Быстрые алгоритмы неотрицательных матричных разложений

Шадриков А. А.

МГУ, ВМК, ММП

4 октября 2014 г.

Содержание

Постановка задачи

Дано: матрица $V \in \mathsf{R}^{N \times M}$
Найти: матрицы $W \in \mathsf{R}^{N \times T}$ и $H \in \mathsf{R}^{T \times M}$ такие, что

$$D(V, WH) = \sum_{i,j} D\left(V_{ij}, \sum_k W_{ik} H_{kj}\right) \to \min_{W \geqslant 0, H \geqslant 0}$$

В качестве $D(A, B)$ обычно рассматриваются:

- $\|A - B\|_F^2 = \sum\limits_{i,j} (A_{ij} - B_{ij})^2$ — норма Фробениуса.
- $\mathsf{KL}(A||B) = \sum\limits_{i,j} \left(A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij}\right)$ — обобщённая KL-дивергенция.

Особенности задачи

- Отлично от популярных разложений (SVD, LU) ограничением неотрицательности.
- Предполагается, что $T << \min(N, M)$.
- Неоднозначность разложения.
- Много локальных минимумов.

Общий алгоритм

Вход: матрица V, T, # итераций $iter_{\max}$;
Выход: матрицы W и H;

Инициализировать W_{ik}, H_{kj} $\forall i, k, j$;
для всех $iter = 1, \ldots, iter_{\max}$
  $W^{new} = F(W^{old}, H^{old})$;
  $H^{new} = G(W^{old}, H^{old})$;

Примеры итерационных методов

1. **PLSA — Probabilistic Latent Semantic Analysis [Hoffman, 1999]**

$$n_{ikj} = V_{ij} \frac{W_{ik} H_{kj}}{\sum_{s \in T} W_{is} H_{sj}}$$

$$W_{ik} = \frac{n_{ik}}{n_k} \equiv \frac{\sum_j n_{ikj}}{\sum_{ij} n_{ikj}}, \qquad H_{kj} = \frac{n_{kj}}{n_j} \equiv \frac{\sum_i n_{ikj}}{\sum_{ik} n_{ikj}},$$

Краткая запись через знак пропорциональности $\propto$:

$$W_{ik} \propto n_{ik}; \qquad H_{kj} \propto n_{kj};$$

Примеры итерационных методов

2. **MU — Gradient Descent with Multiplicatize Update Rule [Lee, Seung, 2001]**

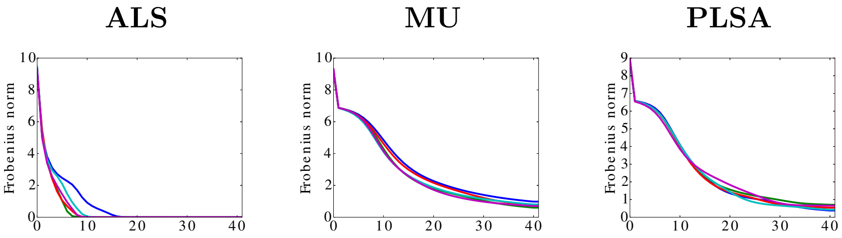
$$W_{ik} = W_{ik} \frac{(VH^T)_{ik}}{(WHH^T)_{ik}}, \qquad H_{kj} = H_{kj} \frac{(W^TV)_{kj}}{(W^TWH)_{kj}}$$

3. **ALS — Alternating Least Squares [Paatero, Tapper, 1994]**

$$W \leftarrow [solve\, HH^TW^T = HV^T]_+$$

$$H \leftarrow [solve\, W^TWH = W^TV]_+$$

Проблемы общей задачи



- Множество локальных минимумов
- Долгая сходимость

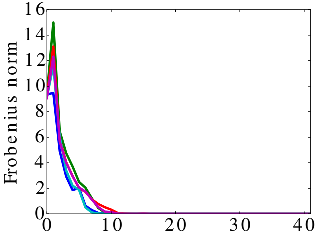
Использование методов для стохастического разложения

Для решения задачи стохастического матричного разложение общие методы также можно использовать следующими способами:

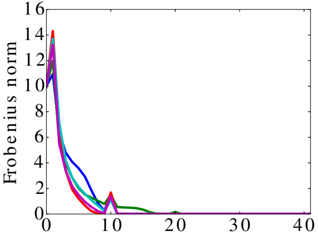
- Нормировка матриц W, H в конце итерации.
- Чередование с методом стохастического матричного разложения (например, PLSA).

