

OzonMasters–NLA Course

Рандомизированное SVD разложение и его приложения

Салахов Дамир, Шубочкин Евгений

Москва – 2021

- 1 Наивное описание метода
- 2 Мотивация
- 3 Результаты статьи и наши надежды
- 4 Наши результаты имплементации

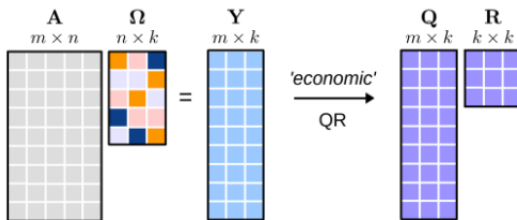
Простыми словами о сложном

Пусть мы имеем матрицу A размера $m \times n$.

Выберем целевой ранг k такой, что $k < \min(m, n)$.

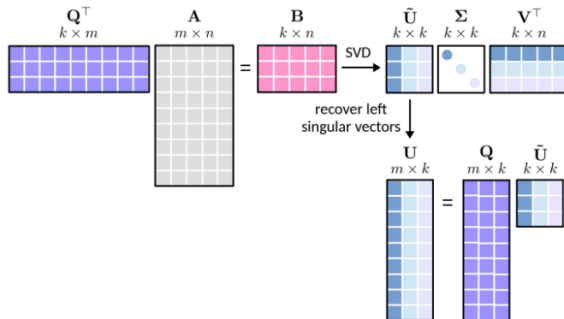
Тогда для реализации RSVD нужно будет:

- 1) сгенерировать гауссовскую случайную матрицу Ω размером $n \times k$;
- 2) вычислить новую матрицу Y размером $m \times k$;
- 3) применить QR -разложение к матрице Y .



Простыми словами о сложном

- 4) получить матрицу B размером $k \times n$ путем умножения транспонированной матрицы Q на матрицу A ;
 - 5) вычислить SVD для матрицы B .
- Здесь, вместо вычисления SVD исходной матрицы A , нужно вычислять SVD B , которая представляет собой меньшую матрицу, что дает преимущества асимптотически.



Почему нас это заинтересовало?

Замечание:

- 1 Поскольку сингулярные значения (т.е. Σ) и правые сингулярные векторы (т.е. V) матрицы B также "являются" сингулярными значениями и правыми сингулярными векторами исходной матрицы A , поэтому мы должны сохранить сингулярные значения и правые сингулярные векторы матрицы B .
- 2 Матрицу U , если необходимо, можно получить, как произведение матриц Q и U^\sim .

Наивная оценка сложности:

- 1 Для вычисления SVD матрицы размера $m \times n$ необходимо время, асимптотически равное $O(m^2n)$;
- 2 для вычисления Randomized SVD матрицы размера $m \times n$ необходимо время, асимптотически равное $O(mnk)$.

Рекомендательные системы и с чем их едят

Определение: Рекомендательные системы — программы, которые пытаются предсказать, какие объекты будут интересны пользователю, имея определенную информацию о его профиле.

Проблемы:

- 1 матрица предпочтений огромна;
- 2 матрица предпочтений очень разрежена.

Подход из статьи

Пусть X матрица размера $m \times n$ ($m \leq n$);

μ – вектор размерности m в пространстве векторов-столбцов X .

Хотим получить приближение ранга k сингулярного разложения матрицы $X' = X - \mu \mathbf{1}^T$ без явного формирования X' .

Различия между расширенным алгоритмом и исходным находятся в строках 6, 9, 10 и 12, если кратко.

Algorithm 1 The rank- k singular value decomposition of the $m \times n$ matrix $X - \mu \mathbf{1}^T = U \Sigma V^T$ with ($m \leq n$) using the sampling parameter K ($k < K \ll m$) and $q \in \{0, 1, 2, \dots\}$.

```

1: procedure SHIFTED-RANDOMIZED-SVD( $X, \mu, k, K, q$ )
2:   Draw an  $n \times K$  standard Gaussian matrix  $\Omega$ 
3:   Form the sample matrix  $X_1 \leftarrow X\Omega$ 
4:   Compute the QR factorization  $X_1 = Q_1 R_1$ 
5:   if  $\mu \neq 0$  then
6:     Compute  $QR = Q_1 R_1 - \mu \mathbf{1}^T$  using the QR-update algorithm
7:   end if
8:   for  $i = 1, 2, \dots, q$  do
9:     Compute the QR-factorization  $Q' R' = X^T Q - \mathbf{1}(\mu^T Q)$ 
10:    Compute the QR-factorization  $Q R = X Q' - \mu(\mathbf{1}^T Q')$ 
11:  end for
12:  Form  $Y \leftarrow Q^T X - (Q^T \mu) \mathbf{1}^T$ 
13:  Compute the singular value decomposition of  $Y = U_1 \Sigma V^T$ 
14:   $U \leftarrow Q U_1$ 
15:  return ( $U, \Sigma, V$ )
16: end procedure

```

Подход из статьи

Алгоритм SHIFTED-RANDOMIZED-SVD состоит из трех этапов:

- 1 оценить QR матрицы для X' ;
- 2 спроецировать X' на пространство базисной матрицы Q ;
- 3 вычислить SVD полученной проекции.

