Исследование рандомизированых методов факторизации матриц в задачах машинного обучения

Федоров Артем Максимович
Анохин Андрей Владимирович
Певцов Артём Алексеевич

GitHub проекта

AIMasters

16 декабря 2024 г.

Введение

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 PCA 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: $\tilde{X} \in \mathbb{R}^{N \times d}, W \in \mathbb{R}^{dD}$, $\mathrm{d} \ll D$

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

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

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

Решаемая проблема

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

Решение ⇒ переход к неточным вычислениям, рандомизированным.

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

Постановка задачи

Группа ставит перед собой задачу сравнить основные подходу 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

Рандомизированый SVD

Будем считать матрицу признаков X как некоторый линейный оператор. Тогда будем считать, что размерность минимального подпространства содержащего image меньше его реального пространсвта 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

Рандомизированый SVD

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

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

Draw standard Gaussian vectors $\boldsymbol{\omega}^{(1)}, \ldots, \boldsymbol{\omega}^{(r)}$ of length n.
For $i = 1, 2, \ldots, r$, compute $\boldsymbol{y}^{(i)} = \boldsymbol{A}\boldsymbol{\omega}^{(i)}$.
$j = 0$.
$\boldsymbol{Q}^{(0)} = [\ ]$, the $m \times 0$ empty matrix.
while $\max\left\{\left\|\boldsymbol{y}^{(j+1)}\right\|, \left\|\boldsymbol{y}^{(j+2)}\right\|, \ldots, \left\|\boldsymbol{y}^{(j+r)}\right\|\right\} > \varepsilon/(10\sqrt{2/\pi})$,
    $j = j + 1$.
    Overwrite $\boldsymbol{y}^{(j)}$ by $\left(\mathbf{I} - \boldsymbol{Q}^{(j-1)}(\boldsymbol{Q}^{(j-1)})^*\right)\boldsymbol{y}^{(j)}$.
    $\boldsymbol{q}^{(j)} = \boldsymbol{y}^{(j)} / \left\|\boldsymbol{y}^{(j)}\right\|$.
    $\boldsymbol{Q}^{(j)} = [\boldsymbol{Q}^{(j-1)}\ \boldsymbol{q}^{(j)}]$.
    Draw a standard Gaussian vector $\boldsymbol{\omega}^{(j+r)}$ of length n.
    $\boldsymbol{y}^{(j+r)} = \left(\mathbf{I} - \boldsymbol{Q}^{(j)}(\boldsymbol{Q}^{(j)})^*\right)\boldsymbol{A}\boldsymbol{\omega}^{(j+r)}$.
    for $i = (j+1), (j+2), \ldots, (j+r-1)$,
        Overwrite $\boldsymbol{y}^{(i)}$ by $\boldsymbol{y}^{(i)} - \boldsymbol{q}^{(j)}\left\langle \boldsymbol{q}^{(j)},\ \boldsymbol{y}^{(i)}\right\rangle$.
    end for
end while
$\boldsymbol{Q} = \boldsymbol{Q}^{(j)}$.

Рис.: Алгоритм автоматического поиска достаточно хорошего image

Рандомизированый 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 PCA

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

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

Вывод производится на с помощью EM 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 впускаем
- Требуется максимизировать эмпирический риск:

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

Данные эксперимента

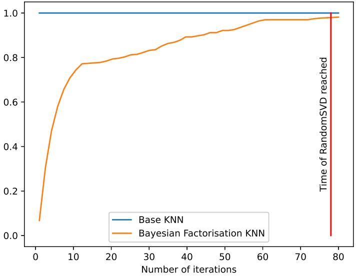
В качестве данных использовалась выборка из фотографий людей датасета [Kaggle face-recognition-dataset]. Случайным образом из $|Y/\sim| = 1680$ классов отобрано $|U| = 50$, изображения $x_i \in \mathbb{R}^{128\times128}$ переведены в чернобелый цвет и нормализованы. После чего проводились замеры в следующем 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 обозначеной оценке эмпирического риска $\mathcal{R}$.

Рис.: Пример объектов датасета

Эксперимент	time SVD	time KNN	$\lvert\tilde{\Sigma}\rvert$	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 байесовского PCA

(b) Пример базиса: сверху Randomized SVD, снизу SparseSVD

Выводы

1. Рандомизированные методы SVD способны при сравнительно малом трейдофе качества полчить 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 ($\approx$ 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 решении 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:

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

EM-алгоритм:

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

Постановка задачи

Так как LDA не способен восстанавливать темы с некоторым линеризованным 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)**

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

Результаты

Эксперимент	time SVD	time LDA	$\lvert\tilde{\Sigma}\rvert$	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

Выводы

1. Видно, что рандомизированные методы очень хорошо справляются с оптимзиацией 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).

Работа в команде

Над проектом работали

1. **Анохин Андрей Владимирович** –
   - Работа над рандомизированным SVD
   - Дизайн пайплайна KNN
   - Оформление экспериментов по KNN
2. **Певцов Артём Алексеевич** –
   - Работа со спарсифицированной версией randomised SVD
   - Дизайн задачи KNN, поиск датасетов, вся предобработка.
   - Автоматизация пайплайна файнтюна моделей (optuna, логирование экспериментов).
3. **Федоров Артем Максимович** –
   - Байесовские модели (Bayesian SVD, LDA)
   - Задача на LDA
   - Презентация

Ссылки

- 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