Принцип максимального объема в крестовых аппроксимациях матриц

Команда «MatrixMasters»

Подготовили:
1) Середа Константин
2) Николаев Олег
3) Гавриш Борис
4) Матевосова Анастасия

Москва, 2023

Постановка задачи

Во многих практических задачах для оптимизации хранения матриц и вычислений с ними находят аппроксимации. Хорошие приближения по некоторым нормам дают SVD, QR, которые, однако, требуют слишком большого количества операций. Гораздо более быстрым способом является использование скелетного разложения:

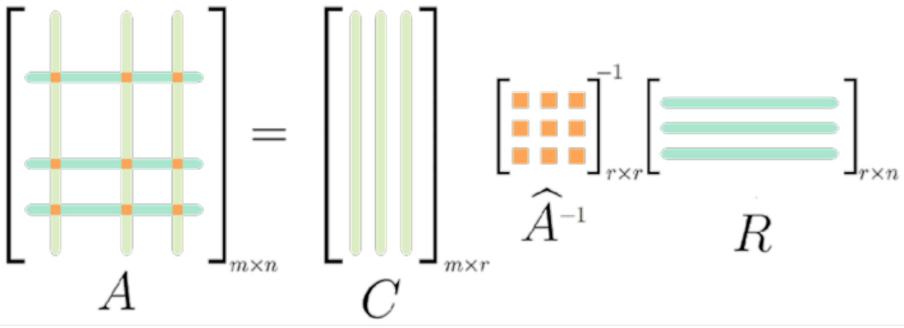
Крестовой аппроксимацией матрицы $A \in \mathbb{C}^{n\times n}$ называют

$$A \approx C\hat{A}^{-1}R, \tag{1}$$

где

$$C = A(:,\mathcal{J}) \in \mathbb{C}^{n\times r}, \qquad R = A(\mathcal{I},:) \in \mathbb{C}^{r\times n}, \qquad \hat{A} = A(\mathcal{I},\mathcal{J}) \in \mathbb{C}^{r\times r}.$$

Множества индексов строк и столбцов $\mathcal{I}$, $\mathcal{J} \subset \{1,\dots,n\}$ имеют размер $|\mathcal{I}| = |\mathcal{J}| = r$. Матрица C образована столбцами A, номера которых входят в $\mathcal{J}$, аналогично строки R образованы на строках $\mathcal{I}$. Матрица $\hat{A}$ формируется как подматрица на пересечении столбцов $\mathcal{J}$ и строк $\mathcal{I}$. Формула (1) задаёт аппроксимацию ранга r исходной матрицы.



Но как осуществить выбор столбцов и строк, которые дадут наилучшую аппроксимацию?

Задача: изучить метод крестовой аппроксимации maxvol и его модификации.
Гипотеза: вариации метода работают быстрее и дают примерно то же качество, что и стандартные разложения (SVD, QR) для специальных матриц.
Оценка качества: измерение времени работы и Чебышевской нормы приращения.

Adaptive Cross Approximation (ACA)

Целью адаптивной крестовой аппроксимации (ACA) является построение малоранговой аппроксимации матрицы *A* непосредственно на основе информации, предоставляемой правильно выбранными строками и столбцами *A*.

Algorithm ACA with full pivoting

1: Set $R_0 := A$, $r := \{\}$, $c := \{\}$, $k := 0$
2: **repeat**
3: $\quad k := k+1$
4: $\quad (i^*, j^*) := \arg\max_{i,j} |R_{k-1}(i,j)|$
5: $\quad r := r \cup \{i^*\}$, $c := c \cup \{j^*\}$
6: $\quad \delta_k := R_{k-1}(i^*, j^*)$
7: $\quad u_k := R_{k-1}(:, j^*)$, $v_k := R_{k-1}(i^*, :)^T / \delta_k$
8: $\quad R_k := R_{k-1} - u_k v_k^T$
9: **until** $\|R_k\|_F \leq \varepsilon \|A\|_F$

Основной минус реализации ACA with full pivoting – для точной проверки критерия остановки непосредственное вычисление нормы $||A||_F$ сложно и требует *O(mn)* операций.

В ходе работы алгоритма строится r одноранговых крестовых аппроксимаций (*скелетонов*), основанных на поиске максимального элемента в матрице. Итоговое приближение $\widetilde{A}_r$ для *A* представляется в виде их суммы:

$$\widetilde{A}_r = \sum_{i=1}^{r} u_i v_i^T \qquad u_i = \frac{C_i}{\sqrt{|\widehat{A}_i|}}, \quad v_i = \frac{R_i\sqrt{|\widehat{A}_i|}}{\widehat{A}_i}, \quad U_k = [u_1\ u_2\ \dots\ u_k], \quad V_k = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{bmatrix}.$$

Преимущества метода:

1. Матричный крестовый метод является быстрым методом аппроксимации матриц матрицами малого ранга.
2. Приближение, находимое данным методом, является квазиоптимальным, т.е. близким к оптимальному r-ранговому приближению.
3. Не использует все элементы матрицы для нахождения приближения.

