

Исследование рандомизированных методов факторизации матриц в задачах машинного обучения

Федоров Артем Максимович
Анохин Андрей Владимирович
Певцов Артём Алексеевич

GitHub проекта

AIMasters

15 декабря 2024 г.

В задачах машинного обучения отдельное место уделено алгоритмам на базе KNN, характерные проблемы которых:

- Требуется работа с большим объемом данных.
- Большая размерность признакового пространства приводит к неадекватной работе метрик (Проклятие размерности)
- Использование нелинейных методов (**Umap**, **T-SNE**, **NCA**) приводит к потере свойств корреляции признаков.

Самым частым методом понижения размерности так или иначе является **SVD** (или же PCA) над матрицей признаков: автоматический подбор степени сжатия и допуска ошибки, приведение к новому представлению приходящих объектов.

Применение KNN в задачах понижения размерности

Пусть: $X \in \mathbb{R}^{N \times D}$ выборка объектов $x \in \mathbb{R}^D$ объема N .

Требуется решить задачу минимизации: $\tilde{X} \in \mathbb{R}^{N \times d}, W \in \mathbb{R}^{d \times D}, d \ll D$

$$\|X - \tilde{X}^T W\|_{fro}^2 \rightarrow \min_{W, \tilde{X}}$$

Решением такой задачи в аналитическом виде будет являться **SVD** разложение с обрезанным спектром:

$$X = U \Sigma V^T \Rightarrow \tilde{X} = U, W = \Sigma V^T$$

Применение честного **SVD** является не дешевой операцией: большие вычислительные ресурсы, проблемы с работой над разреженными или непомещающимися в оперативную память матрицами.

Решение \Rightarrow переход к неточным вычислениям, рандомизированным.

- 1 Рандомизированный **SVD**.
- 2 Sparse-Aware **Sparse-SVD**.
- 3 **Итеративный PCA** через байесовский вариационный вывод.

Группа ставит перед собой задачу сравнить основные подходы трех категорий, получить представления о сильных и слабых сторонах алгоритмов на основе замеров качества для модельной задачи на KNN:

- 1 Время работы алгоритма понижения размерности
- 2 Время работы алгоритма
- 3 Качество работы алгоритма относительно честного KNN

Рандомизированный SVD

Будем считать матрицу признаков X как некоторый линейный оператор. Тогда будем считать, что размерность минимального подпространства содержащего image меньше его реального пространства отображения. Тогда имеем идею для решения задачи:

Data: X

Result: U, Σ, V^T

$\Omega \sim \mathcal{N}(0, I)$ # generation range sampler

$Y \leftarrow X \cdot \Omega$

$Q, R \leftarrow qr(Y)$ # randomized range finder

$U, \Sigma, V^T \leftarrow SVD(R)$

Return $Q \cdot U, \Sigma, V^T$

Algorithm 1: Base Random SVD

Рандомизированный SVD

Данный алгоритм имеет проблемы с определением числа сэмплов, требуемых для восстановления image. В нашем исследовании мы решили эту проблему используя алгоритм

