

# Исследование одного алгоритма метода главных компонент

Салова Яна Алексеевна, гр.20.Б04-мм

Санкт-Петербургский государственный университет  
Прикладная математика и информатика  
Вычислительная стохастика и статистические модели

**Научный руководитель:** д.ф.-м.н., профессор Ермаков М.С.

**Рецензент:** Старший научный сотрудник, Петербургское  
отделение математического института РАН, лаборатория  
статистических методов В.Н.Солев

Санкт-Петербург, 2024

## Мотивация

- Одной из ключевых задач в методе главных компонент является нахождение наибольшего собственного числа и соответствующего собственного вектора.
- Она используется во многих приложениях.
- Позволяет выделить направление максимальной дисперсии данных, что способствует их лучшему пониманию и упрощению структуры.

## Цель дипломной работы

- Исследовать алгоритм нахождения наибольшего собственного числа и соответствующего собственного вектора.

## Задачи дипломной работы

- Знакомство с алгоритмом и написание программы его моделирования.
- Анализ сходимости алгоритма и доказательство его состоятельности.
- Исследование свойств алгоритма в зависимости от размерности и от значения второго собственного числа.

# Математическая постановка задачи

Рассматриваемый алгоритм является вариантом метода стохастического градиента для решения следующей экстремальной задачи:

## Задача нахождения наибольшего собственного вектора

Дана ковариационная матрица  $A \in \mathbb{R}^{n \times n}$ . Необходимо найти вектор  $x \in \mathbb{R}^n$  единичной нормы, который максимизирует функцию

$$\lambda = \max_{\|x\|=1} \|Ax\|,$$

где  $\lambda$  — наибольшее собственное значение матрицы  $A$ , а  $x$  — соответствующий собственный вектор.

## Концепция

- 1 Моделируем случайный вектор  $\mathbf{u}_1$ , равномерно распределённый на сфере.
- 2 Моделируем вектор  $\mathbf{u}_2$ , ортогональный  $\mathbf{u}_1$ , и ищем наибольшее собственное число  $\lambda$  и собственный вектор  $\mathbf{v}$  в подпространстве, образованном  $\mathbf{u}_1$  и  $\mathbf{u}_2$ .
- 3 Повторяем предыдущий шаг, моделируя новые ортогональные векторы и обновляя  $\mathbf{v}$  и  $\lambda$ , пока не выполнится критерий остановки.

## Предложение

С ростом числа итераций последовательность приближений сходится по вероятности к наибольшему собственному числу  $\lambda_{\max}$  и соответствующему ему уникальному собственному вектору матрицы  $A$ .

# Моделирование для матрицы $3 \times 3$

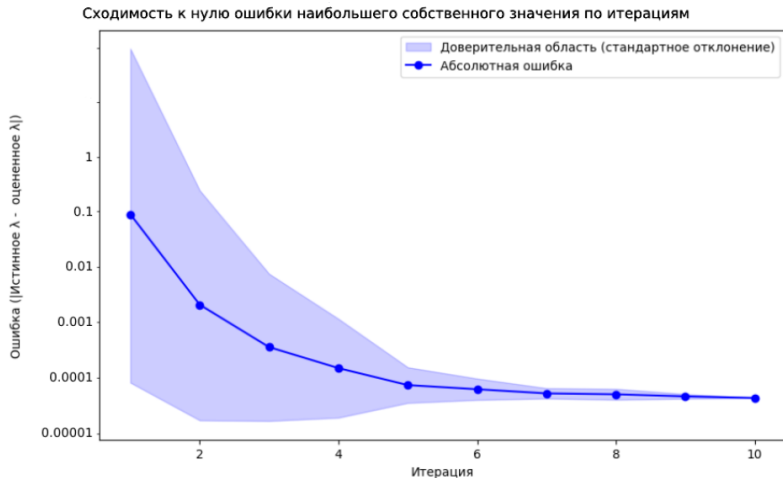


Рис.: Ошибка вычисления собственного числа

# Моделирование для матрицы $3 \times 3$ .

График показывает погрешности

$$\hat{\lambda}_{ik} = (\lambda - \lambda_k), \quad k \in [1, 100], \quad i \in [10, 100],$$

где  $\lambda$  — истинное значение,  $\lambda_k$  — средние значения по  $k$  запускам алгоритма для разных  $i$ .