Adaptive Cross Approximation (ACA)

Algorithm ACA with partial pivoting

Set $R_0 := A,\ r := \{\},\ c := \{\},\ k := 1,\ i^* := 1$
repeat
  $j^* := \arg\max_j |R_{k-1}(i^*, j)|$
  $\delta_k := R_{k-1}(i^*, j^*)$
  if $\delta_k = 0$ **then**
    if $\#r = \min\{m,n\} - 1$ **then**
      Stop
    end if
  else
    $u_k := R_{k-1}(:, j^*),\ v_k := R_{k-1}(i^*, :)^T / \delta_k$
    $R_k := R_{k-1} - u_k v_k^T$
    $k := k + 1$
  end if
  $r := r \cup \{i^*\},\ c := c \cup \{j^*\}$
  $i^* := \arg\max_{i, i \notin r} u_k(i)$
until stopping criterion is satisfied

Чтобы исправить недостатки используется модифицированный алгоритм. В нем заменяется рассмотрение $||A||_F$ на $||A_k||_F$:

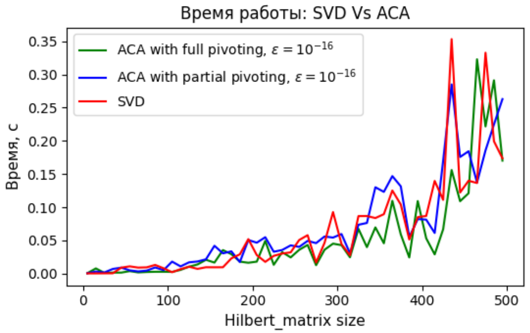
$$A_k = \sum_{j=1}^{k} u_j v_j^T : \|A_k\|_F^2 = \|A_{k-1}\|_F^2 + \sum_{j=1}^{k-1} u_k^T u_j v_j^T v_k + \|u_k\|_2^2 \|v_k\|_2^2,$$

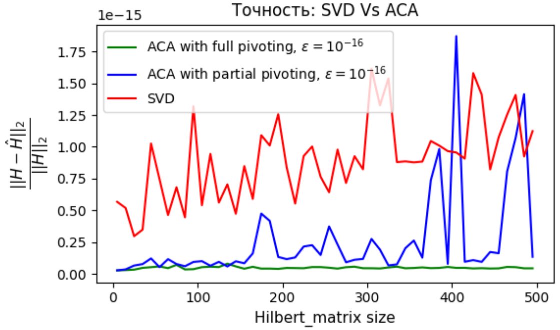
Тогда критерий остановки перепишется следующим образом:

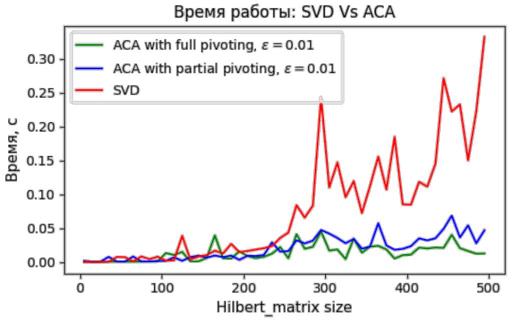
$$\|u_k\|_2 \|v_k\|_2 \leq \epsilon \|A_k\|_F$$

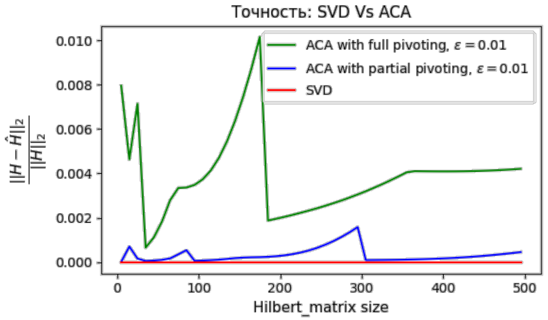
В некоторых случаях он не может получить надежные аппроксимации низкого ранга.

Сложность метода: общая сложность представленного алгоритма $O((m+n)r^2)$ операций и $O((m+n)r)$ вычислений элементов матрицы.









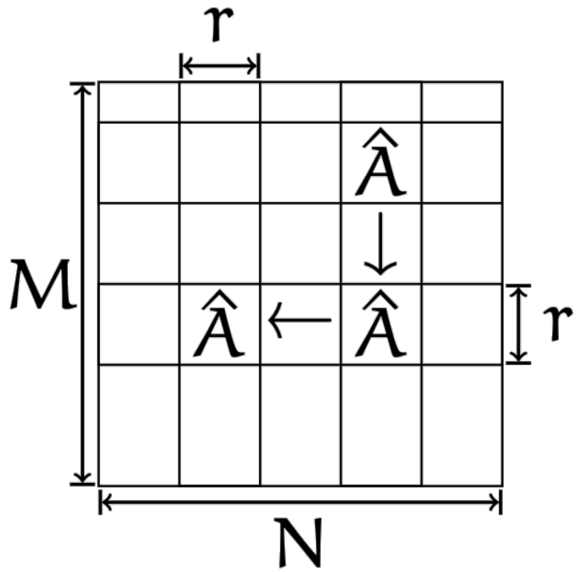
Метод maxvol

Естественное обобщение ACA возникло после появления теоремы, оценивающей аппроксимацию строками и столбцами подматрицы максимального объема. Если сингулярные числа A быстро убывают, то метод эффективен:

Теорема 1.1. *Пусть матрица $A \in \mathbb{C}^{n\times n}$ и подматрица $A_{11} \in \mathbb{C}^{r\times r}$ имеет максимальный объём среди всех подматриц $r \times r$ в A. Тогда крестовая аппроксимация, построенная на A_{11} (см. (2)), удовлетворяет*

$$\|A - C\hat{A}^{-1}R\|_C \leqslant (r+1)\cdot\sigma_{r+1}(A).$$

Выше мы считаем, что A_{11} обратима, т. е. $\operatorname{rank}(A) \geqslant r$.



