and TDT-3.* In Proceedings of Topic Detection and Tracking Workshop, pages 167–174.p. 167-174, Vienna, VA, 2000
[5] Blei, David M. (April 2012). *Introduction to Probabilistic Topic Models*. Comm. ACM 55 (4): 77–84.
[6] Thomas Hofmann. *Probabilistic Latent Semantic Analysis*. UAI 1999: 289-296
[7] Thomas Hofmann. *Probabilistic Latent Semantic Indexing*. SIGIR 1999: 50-57
[8] T.K. Moon. *The expectation-maximization algorithm*. IEEE Signal Processing Mag., vol. 13, pp. 47–60, Nov. 1996
[9] D. Blei, A. Ng, and M. Jordan. *Latent Dirichlet allocation*. Journal of Machine Learning Research, 3:993–1022, January 2003
[10] Gregor Heinrich. *Parameter estimation for text analysis*. Technical report, Fraunhofer IGD, 2005
[11] D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. *Hierarchical topic models and the nested Chinese restaurant process*. Neural Information Processing Systems 16, 2003
[12] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal and David M. Blei. *Hierarchical Dirichlet Processes*. Journal of the American Statistical Association, 101:476, 1566-1581, 2006
[13] C. Wang, J. Paisley, and D. Blei. *Online variational inference for the hierarchical Dirichlet process*. Artificial Intelligence and Statistics , 2011
[14] *Mining Text Data* (Springer) Ed. Charu Aggarwal, ChengXiang Zhai, March 2012
[15] D. Blei and J. Lafferty. *Dynamic topic models*. In Proceedings of the 23rd International Conference on Machine Learning, 2006
[16] Xuerui Wang, Andrew McCallum. *Topics over time: a non-Markov continuous-time model of topical trends*. KDD 2006: 424-433
[17] M. Hoffman, D. Blei, and F. Bach. *Online learning for latent Dirichlet allocation*. Neural Information Processing Systems, 2010
[18] Kevin Robert Canini, Lei Shi, Thomas L. Griffiths. *Online Inference of Topics with Latent Dirichlet Allocation*. Journal of Machine Learning Research - Proceedings Track 5: 65-72 (2009)
[19] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled LDA. *A supervised topic model for credit attribution in multi-labeled corpora*. In Empirical Methods in Natural Language Processing, pages 248–256, 2009
[20] G. Lisowsky and L. Rost. *Konkordanz zum hebräischen Alten Testament*. Deutsche Bibelgesellschaft, 1958.
[21] Lee, S., Song, J., and Kim, Y. *An Empirical Comparison of Four Text Mining Methods*. Journal of Computer Information Systems, (51:1), 2010, pp. 1-10
[22] D. Blei and J. Lafferty. *Topic Models*. In A. Srivastava and M. Sahami, editors, Text Mining: Classification, Clustering, and Applications. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, 2009
[23] David M. Blei topic modeling page - http://www.cs.princeton.edu/~blei/topicmodeling.html
[24] D. Mimno and A. McCallum. *Topic models conditioned on arbitrary features with dirichlet-multinomial regression*. In UAI, 2008
[25] Zelong Liu, Maozhen Li, Yang Liu, Mahesh Ponraj. *Performance evaluation of Latent Dirichlet Allocation in text mining*. FSKD 2011: 2695-2698
[26] Steyvers, M. & Griffiths, T. *Probabilistic topic models*. In T. Landauer, D McNamara, S. Dennis, and W. Kintsch (eds), Latent Semantic Analysis: A Road to Meaning. Laurence Erlbaum, 2007