Эксперименты с нормировкой

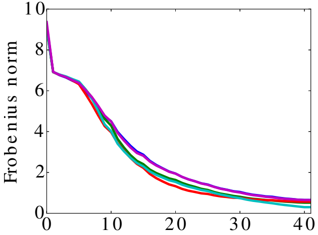
ALS, 5 итерация



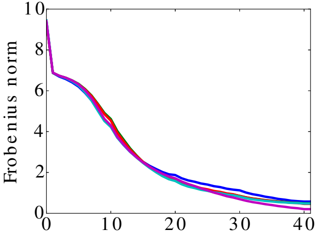
ALS, 10 итерация



MU, 5 итерация

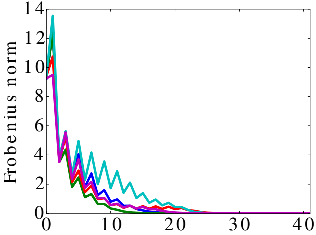


MU, 10 итерация

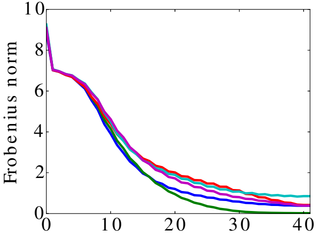


Эксперименты с последовательностью

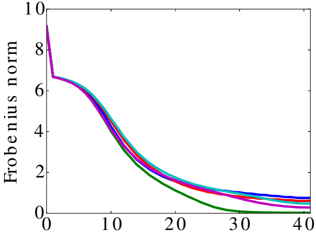
ALS + PLSA



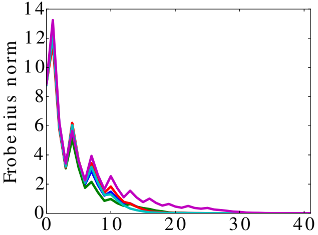
MU + PLSA



ALS + MU



ALS + MU + PLSA



Якорные слова

Понятие «слова» пришло из задачи тематического моделирования, где строки матрицы V интерпретируются как слова, столбцы — документы, а T — число тем.

Якорные слова

- *Пусть в матрице W $\forall$ столбца k $\exists$ строка i такая, что $W_{ik} > 0$, $W_{is} = 0$, $\forall s \neq k$*
- *В матрице W можно выделить диагональную подматрицу.*
- *Для каждой темы существует слово, которое исключительно выделяет данную тему.*

Преимущество якорных слов

Определение

Множество точек из R^T, *удовлетворяющих* $\sum_i x_i = 1, x_i \geqslant 0$ *называется симплексом.*

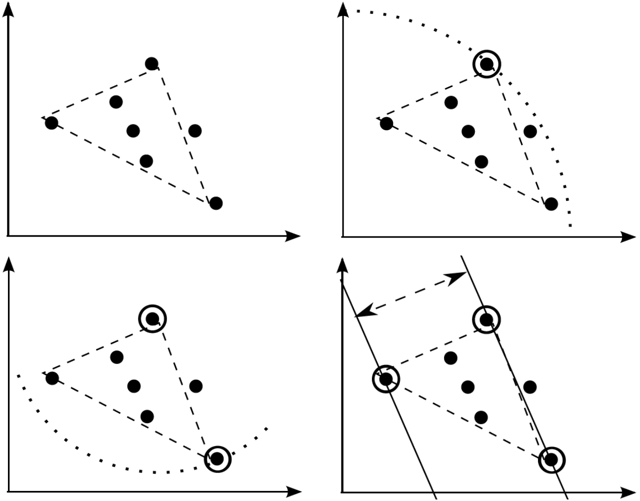
- Задача поиска якорных слов эквивалентна поиску вершин T-мерного симплекса.
- Найдя якорные слова можно легко восстановить матрицы W и H.
- Полученное решение является стационарной точкой EM-алгоритма. [Arora et al., 2013]
- Если матрица V без шума, то полученное разложение единственно. [Donoho, Stodden, 2004]