Given an $m \times n$ matrix \mathbf{A} , a tolerance ε , and an integer r (e.g., $r = 10$), the following scheme computes an orthonormal matrix \mathbf{Q} such that (4.2) holds with probability at least $1 - \min\{m, n\}10^{-r}$.

```
1  Draw standard Gaussian vectors  $\omega^{(1)}, \dots, \omega^{(r)}$  of length  $n$ .
2  For  $i = 1, 2, \dots, r$ , compute  $\mathbf{y}^{(i)} = \mathbf{A}\omega^{(i)}$ .
3   $j = 0$ .
4   $\mathbf{Q}^{(0)} = [\ ]$ , the  $m \times 0$  empty matrix.
5  while  $\max\{\|\mathbf{y}^{(j+1)}\|, \|\mathbf{y}^{(j+2)}\|, \dots, \|\mathbf{y}^{(j+r)}\|\} > \varepsilon/(10\sqrt{2/\pi})$ ,
6       $j = j + 1$ .
7      Overwrite  $\mathbf{y}^{(j)}$  by  $(\mathbf{I} - \mathbf{Q}^{(j-1)}(\mathbf{Q}^{(j-1)})^*)\mathbf{y}^{(j)}$ .
8       $\mathbf{q}^{(j)} = \mathbf{y}^{(j)} / \|\mathbf{y}^{(j)}\|$ .
9       $\mathbf{Q}^{(j)} = [\mathbf{Q}^{(j-1)} \ \mathbf{q}^{(j)}]$ .
10     Draw a standard Gaussian vector  $\omega^{(j+r)}$  of length  $n$ .
11      $\mathbf{y}^{(j+r)} = (\mathbf{I} - \mathbf{Q}^{(j)}(\mathbf{Q}^{(j)})^*)\mathbf{A}\omega^{(j+r)}$ .
12     for  $i = (j + 1), (j + 2), \dots, (j + r - 1)$ ,
13         Overwrite  $\mathbf{y}^{(i)}$  by  $\mathbf{y}^{(i)} - \mathbf{q}^{(j)}\langle \mathbf{q}^{(j)}, \mathbf{y}^{(i)} \rangle$ .
14     end for
15 end while
16  $\mathbf{Q} = \mathbf{Q}^{(j)}$ .
```

Рис.: Алгоритм автоматического поиска достаточно хорошего image

Рандомизированный Sparse-PCA

В качестве алгоритма для Randomized Sparse-PCA рассматривался алгоритм ALS с регуляризатором ℓ_0 . Нужно заметить, что в данной задаче размерность

Data: X

Result: U, V^T

$U_0 \sim \mathcal{N}(0, I), V_0^T \sim \mathcal{N}(0, I)$

generation range sampler

for $k \leftarrow 1$ **to** max_iter **do**

$U_k \leftarrow \arg \min_U \|X - UV_{k-1}^T\|_{fro}^2 + \|U\|_1$
 $V_k \leftarrow \arg \min_V \|X - UV_{k-1}^T\|_{fro}^2 + \|V\|_1$

end

Algorithm 2: Sparse Randomized PCA

Итеративный РСА

Будем считать, что $X \sim \mathcal{N}(\mu + Wt, \sigma^2 I)$; $t \sim \mathcal{N}(0, I)$ - латентные переменные \rightarrow задача вариационного вывода подобрать такие латентные переменные t , что позволили максимизировать оценку неполного правдоподобия

$$\log p(X|\theta) = \mathbb{E}_Z \log \left(\frac{p(X, t | \theta)}{p(t | \theta)} \right) + KL(q(t | \theta) \| p(t | X, \theta))$$

Вывод производится на с помощью ЕМ алгоритма.

Заметим, что в данной версии размерность пространства t не является латентной переменной, а потому число компонент предопределено и является гиперпараметром.

Эксперимент №1

Рассмотрим применение трех алгоритмов на датасете, одновременно содержащем большое число объектов большой размерности, каждый из которых является sparse матрицей.

Будем решать задачу **binary-classification** над фотографиями людей – пришедший для распознавания снимок принадлежит числу людей, которых мы пропускаем на вход в здание или нет. Пусть:

- (X, Y) - обучающая выборка
 - $X \in \mathbb{R}^{N,D}$ – набор снимков обучающей выборки
 - $Y \in \mathbb{R}^N$ – класс (является персоналом или нет)
- $U \subset Y/\sim$ подмножество всех людей, которых мы пропускаем
- Требуется максимизировать эмпирический риск:

$$\mathcal{R} = \sum_{i=1}^{\tilde{N}} [a(\tilde{X}_i) = \tilde{Y}_i] \longrightarrow \max$$