[27] Ali Daud, Juanzi Li, Lizhu Zhou, Faqir Muhammad. *Knowledge discovery through directed probabilistic topic models: a survey*. In Proceedings of Frontiers of Computer Science in China. 2010, 280-301. — перевод на русский К. В. Воронцов, А. В. Темлянцев и др.
[28] Buntine W. L. *Operations for learning with graphical models*. Journal of Artificial Intelligence Research, 1994, 2: 159 – 225
[29] S. Choi, S. Cha, C. C. Tappert. *A Survey of Binary Similarity and Distance Measures*, Journal of Systemics, Cybernetics and Informatics, Vol 8 No 1 2010, pp 43-48
[30] Rui Xu, Donald C. Wunsch II. *Survey of clustering algorithms*. IEEE Transactions on Neural Networks 16(3): 645-678 (2005)
[31] L. Bahl, J. Baker, E. Jelinek, and R. Mercer. *Perplexity — a measure of the difficulty of speech recognition tasks*. In Program, 94th Meeting of the Acoustical Society of America, volume 62, page S63, 1977
[32] H. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. *Evaluation methods for topic models*. In Proceedings of the 26th International Conference on Machine Learning (ICML 2009), 2009
[33] Jonathan Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang, David M. Blei. *Reading Tea Leaves: How Humans Interpret Topic Models*. NIPS 2009: 288-296
[34] Newman, Lau, Grieser, Baldwin. *Automatic Evaluation of Topic Coherence*. NAACL HLT 2010
[35] Rosen-Zvi, M., Griffiths, T., Steyvers, M., & Smyth, P. (2004). *The author-topic model for authors and documents*. Proc. of Conf. on Uncertainty in Artificial Intelligence (UAI'04) (pp. 487–494)
[36] Dumais, S. T., Furnas, G. W., Landauer, T. K. and Deerwester, S. (1988). *Using latent semantic analysis to improve information retrieval*. In Proceedings of CHI'88: Conference on Human Factors in Computing, New York: ACM, 281-285
[37] Sanjeev Arora, Rong Ge, Ankur Moitra. *Learning Topic Models - Going beyond SVD*. CoRR abs/1204.1956 (2012)
[38] Daniel D. Lee and H. Sebastian Seung (1999). *Learning the parts of objects by non-negative matrix factorization*. Nature 401 (6755): 788–791

Topic modeling in natural language texts

Anton Korshunov, Andrey Gomzin
{korshunov, gomzin}@ispras.ru
ISP RAS, Moscow, Russia

Abstract. Topic modeling is a method for building a model of a collection of text documents. The model is able to determine topics for each of documents. Shifting from term space to space of extracted topics helps resolving synonymy and polysemy of terms. Besides, it allows for more efficient topic-sensitive search, classification, summarization, and annotation of document collections and news feeds. The paper shows an evolution of topic modeling techniques. The earlier methods are based on clustering. These algorithms use some similarity function defined on two documents. The next generation of topic modeling techniques is based on Latent Semantic Indexing (LSA). Words co-occurrences in documents are analyzed here. Currently, the most popular are approaches based on Bayesian networks — directed probabilistic graphical models which incorporate different kinds of entities and metadata: document authorship, connections between words, topics, documents, and authors, etc. The paper contains a comparative survey of different models along with methods for parameter estimation and accuracy measurement. The following topic models are considered in the paper: Probabilistic Latent Semantic Indexing, Latent Dirichlet Allocation, non-parametric models, dynamic models, and semi-supervised models. The paper describes well-known quality evaluation metrics: perplexity and topic coherence. Freely available implementations are listed as well.

Keywords: topic modeling; topic-sensitive search; document classification; probabilistic graphical models; Bayesian networks; latent Dirichlet allocation; dimensionality reduction; text mining; information retrieval; machine learning.

References

[1]. James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic Detection and Tracking Pilot Study. Final Report. Proceedings of the Broadcast News Transcription and Understanding Workshop (Sponsored by DARPA), Feb. 1998

[2]. A.K. Jain, M.N. Murty, P.J. Flynn. Data Clustering: A Review; ACM Computing Surveys, Vol. 31, No. 3, September 1999

[3]. Fabrizio Sebastiani. Machine Learning in Automated Text Categorization, ACM Computing Surveys, Vol. 34, No.1, pp.1-47, 2002.

[4]. Allan, J. and Lavrenko, V. and Malin, D. and Swan, R. Detections, bounds, and timelines: UMass and TDT-3. In Proceedings of Topic Detection and Tracking Workshop, pages 167–174.p. 167-174, Vienna, VA, 2000

[5]. Blei, David M. (April 2012). Introduction to Probabilistic Topic Models. Comm. ACM 55 (4): 77–84.

