Полностью разреженные тематические модели (Fully Sparse Topic Models)

1. Введение

Тематическое моделирование – активно развивающаяся область исследований. Особенный интерес в последнее время приобрели тематические модели большого масштаба. Обычно речь идет о следующих свойствах задач:

(a) большое число обучающих документов;
(b) большое число тем;
(c) большой размер словаря;
(d) большое число документов, которые надо описать уже готовой моделью за ограниченное время.

Большинство предыдущих работ концентрировались на свойствах (a) и (b), это были LDA с параллельной/распределенной/онлайн архитектурой. Однако при большом размере словаря возникали проблемы. Основная причина в том, что используемое в LDA распределение Дирихле не позволяет профилям тем и профилям документов содержать нули, поэтому они плотные и занимают много места. Кроме того, описание документов с помощью модели LDA часто бывает медленным, а это не подходит для задач со свойством (d).

Частый подход – использовать регуляризацию, чтобы потребовать разреженность от профилей тем и/или документов. Он приводит к моделям RLSI, SRS, STC. Однако в SRS процесс обучения модели/описания документов не обязательно сходится, кроме того, не известна масштабируемость; обучение STC очень сложное – необходимо решать оптимизационную задачу с большим числом неразделяемых переменных; у RLSI большая сложность и алгоритма обучения модели, и описания документов. У всех моделей 1. есть параметры регуляризации, приходится как-то их выбирать, и это трудно в больших задачах, 2. нельзя четко задать желаемый баланс между разреженностью решения и качеством, а также временем работы.

В этой работе мы представляем модель FSTP – упрощенный вариант LDA и PLSA. В модели предполагается, что корпус документов порожден из смеси тем $\boldsymbol{\varphi}_1, \dots \boldsymbol{\varphi}_{|T|}$, и каждый документ генерируется следующим образом:

1. Выбрать случайно распределение по темам $\boldsymbol{\theta}$.
2. Для каждого слова документа d:
   — выбрать тему t с вероятностью $p(t \mid d) = \theta_t$;
   — сгенерировать слово w с вероятностью $p(w \mid t) = \varphi_{wt}$.

FSTM не использует распределение Дирихле, что позволит создавать разреженные профили и экономить память. В совокупности с линейным по времени обучением модели это позволяет успешно работать со свойствами (a)-(d).

2. Обучение модели

Обучение модели представляет собой EM-алгоритм. На E-шаге мы описываем документы (то есть находим Θ) по фиксированным темам (то есть Φ), этот процесс называем inference. На M-шаге наоборот обучаем темы по фиксированным описаниям документов (learning).

E-шаг: inference. Пусть даны профили тем $\boldsymbol{\varphi}_1, \dots \boldsymbol{\varphi}_{|T|}$ и некоторый документ $\boldsymbol{d}$. Необходимо найти вектор его латентного представления $\boldsymbol{\theta} = (\theta_1, \dots \theta_{|T|})$. По принципу максимума правдоподобия это такой вектор $\boldsymbol{\theta}$, что:

$$\ln P(\boldsymbol{d}) = \sum_{w \in d} n_{dw} \, ln \sum_{t \in T} \theta_t \, \varphi_{wt} \to \max .$$

Обозначим $x_w = \sum_{t \in T} \theta_t \, \varphi_{wt}$, $\boldsymbol{x} = (x_1, \dots x_{|W|})$. Тогда вектор $\boldsymbol{x} = \sum_{t \in T} \theta_t \, \boldsymbol{\varphi}_t$ – это выпуклая линейная комбинация профилей тем $\boldsymbol{\varphi}_1, \dots \boldsymbol{\varphi}_{|T|}$. Таким образом, мы можем перейти от задачи поиска оптимального вектора $\boldsymbol{\theta}$ к поиску оптимального вектора $\boldsymbol{x}$ на симплексе тем:

$$\boldsymbol{x}^* = arg \max_{x \in \Delta} \sum_{w \in d} n_{dw} \ln x_w,$$

где $\Delta = conv(\boldsymbol{\varphi}_1, \dots \boldsymbol{\varphi}_{|T|})$.