Данные эксперимента

В качестве данных использовалась выборка из фотографий людей датасета [Kaggle face-recognition-dataset]. Случайным образом из $|Y/\sim| = 1680$ классов отобрано $|U| = 50$, изображения $x_i \in \mathbb{R}^{128 \times 128}$ переведены в чернобелый цвет и нормализованы. После чего проводились замеры в следующем порядке:

- 1 Получен лучший результат для KNN без факторизации признаков X .
- 2 Для каждого метода факторизации выполняется PCA и оставляются лишь сингулярные числа, в сумме дающие 70% от ℓ_1 нормы Σ .
- 3 Score есть отношение ROC-AUC исследуемого алгоритма и BaseKNN (Больше – хуже)
- 4 Оптимизация K для KNN со взятием наилучшего запуска по отложенной выборке по обозначенной оценке эмпирического риска \mathcal{R} .

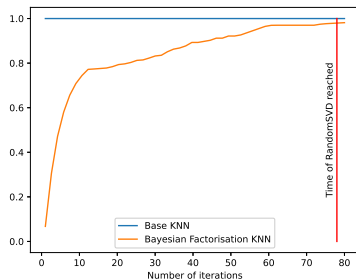


Рис.: Пример объектов датасета

Результаты

Эксперимент	time SVD	time KNN	$ \tilde{\Sigma} $	K	Score
ClassicSVD	22.1 сек	1.8 сек	81	4	0.98
RandomizedSVD	4.8 сек	2.2 сек	124	4	0.94
SparseSVD	28.3 сек	11.7 сек	423	4	0.94
BayesianSVD	4.1 сек	1.8 сек	100	5	0.95

Таблица: Результаты лучших запусков



- (a) Зависимость относительного качества от числа итерация для Randomized SVD, снизу байесовского PCA
- (b) Пример базиса: сверху SparseSVD

- 1 Рандомизированные методы SVD способны при сравнительно малом трейдофе качества получить серьезное улучшение по скорости как обучения так и инференса
- 2 Sparse SVD показало себя наименее удачно – долгое время работы и самое плохое уменьшение размерности.
- 3 Sparse SVD эмпирически сильно отстает от других методов по качеству получаемого \tilde{X} . Получаемые изображения обладают меньшей общностью (явны выделенные лица, а не отдельные черты) \Rightarrow размерность моделируемого пространства сильно больше.
- 4 Байесовский PCA не только является самым быстрым, но и получил лучшее качество среди рандомизированных моделей. Вместе с этим от стремительно выходит на плато качества.

Эксперимент №2

Рассмотрим работу алгоритмов на Больших Данных, с которыми алгоритмы без предварительной факторизации не справляются.

Будем решать задачу **LDA** – построения тематической модели, что сможет автоматически выделять тематики в некотором корпусе текстов. Осложним задачу и будем рассматривать огромный корпус текстов (≈ 150000 документов) аннотацией статей на английском языке из лаборатории Семантического Анализа Текстов МГУ.

- X - корпус текстов представленных в виде мешка слов
- T – множество тем (мы считаем их число равным 30)
- Каждый текст – аннотация или первый абзац новости из телеграмма или интернета.
- Требуется максимизировать логарифм неполного правдоподобия

Модель LDA

В данном эксперименте втроенные LDA в пакет Sklearn не позволяют добиться адекватного времени исполнения. Для решения данной проблемы был использован hard EM, где на E-step распределение выбирается из семейства дельта функций:

Параметры модели:

- $\Phi = \{\phi_{tw}\} \in \mathbb{R}^{T \times V}$ – распределение слов по темам
- $\pi \in \mathbb{R}^T$ вектор распределения тем в корпусе документов

Совместное распределение:

- $p(W, t \mid \Phi, \pi) = p(W \mid t, \Phi)p(t \mid \pi) = \prod_{d=1}^D \prod_{n=1}^{N_d} \phi_{t_d w_{dn}} \pi_{t_d}$