Предположения об исходных матрицах

- Строки матрицы V должны быть из T-мерного симплекса.
- Каждая вершина T-мерного сиплекса не должна аппроксимироваться линейной оболочки других.
- В «реальной» матрице W выделяется диагональная подматрица.
- Строки «реальной» матрице H линейно независимы.

Идея нахождения якорных слов

Якорные слова фактически задают вершины искомого T-мерного симплекса:



Алгоритм нахождения якорных слов

Вход: N точек $\{d_1, d_2, \ldots, d_N\}$, $\#$ якорных слов T;
Выход: якорные слова S;

$S \leftarrow \{d_i | d_i = \arg\max_i \|d_i\|\}$;
для всех $i = 1, \ldots, T-1$
  Берём d_i наиболее удалённое от $\mathrm{span}(S)$;
  $S \leftarrow S \cup \{d_i\}$;
Переименуем вектора: $S = \{u_1, u_2, \ldots, u_T\}$;
для всех $i = 1, \ldots, T$
  Берём d_i наиболее удалённое от $\mathrm{span}(\{u_1, \ldots, u_T\} \setminus u_i)$;
  Заменяем u_i на d_i;

Возможные изменения алгоритма

- Перед использованием алгоритма уменьшить размерность проецированием на подпространство меньшей размерности.
- Возможно уменьшение T, если на определённом шаге d_i находится близко к $\mathrm{span}(S)$

Восстановление матрицы W

Обозначим за s_k индекс для k-го якорного слова.

Вход: Матрица V, набор якорных слов S
Выход: матрица W

для всех $i = 1, \dots, N$
  Решить оптимизационную задачу
  $C_i = \arg\min_{C_i} D\left(V_i, \sum_{k \in S} C_{i,k} V_{s_k}\right)$, с ограничениями
  $C_{i,k} \geqslant 0$, $\sum_k C_{i,k} = 1$;
$W \leftarrow C$;

Использование

Зададим матрицу Q:

$$Q = \tilde{V}\tilde{V}^T - \bar{V},$$

$$\tilde{V}_j = \frac{V_j}{\sqrt{n_j(n_j-1)}} \qquad \bar{V} = \sum_j \frac{diag(V_j)}{n_j(n_j-1)} \qquad n_j = \sum_i V_{ij}$$

Использование матрицы Q вместо матрицы V делает алгоритм нахождения якорных слов устойчивее. [Arora et al., 2013]

Развитие алгоритма

- Можно ли требовать существование якорных слов не для каждой темы?
- Будет ли в этом случае алгоритм находить хорошее начальное приближение для EM-алгоритма?
- Можно ли найти приближение якорных слов путём просмотра коллекции?
- В алгоритме поиска якорных слов мы просматриваем все строки матрицы V. Как масштабировать алгоритм, чтобы время работы росло не квадратично по V?

Список литературы

- *Arora S., Ge R., Halpern Y., Mimno D., Moitra A., Sontag D., Wu Y., Zhu M.* A Practical Algorithm for Topic Modeling with Provable Guarantees.
- *Asuncion A., Welling M., Smyth P., Teh Y. W.* On smoothing and inference for topic models. Int'l Conf. on Uncertainty in Artificial Intelligence, 2009.
- *Donoho D., Stodden V.* When Does Non-Negative Matrix Factorization Give a Correct Decomposition into Parts?
- *Lee D., Seung S.* Algorithms fot Nonnegative Matrix Factorization.
- *Hofmann T.* Probabilistic Latent Semantic Indexing.

Спасибо за внимание!

http://bitbucket.org/vuvko/nmf/