Это задача вогнутой максимизации на симплексе, существует много алгоритмов её решения. В частности, алгоритм Франка-Вульфа ищет разреженное приближение решения такой задачи. Применяя его к нашей задаче, получим алгоритм 1.

Алгоритм 1. E-шаг: inference

Вход: профили тем $\boldsymbol{\varphi}_1, \dots \boldsymbol{\varphi}_{|T|}$, документ $\boldsymbol{d}$;
Выход: вектор $\boldsymbol{\theta}^*$, для которого $\sum_{t\in T} \theta_t^* \boldsymbol{\varphi_t} = \boldsymbol{x^*}$ максимизирует $f(\boldsymbol{x}) = \sum_{w\in d} n_{dw} \ln x_w$;
1: Выбираем вершину $\boldsymbol{\varphi_r}$ симплекса $\Delta = conv(\boldsymbol{\varphi}_1, \dots \boldsymbol{\varphi}_{|T|})$ с наибольшим значением функции f: $\boldsymbol{x^0} = \boldsymbol{\varphi}_r$, $\theta_r^0 = 1$, $\theta_k^0 = 0$, $\forall k \neq r$;
2: **для** $l = 1, \dots, \infty$
3: $\quad i' := arg\max_i \boldsymbol{\varphi}_i^T \nabla f(\boldsymbol{x}^l)$;
4: $\quad \alpha' = arg \max_{\alpha\in[0,1]} f(\alpha\, \boldsymbol{\varphi}_{i'} + (1-\alpha)\, \boldsymbol{x}^l)$;
5: $\quad \boldsymbol{x}^{l+1} := \alpha'\, \boldsymbol{\varphi}_{i'} + (1-\alpha')\, \boldsymbol{x}^l$;
6: $\quad \boldsymbol{\theta}^{l+1} := (1-\alpha')\, \boldsymbol{\theta}^l$, $\theta_{i'}^{l+1} := \theta_{i'}^{l+1} + \alpha'$;

Для нашего алгоритма наследуются свойства алгоритма Франка-Вульфа.

1. Алгоритм сходится к оптимальному решению за линейное время.
2. Найденное решение разреженное: после l итераций вектор $\boldsymbol{\theta}$ имеет не более $l+1$ ненулевой компоненты.
3. Можно задавать желаемое соотношение между разреженностью решения, близостью решения к оптимальному и временем работы алгоритма: чем больше итераций совершаем, тем больше время работы, тем ближе решение к оптимальному, и тем менее оно разреженное. (В статье выписана оценка разности значения функции на оптимальном решении и на найденном на очередной итерации).

M-шаг: learning. Теперь по коллекции документов и найденным для каждого документа векторам $\boldsymbol{\theta}_d$ необходимо оценить профили тем. Снова по принципу максимума правдоподобия получаем задачу максимизации, где переменные $\boldsymbol{\varphi}_1, \dots \boldsymbol{\varphi}_{|T|}$ разделяются. Выписывая функцию Лагранжа, приходим к аналитическому решению:

$$\varphi_{wt} \propto \sum_{d\in D} n_{dw}\, \theta_{td}.$$

Таким образом матрица тем Φ – это произведение двух разреженных матриц: матрицы встречаемости слов в документах и матрицы латентных представлений документов Θ.

3. Теоретические свойства модели.

1. E-шаг (решение задачи максимизации аналогично алгоритму Франка-Вульфа) выполняется за линейное время, M-шаг (перемножение двух разреженных матриц) выполняется очень быстро. Таким образом, обучение модели – быстрый алгоритм.
2. Получаемые профили документов и тем разреженные.
3. Можно четко задавать желаемое соотношение между разреженностью решения, его качеством и временем обучения модели.
4. В модели явным образом не предполагается, что $\boldsymbol{\theta}$ имеет априорное распределение. Это могло бы служить причиной переобучения, однако в нашем случае такое априорное распределение все же неявно существует. Итак, в алгоритме поиска профилей документов мы можем явным образом потребовать, чтобы профили содержали не более t ненулевых элементов, т.е. искать $\max_{\theta\in\Delta_1}\{f(\boldsymbol{\theta}) : \|\boldsymbol{\theta}\|_0 \leqslant t\}$. Это эквивалентно введению штрафа в целевую функцию $\lambda \|\boldsymbol{\theta}\|_0$ при некотором значении λ. Таким образом,