Отыскание максимальной подматрицы является NP-полной задачей, поэтому будем искать *доминантную подматрицу*:

Теорема 2.2. *Пусть $\hat{A}_{dom}, \hat{A}_{\max} \in \mathbb{C}^{r\times r}$ это соответственно доминантная матрица и матрица максимального объёма. Тогда выполнено неравенство*

$$\frac{|\det \hat{A}_{\max}|}{|\det \hat{A}_{dom}|} \leqslant r^{r/2}.$$

Метод maxvol

Алгоритм нахождения доминантной подматрицы позволяет сформулировать следующая теорема: будем менять строки/столбцы пока |*c_ij*| > *1.*

Algorithm 2 maxvol [5]

Input: Matrix $A \in \mathbb{C}^{M \times N}$, starting sets of row indices $\mathcal{I}$ and column indices $\mathcal{J}$ of cardinality r. For example, $\mathcal{I} = \mathcal{J} = \{1, ..., r\}$.

Output: The row and column indices of the dominant submatrix of rank r written in $\mathcal{I}$ and $\mathcal{J}$.

```
1:  while replacements occur do
2:    for changes_in in {I, J} do
3:      C := A_{:,J} A_{I,J}^{-1} if we change the selected set of rows I
4:      C := (A_{I,J}^{-1} A_{I,:})^* if we change the selected set of columns J
5:      Select i and j corresponding to max_{i,j} |C_{i,j}|
6:      while |C_{i,j}| > 1 do
7:        Update C
8:        Replace index j with i in the set changes_in (in either I or J)
9:        Select i and j corresponding to max_{i,j} |C_{i,j}|
10:     end while
11:   end for
12: end while
```

Сложность метода: *O((M+N)r·iter+(M+N)r*^{2}*·IT)*,
где *iter* – общее количество замен строк и столбцов, то есть количество итераций цикла “while” в строке 6,
IT – количество проходов по строкам и столбцам, то есть общее количество итераций цикла “for” (строка 2).

Теорема 2.1. *Пусть* $A \in \mathbb{C}^{n \times r}$ *и выбрана подматрица* $\hat{A} \in \mathbb{C}^{r \times r}$ *(не умоляя общности считаем, что она располагается на первых* r *строках* A*, см. рис.* $\boxed{1}$*). Тогда при замене строки* $1 \leqslant j \leqslant r$ *на строку* $r < i \leqslant n$ *объём изменится как*

$$\frac{|\det(\hat{A}_{new})|}{|\det(\hat{A})|} = |C_{i,j}|. \qquad (3)$$

Где $C_{i,j}$ *это элемент матрицы* $C = A\hat{A}^{-1}$.

Преимущества:
– Если матрица А малого ранга и мы взяли небольшое число строк для ее приближения, то это весьма эффективно (в силу Теоремы 1.1);
– Быстрое разложение по сравнению с SVD, QR.

Недостатки:
– Объем доминантной матрицы может сильно отличаться от объема максимальной (в силу Теоремы 2.2).
– Метод с переменными направлениями не гарантирует нахождение доминантной матрицы.

Формула Шермона-Моррисона-Вудбери $\longrightarrow$ $\left(A + uv^{\mathsf{T}}\right)^{-1} = A^{-1} - \frac{A^{-1}uv^{\mathsf{T}}A^{-1}}{1 + v^{\mathsf{T}}A^{-1}u}.$

Метод rect-maxvol (maxvol-2)

Классический maxvol хорош для «стоячих» матриц малого ранга, поэтому нужны модификации. rect-maxvol является естественным расширением исходного алгоритма для нахождения прямоугольной подматрицы максимального объема.

Algorithm 1 rect_maxvol ("Greedy" maximization of the volume of submatrix)

Require: A full-rank $A \in \mathbb{C}^{N \times r}$, $N > r$, parameter τ

Ensure: A submatrix $\widehat{A}$, a set of pivot rows $\{p\}$ and a matrix of coefficients C such, that $A = C\widehat{A}, \forall i \notin \{p\} : \|C_i\|_2 \le \tau$