Подход из статьи

Почему вообще данные разложения работают хорошо, и, действительно, дают правильный ответ?
Результат для RandomizedSVD:

Theorem 1 (Halko et al. (2011)) Let \mathbf{A} be an $m \times n$ matrix, $k \geq 1$ an integer, and choose an oversampling parameter $p \geq 4$. If $\Omega \in \mathbb{R}^{n \times (k+p)}$ is a standard Gaussian random matrix and $\mathbf{QR} = \mathbf{A}\Omega$ is the economized QR decomposition of $\mathbf{A}\Omega$, then for all $u, t \geq 1$,

$$\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^* \mathbf{A}\|_F \leq \left(1 + t \sqrt{\frac{3k}{p+1}}\right) \sqrt{\sum_{j=k+1}^n \sigma_j^2(\mathbf{A})} + ut \frac{\sqrt{k+p}}{p+1} \sigma_{k+1}(\mathbf{A}), \quad (1)$$

with failure probability at most $2t^{-p} + e^{-u^2}$.

Результат для ShiftedRandomizedSVD:

$$\mathbb{E}[\|\bar{\mathbf{X}} - \mathbf{U}\mathbf{S}\mathbf{V}^T\|] \leq \left[1 + 4\sqrt{\frac{2m}{k-1}}\right]^{\frac{1}{2q+1}} \sigma_{k+1} \quad (12)$$

where σ_{k+1} is the $(k+1)$ th singular value of the $m \times n$ matrix $\bar{\mathbf{X}}$ with $m \leq n$, $2 \leq k \leq \frac{m}{2}$ is the decomposition rank, and $q \in \mathbb{Z}^+$ is a power value as explained in Algorithm 1.

Надежды

Для матрицы X' размера $m \times n$ вычислительная сложность исходного рандомизированного алгоритма SVD составляет: $O(\alpha k + (m + n)k^2)$, где α - стоимость умножения матрицы на вектор с входной матрицей X' .

- 1 если X' плотная матрица, то $\alpha = mn$;
- 2 если X' разреженная матрица, то $\alpha = T$, где T – малая постоянная величина.

Надежды

Если входная матрица X является плотной, то алгоритмы отработают одинаково по сложности.

НО! если входная матрица X является разреженной, то X' является плотной для любого $\mu \neq 0$. Тогда

- ❶ сложность алгоритма **SHIFTED-RANDOMIZED-SVD**: $O(Tk + m^2 + (m + n)k^2)$, где T - это стоимость умножения разреженной матрицы на вектор, а параметр m^2 связан со сложностью этапа обновления QR ;
- ❷ сложность **RANDOMIZED-SVD**: $O(mnk + (m + n)k^2)$.

Описание выбранной модели

Как я и обещал, мы сделаем систему рекомендаций. И мы сделаем совместную фильтрацию, используя технику SVD (Singular Vector Decomposition); это на ступеньку выше базовой системы рекомендаций, основанной на содержании.

Мы будем использовать знаменитый набор данных Movielens для создания нашей системы рекомендаций.

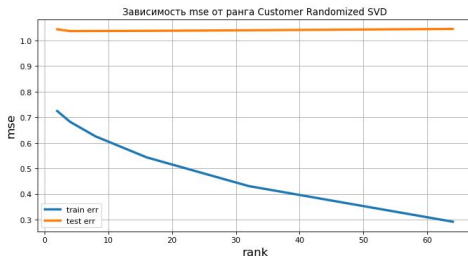
Итоги нового подхода

На данном датасете и данной задаче, выбранный метод, действительно показывает себя очень хорошо!

```
Randomized SVD: k = 60, CPU times: user 914 ms, sys: 54.1 ms, total: 968 ms, Wall time: 505 ms;  
  error_matrix_rank.mean = 0.006463511869165603  
Shifted Randomized SVD: k = 60, CPU times: user 759 ms, sys: 50.5 ms, total: 810 ms, Wall time: 421 ms;  
  error_matrix_rank.mean = 0.006608241821229209  
scipy.sparse.linalg.svds: k = 60, CPU times: user 1.85 s, sys: 17.4 ms, total: 1.86 s, Wall time: 947 ms;  
  error_matrix_rank.mean = 0.006456533551669767
```

Валидация рекомендательной системы

Ниже приведен график зависимости mse предсказанных рейтингов фильмов в зависимости от ранга в SVD



Спасибо за Ваше внимание!