[6]. Thomas Hofmann. Probabilistic Latent Semantic Analysis. UAI 1999: 289-296

[7]. Thomas Hofmann. Probabilistic Latent Semantic Indexing. SIGIR 1999: 50-57

[8]. T.K. Moon. The expectation-maximization algorithm. IEEE Signal Processing Mag., vol. 13, pp. 47–60, Nov. 1996

[9]. D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. Journal of Machine Learning Research, 3:993–1022, January 2003

[10]. Gregor Heinrich. Parameter estimation for text analysis. Technical report, Fraunhofer IGD, 2005

[11]. D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. Neural Information Processing Systems 16, 2003

[12]. Yee Whye Teh, Michael I. Jordan, Matthew J. Beal and David M. Blei. Hierarchical Dirichlet Processes. Journal of the American Statistical Association, 101:476, 1566-1581, 2006

[13]. C. Wang, J. Paisley, and D. Blei. Online variational inference for the hierarchical Dirichlet process. Artificial Intelligence and Statistics , 2011

[14]. Mining Text Data (Springer) Ed. Charu Aggarwal, ChengXiang Zhai, March 2012

[15]. D. Blei and J. Lafferty. Dynamic topic models. In Proceedings of the 23rd International Conference on Machine Learning, 2006

[16]. Xuerui Wang, Andrew McCallum. Topics over time: a non-Markov continuous-time model of topical trends. KDD 2006: 424-433

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

[18]. Kevin Robert Canini, Lei Shi, Thomas L. Griffiths. Online Inference of Topics with Latent Dirichlet Allocation. Journal of Machine Learning Research - Proceedings Track 5: 65-72 (2009)

[19]. D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled LDA. A supervised topic model for credit attribution in multi-labeled corpora. In Empirical Methods in Natural Language Processing, pages 248–256, 2009

[20]. G. Lisowsky and L. Rost. Konkordanz zum hebräischen Alten Testament. Deutsche Bibelgesellschaft, 1958.

[21]. Lee, S., Song, J., and Kim, Y. An Empirical Comparison of Four Text Mining Methods. Journal of Computer Information Systems, (51:1), 2010, pp. 1-10

[22]. D. Blei and J. Lafferty. Topic Models. In A. Srivastava and M. Sahami, editors, Text Mining: Classification, Clustering, and Applications. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, 2009

[23]. David M. Blei topic modeling page - http://www.cs.princeton.edu/~blei/topicmodeling.html

[24]. D. Mimno and A. McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In UAI, 2008

[25]. Zelong Liu, Maozhen Li, Yang Liu, Mahesh Ponraj. Performance evaluation of Latent Dirichlet Allocation in text mining. FSKD 2011: 2695-2698

[26]. Steyvers, M. & Griffiths, T. Probabilistic topic models. In T. Landauer, D McNamara, S. Dennis, and W. Kintsch (eds), Latent Semantic Analysis: A Road to Meaning. Laurence Erlbaum, 2007

[27]. Ali Daud, Juanzi Li, Lizhu Zhou, Faqir Muhammad. Knowledge discovery through directed probabilistic topic models: a survey. In Proceedings of Frontiers of Computer Science in China. 2010, 280-301.

[28]. Buntine W. L. Operations for learning with graphical models. Journal of Artificial Intelligence Research, 1994, 2: 159 – 225

[29]. S. Choi, S. Cha, C. C. Tappert. A Survey of Binary Similarity and Distance Measures, Journal of Systemics, Cybernetics and Informatics, Vol 8 No 1 2010, pp 43-48

[30]. Rui Xu, Donald C. Wunsch II. Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3): 645-678 (2005)

[31]. L. Bahl, J. Baker, E. Jelinek, and R. Mercer. Perplexity — a measure of the difficulty of speech recognition tasks. In Program, 94th Meeting of the Acoustical Society of America, volume 62, page S63, 1977
[32]. H. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In Proceedings of the 26th International Conference on Machine Learning (ICML 2009), 2009
[33]. Jonathan Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang, David M. Blei. Reading Tea Leaves: How Humans Interpret Topic Models. NIPS 2009: 288-296
[34]. Newman, Lau, Grieser, Baldwin. Automatic Evaluation of Topic Coherence. NAACL HLT 2010
[35]. Rosen-Zvi, M., Griffiths, T., Steyvers, M., & Smyth, P. (2004). The author-topic model for authors and documents. Proc. of Conf. on Uncertainty in Artificial Intelligence (UAI'04) (pp. 487–494)
[36]. Dumais, S. T., Furnas, G. W., Landauer, T. K. and Deerwester, S. (1988). Using latent semantic analysis to improve information retrieval. In Proceedings of CHI'88: Conference on Human Factors in Computing, New York: ACM, 281-285
[37]. Sanjeev Arora, Rong Ge, Ankur Moitra. Learning Topic Models - Going beyond SVD. CoRR abs/1204.1956 (2012)
[38]. Daniel D. Lee and H. Sebastian Seung (1999). Learning the parts of objects by non-negative matrix factorization. Nature 401 (6755): 788–791