Start with a non-singular square submatrix $\widehat{A}$ {Result of the **maxvol**}
$\{p\} \leftarrow$ pivot rows; $C \leftarrow A\widehat{A}^{-1}$; $\forall i : L_i \leftarrow \|C_i\|_2^2$ {Result of the **maxvol**}
$i \leftarrow \mathrm{argmax}_{i \notin \{p\}}(L_i)$ {Find maximal row in C}
while $L_i > \tau^2$ **do**
  $\{p\} \leftarrow \{p\} + i$ {Extend set of pivots}
  $\widehat{A} \leftarrow \begin{bmatrix} \widehat{A} \\ A_i \end{bmatrix}$ {Extend $\widehat{A}$}
  $C \leftarrow \begin{bmatrix} C - \frac{CC_i^* C_i}{1+C_iC_i^*} & \frac{CC_i^*}{1+C_iC_i^*} \end{bmatrix}$ {Rank-1 update of C}
  $\forall j : L_j \leftarrow L_j - \frac{|C_jC_i^*|^2}{1+C_iC_i^*}$ {Update lengths of rows of C}
  $i \leftarrow \mathrm{argmax}_{i \notin \{p\}}(L_i)$ {Find maximal row in C}
end while
if $\widehat{C}$ is required to be identity **then**
  $\widehat{C} = I$
end if
return $C, \widehat{A}, \{p\}$

Сложность метода: *O(N(2K²-r²))*, где *N*, *r* - число строк и столбцов матрицы *A* (*N*>=*r*), *K* - желаемое число строк в подматрице максимального объема

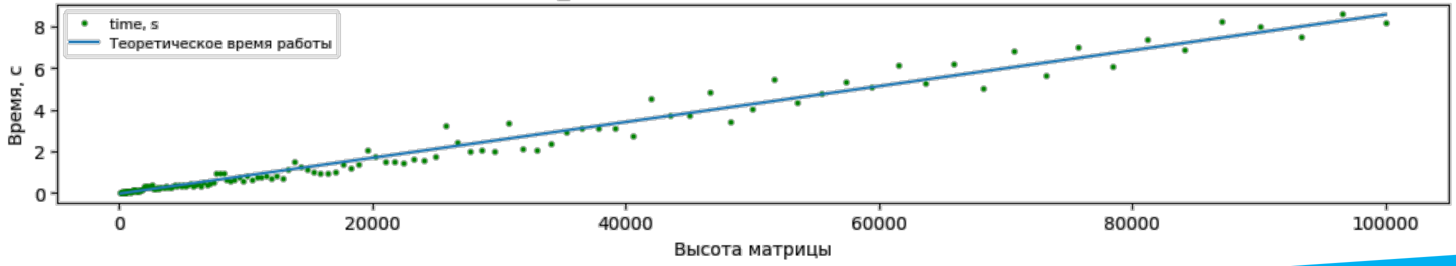
Объем квадратной матрицы *A* имеет естественный геометрический смысл как объем параллелепипеда, натянутого на строки *A*, и равен произведению его сингулярных значений.

Это определение можно прямо обобщить на прямоугольный случай как $\sqrt{det A^* A}$ или $\sqrt{det AA^*}$ в зависимости от формы: $\mathrm{vol}(A) = \sqrt{\det(A^*A)}$

По сравнению с классическим maxvol, rect-maxvol позволяет применять алгоритм для более широких целей:

1. *Рекомендательные системы*
2. *Wireless communications*

Сравнение скорости работы алгоритма rect_maxvol в завистимости от размера (высоты) матрицы (ширина фиксирована)



Метод householder maxvol-2

Для ускорения работы метода maxvol-2 предлагается выполнять обновление матрицы C при помощи Хаусхолдера. Сложность алгоритма снижается в *(n-r)* раз и составляет *O(Mr(n-r))*.

$$\hat{A}' = \begin{bmatrix} Q^* \\ C_{i,:} \end{bmatrix} M.$$

Matrix $\begin{bmatrix} Q^* \\ C_{i,:} \end{bmatrix} \in \mathbb{C}^{(k+1)\times r}$ can be made unitary with the aid of the Householder reflector $H\mathbb{C}^{r\times r}$ such that

$$C_{i,:}H = \begin{bmatrix} \|C_{i,:}\|_2 & 0 & \dots & 0 \end{bmatrix} = \begin{bmatrix} \sqrt{l_i} & 0 & \dots & 0 \end{bmatrix}$$

and through normalization of the first column by the matrix

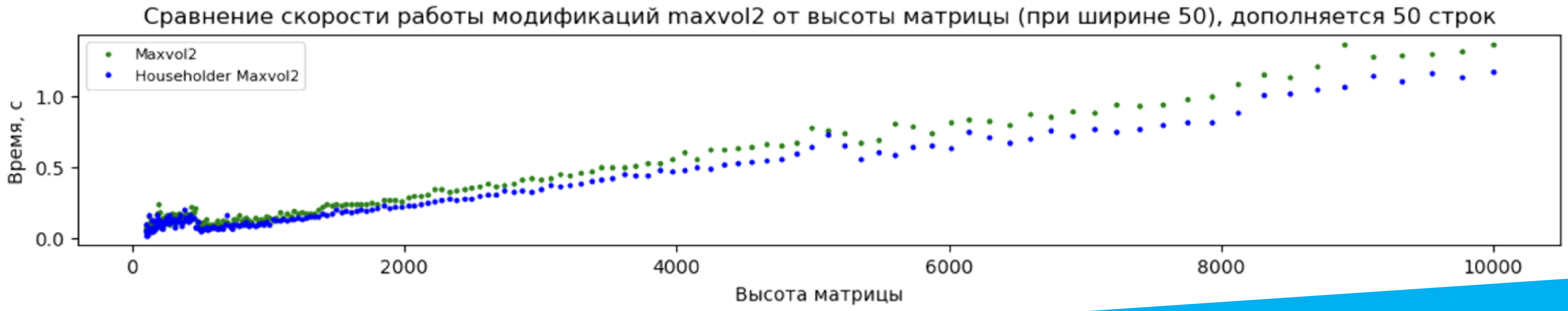
$$D = diag(\frac{1}{\sqrt{1+l_i}}, 1, \dots, 1) \in \mathbb{C}^{r\times r}.$$