$$\boldsymbol{\theta^*} = arg\max_{\theta\in\Delta_1}\{f(\boldsymbol{\theta}) - \lambda\|\boldsymbol{\theta}\|_0\} = arg\max_{\theta\in\Delta_1} P(\boldsymbol{d}\,|\,\boldsymbol{\theta})\, P(\boldsymbol{\theta}) = arg\max_{\theta\in\Delta_1} P(\boldsymbol{\theta}\,|\,\boldsymbol{d}).$$

Итак, наш алгоритм – это максимизация апостериорной вероятности, при неявном предположении о том, что $\boldsymbol{\theta}$ – экспоненциально распределенная случайная величина.

Результаты экспериментов. Эксперименты проводились для 4 разных коллекций (AP, KOS, Grolier, Enron) и для 4 разных алгоритмов (FSTM, PLSA, LDA, STC). 90 процентов было выделено на обучение, 10 процентов – на контроле, подробнее о том, как производилось разбиение, и как модели настраивались на контроле, ничего не сказано. На графиках о скорости работы и разреженности профилей документов отдельно выделяются «learning phase» и «inference phase». Я это понимаю как обучение модели и описание документов контроля соответсвенно - хотя, возможно, это и не так. Термин «document/topic sparsity» вводится как доля ненулевых элементов в матрицах Θ/Φ соответственно. Таким образом, чем *больше* эта величина, тем *менее* разреженны матрицы.

По графикам, приведенным в статье, авторы делают следующие выводы:

1. Разреженность профилей документов и тем у FSTM значительно лучше чем у других моделей. У PLSA и LDA все профили плотные, кроме профилей документов у PLSA на «inference» – авторы объясняют это тем, что при сохранении модели была потеря информации: многие ненулевые элементы ушли в 0.
2. Для FSTM разреженность профилей документов увеличивается при росте числа тем – это логично, так как документ должен относиться к небольшому числу тем, независимо от их общего числа в модели. Так, например, документ из AP в среднем относился только к 2 темам при $|T| = 10$ и только к 3 темам при $|T| = 100$. Разреженность профилей тем также увеличивается при росте числа тем. (Это следует из того, что профили тем – результат произведения матриц, одна из которых – профили документов).
3. Скорость работы у FSTM снова лучше всех. LDA долго обучается и долго описывает документы. Это объясняется 1. тем, что LDA решает NP-трудную задачу, и 2. тем, что при вариационном байесовском выводе используются сложные вычисления (экспонента, гамма и дигамма функции, и т.д.). PLSA быстро обучается, но долго описывает документы. Это объясняется тем, что в обучении модели PLSA нет отдельной стадии описания документов, а адаптация алгоритма обучения к описанию документов с помощью техники folding-in оказывается достаточно медленной.
4. Качество оценивается по критериям BIC, AIC и perplexity. По критериям AIC и BIC модель FSTM оказалась значительно лучше, чем LDA и PLSA. Из того, что модели PLSA и LDA используют больше свободных параметров (у них плотные профили тем), но не достигают значительно лучшего правдоподобия, авторы делают вывод о том, что LDA и PLSA более склонны к переобучению, чем FSTP. При сравнении перплексии лучше всех оказывается PLSA, FSTP сопоставим, LDA значительно хуже. То, что LDA показал себя хуже PLSA, авторы оправдывают 1. тем, что вариационный байесовский вывод не гарантирует, что находит хорошее решение, 2. тем, что в LDA целевая функция – $P(\boldsymbol{\theta}|\boldsymbol{d})$, а не правдоподобие $P(\boldsymbol{d})$, в то время как перплексия в общем-то оценивает правдоподобие.
5. Экспериментально подтвердилась взаимая зависимость разреженности решения, качества модели и времени работы алгоритма.
6. Модель хорошо распараллелилась.