ЕМ-алгоритм:

- **Е-шаг:** $KL(q(t) \parallel p(t \mid \Phi, \pi)) \rightarrow \min_{q(t) \in \{\delta\}}$
- **М-шаг:** $\mathbb{E}_{q(t)} \log p(W, t \mid \Phi, \pi) \rightarrow \max_{\Phi, \pi}$

Постановка задачи

Так как LDA не способен восстанавливать темы с некоторым линейизованным порядком, мы будем сравнивать лишь распределения вероятностей, что данный документ был сгенерирован из распределения конкретной темы (при этом определена суперетрика – минимум по всем паросочетаниям)

- 1 С процедуры мультистарт 20 раз запускаем параллельно LDA на изначальной матрице и запоминаем как гарант.
- 2 Для каждого алгоритма проводим 10 запусков, выбираем модель с наибольшей оценкой правдоподобия, и считаем среднюю KL дивергенцию по каждому документу по каждому из 20 распределений гаранта. **(Меньше – лучше)**

$$Score = \frac{1}{N} \sum_i \frac{\sum_j KL(q_{\max}(t_i) \| q_{garant}^j(t_i))_{\min}}{\mathbb{D}KL(q_{\max}(t_i) \| q_{garant}(t_i))_{\min}}$$

Результаты

Эксперимент	time SVD	time LDA	$ \tilde{\Sigma} $	Score
ClassicSVD	9 мин	16 сек	2433	0.63
RandomizedSVD	126 сек	16 сек	2675	1.29
SparseSVD	4 мин	25 сек	5226	3.97
BayesianSVD	58 сек	16 сек	2600	1.61

Таблица: Результаты лучших запусков

Topic	Word #1	Word #2	Word #3	Word #4	Word #5
Topic #1	john	love	father	wife	family
Topic #2	police	joe	kill	gang	killed
Topic #3	friend	go	tell	get	father
Topic #4	find	house	one	death	body
Topic #5	war	men	one	army	soldie
Topic #6	love	get	father	family	life
Topic #7	film	new	school	life	play

Таблица: Самые весомые слова нескольких тем

- 1 Видно, что рандомизированные методы очень хорошо справляются с оптимизацией решения задачи. Приведенная метрика через дивергенции крайне неустойчива, и минимальное отклонение уже сильно изменяет показатели оценки.
- 2 Каждый из методов дал серьезный прирост к скорости работы программы, так как изначальный вариант Sklearn выполнялся на протяжении 40 минут и более.
- 3 Прямой взгляд на слова, характеризующие тематики, не позволил выявить серьезные различия между алгоритмами факторизации. Единственное исключение есть Sparse SVD, для которого в топе слов не наблюдалось в принципе слов общей лексики (что все же присутствовали в топе каждой темы других методов).

Над проектом работали

❶ Анохин Андрей Владимирович –

- Работа над рандомизированным SVD
- Дизайн пайплайна KNN
- Оформление экспериментов по KNN

❷ Певцов Артём Алексеевич –

- Работа со спарсифицированной версией randomised SVD
- Дизайн задачи KNN, поиск датасетов, вся предобработка.
- Автоматизация пайплайна файнтюна моделей (optuna, логирование экспериментов).

❸ Федоров Артем Максимович –

- Байесовские модели (Bayesian SVD, LDA)
- Задача на LDA
- Презентация

- [Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions](#) – наше все, позволил в принципе придумать тему для проекта. (Спасибо Рахуба Максим Владимирович)
- [Spark-RSVD](#) – библиотека, имплементирующая наш Sparse подход в том числе и нативно на питоне.
- [Randomized methods for computing the Singular Value Decomposition \(SVD\) of very large matrices](#) – обзор и идеи для методов факторизации больших данных.
- [Байесовские методы Ветрова Дмитрия Петровича](#) – все связанное с Байесовскими методами

GitHub проекта