$$\hat{A}' = \begin{bmatrix} Q^* \\ C_{i,:} \end{bmatrix} HDD^{-1}HM = Q'D^{-1}HM = Q'M'$$

$$(M')^{-1} = M^{-1}HD,$$

$$C' = \tilde{A}'(M')^{-1} = \tilde{A}'M^{-1}HD,$$

Данная модификация позволяет добиться ускорения работы maxvol-2, сохранив при этом относительную норму:



Метод 2 maxvol-2

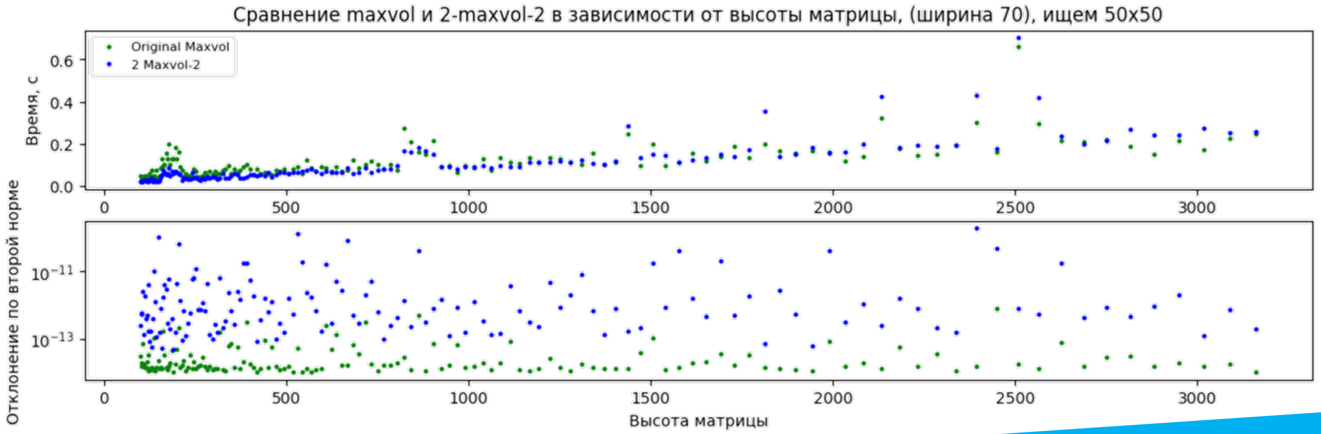
Метод 2 maxvol-2 объединяет в себе алгоритм поиска доминантной матрицы, а также ее расширения по столбцам и по строкам.

Algorithm 8 Fast CGR approximation search (2 maxvol2)

Input: Matrix $A \in \mathbb{C}^{M \times N}$, the starting sets of row indices $\mathcal{I}$ and column indices $\mathcal{J}$ of cardinality r, and final sizes n and m of C and R respectively.

Output: The updated sets $\mathcal{I}$ (of the row indices of R) and $\mathcal{J}$ (of the column indices of C) corresponding to a submatrix of a large projective volume.

1: $\mathrm{maxvol}(A, \mathcal{I}, \mathcal{J})$
2: $C := A_{:,\mathcal{J}}$
3: $R := A_{\mathcal{I},:}$
4: $\mathrm{maxvol2}(C, \mathcal{I}, n)$
5: $\mathrm{maxvol2}(R^T, \mathcal{J}, m)$



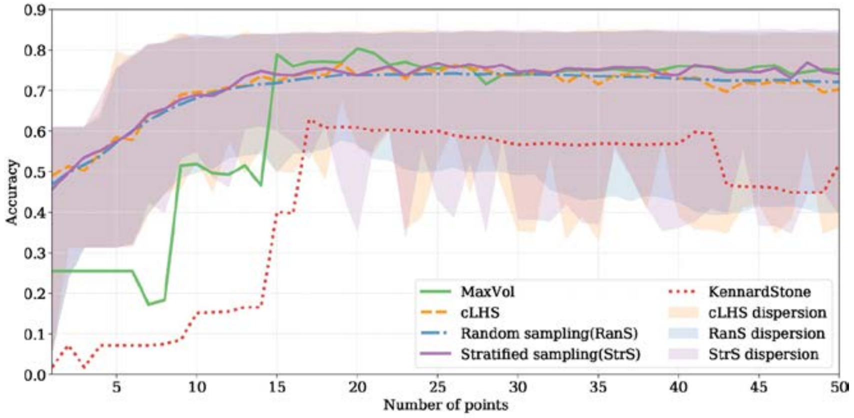
- В случае использования данного алгоритма гарантируется более хорошая оценка сложности
- При этом используется особая модификация метода maxvol-2, применяющая преобразование Хаусхолдера

Сложность метода:

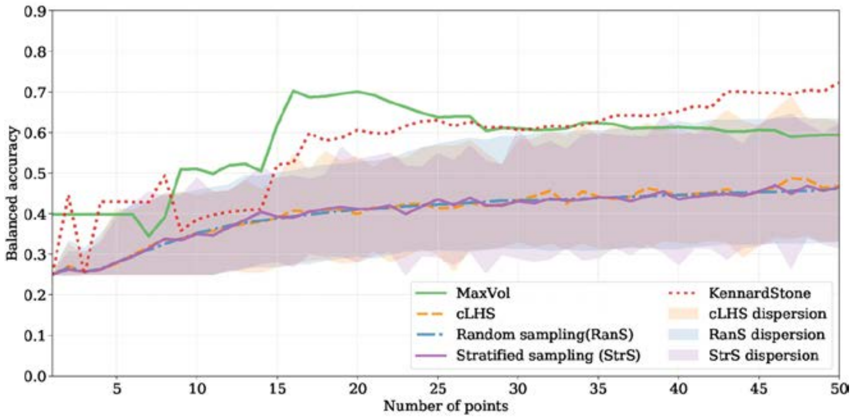
$$\approx O\left((M+N)r^2 \ln r \cdot IT + (M+N)(m+n)r\right)$$

Использование maxvol в приложениях

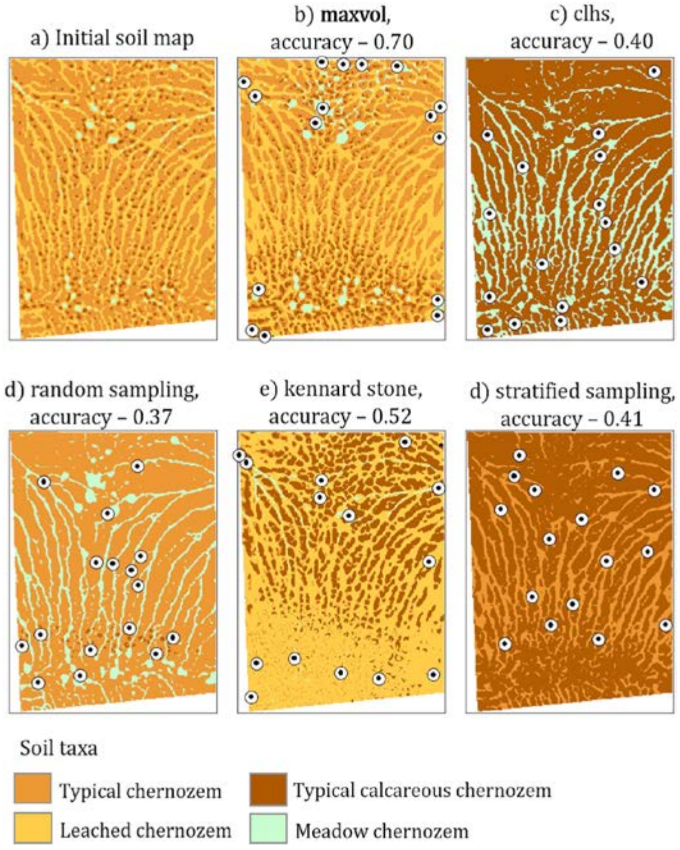
Задача *Optimal soil sampling design:* требуется определить набор точек для взятия проб почвы, по которым можно восстановить тип почв некоторой области.



(a) Comparison by plain accuracy.



(b) Comparison by balanced accuracy.

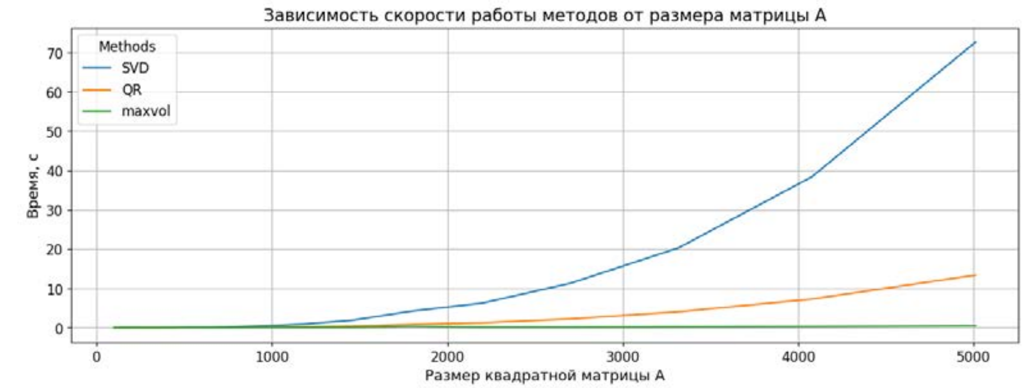


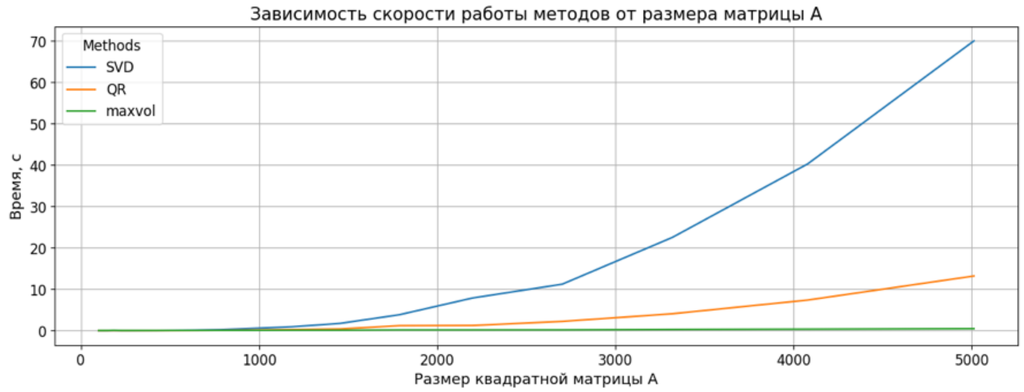
Методом rect-maxvol выбираются точки, которые достаточно хорошо позволяют приблизить всю местность

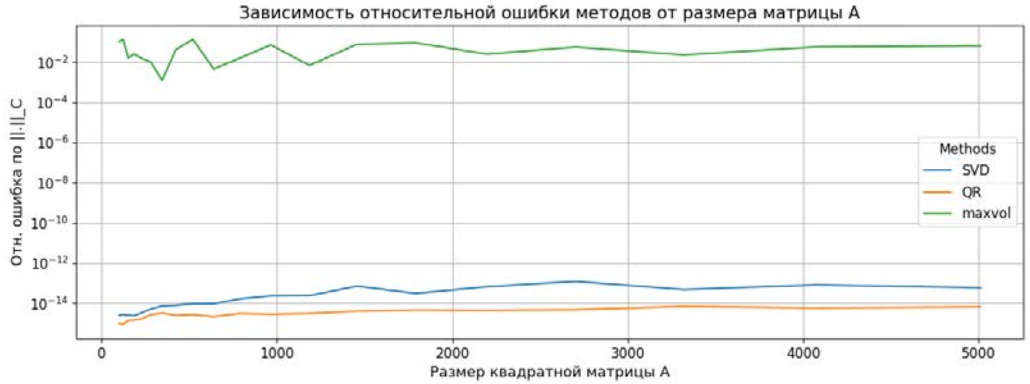
Его скорость работы достаточна, чтобы обработать матрицы признакового описания большого объема, что позволяет использовать на практике

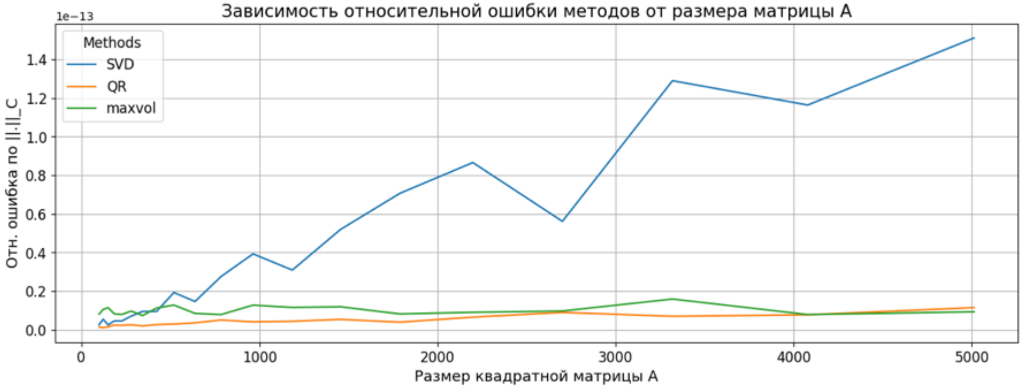
Сравнение maxvol, SVD и QR

Было решено провести сравнение стандартных аппроксимаций (SVD, QR) и метода maxvol с переменными направлениями: сначала на случайных матрицах, затем на матрицах с экспоненциально убывающими синг. числами.





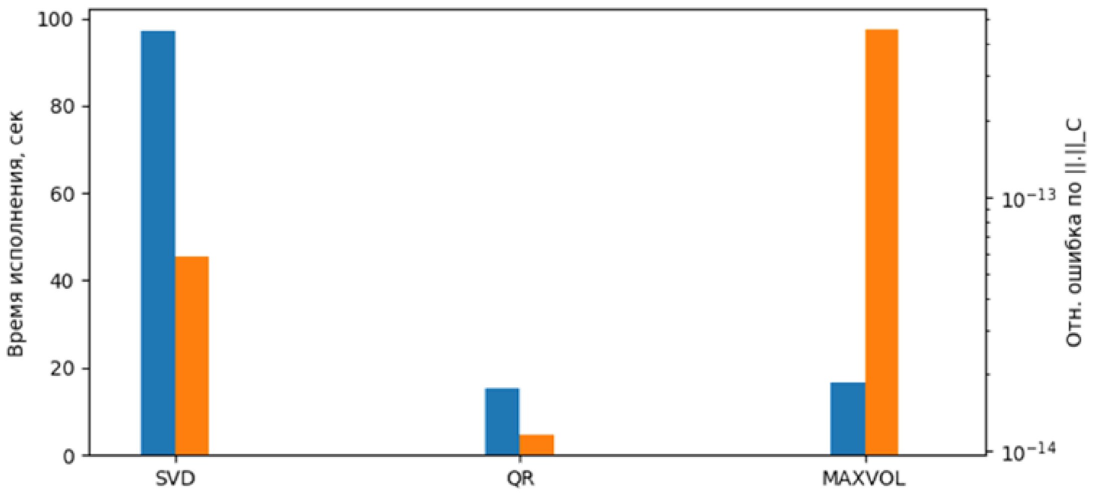


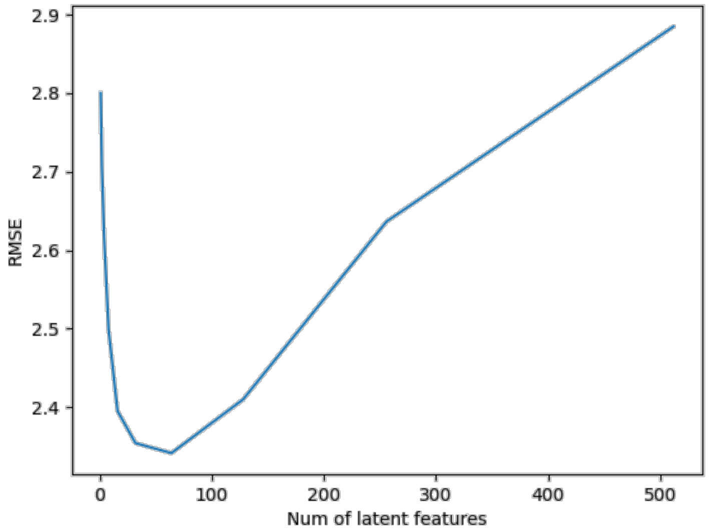


Вывод: в обоих случаях maxvol работает быстрее, однако, если сингулярные числа матрицы случайны, относительная ошибка по норме Чебышева большая, а если они быстро убывают, то построенный метод дает лучшую аппроксимацию.

Приложение: рекомендательные системы

Используем MovieLens 1M Dataset: стабильный набор эталонных данных. 1 млн оценок от 6000 пользователей для 4000 фильмов. Всего около 5% заполненных элементов в матрице, т.е. матрица разреженная.





Сравнение скорости работы и ошибки алгоритмов для различных малоранговых аппроксимаций (см. левый график), rect-maxvol взят с ограничением на ранг *r = 64* (оптимальный ранг определен с помощью итеративного SVD, см. правый график).

Вывод: maxvol можно использовать в качестве препроцессора для матрицы (пользователь x фильм) для уменьшения памяти требуемой для ее хранения, также для предобработки данных для SVD

Выводы

Результаты: в ходе работы мы разобрались с методом maxvol и несколькими его модификациями, запрограммировали их на Python. Затем провели эксперименты по сравнению алгоритмов со стандартными аппроксимациями на случайных матрицах и матрицах из приложений, сравнили некоторые методы между собой.

Вывод: метод maxvol является хорошим способом крестовых аппроксимаций для матриц специального вида (стоячие, лежачие с априорной информацией о низком ранге), используется в приложениях, например, в рекомендательных системах и *wireless communications.*

План на будущее: сравнить все полученные алгоритмы на одинаковых данных между собой, более подробно и реализовать самый новый алгоритм maxvol-proj.

Рассматриваемая литература и ссылка на GitHub

1. Bebendorf, M., Rjasanow, S. Adaptive Low-Rank Approximation of Collocation Matrices. Computing 70, 1–24 (2003)
2. S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, N. L. Zamarashkin, How to find a good submatrix, in: V. Olshevsky, E. Tyrtyshnikov (Eds.), Matrix Methods: Theory, Algorithms, Applications, World Scientific, Hackensack, NY, 2010, pp. 247–256.
3. A. Mikhalev, I.V. Oseledets, Rectangular maximum-volume submatrices and their applications, Linear Algebra and its Applications,Volume 538,2018, Pages 187-211.
4. Alexander Osinsky. Rectangular maximum volume and projective volume search algorithms. Arxiv, 2019.
5. Ballani, Jonas, and Daniel Kressner. “Matrices with Hierarchical Low-Rank Structures.” Lecture Notes in Mathematics, vol. 2173, Springer Verlag, 2016, pp. 161–209, doi:10.1007/978-3-319-49887-4_3.
6. Д. А. Желтков, Е. Е. Тыртышников, Параллельная реализация матричного крестового метода, Выч. мет. программирование, 2015, том 16, выпуск 3, 369– 375

Репозиторий GitHub с кодом проекта: https://github.com/ol3gka/AI_Masters_NLA_project_1_Matrix_Masters

Спасибо за внимание!