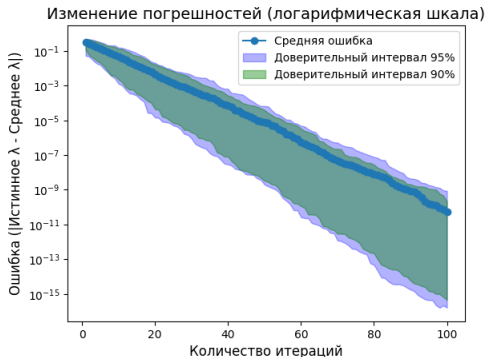


Рис.: Средняя погрешность



# Моделирование для матрицы $6 \times 6$

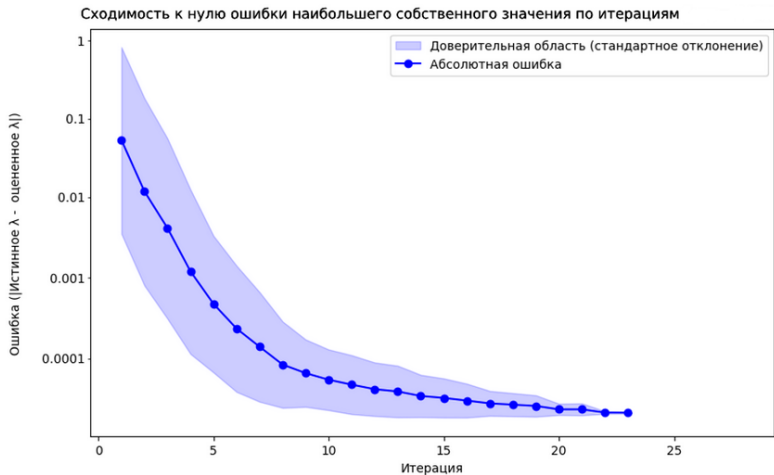


Рис.: Ошибка вычисления собственного числа

# Моделирование для матрицы $6 \times 6$ .

График показывает погрешности

$$\hat{\lambda}_{ik} = (\lambda - \lambda_k), \quad k \in [1, 100], \quad i \in [10, 100],$$

где  $\lambda$  — истинное значение,  $\lambda_k$  — средние значения по  $k$  запускам алгоритма для разных  $i$ .

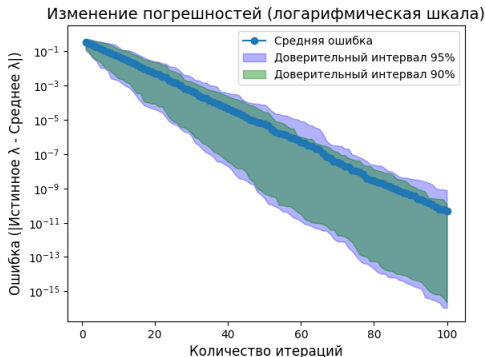


Рис.: Средняя погрешность

# Моделирование для матрицы $11 \times 11$

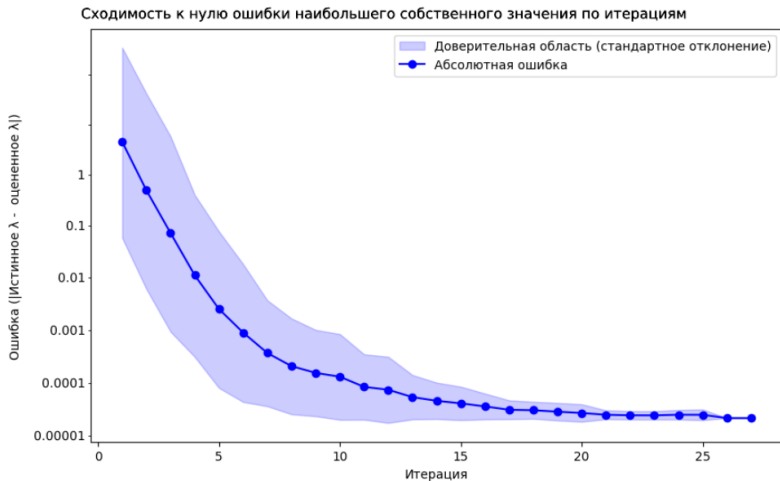


Рис.: Ошибка вычисления собственного числа

# Моделирование для матрицы $11 \times 11$ .

График показывает погрешности

$$\hat{\lambda}_{ik} = (\lambda - \lambda_k), \quad k \in [1, 100], \quad i \in [10, 100],$$

где  $\lambda$  — истинное значение,  $\lambda_k$  — средние значения по  $k$  запускам алгоритма для разных  $i$ .

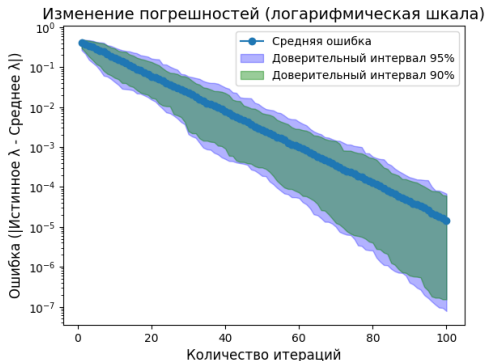


Рис.: Средняя погрешность

# Моделирование для матрицы $20 \times 20$

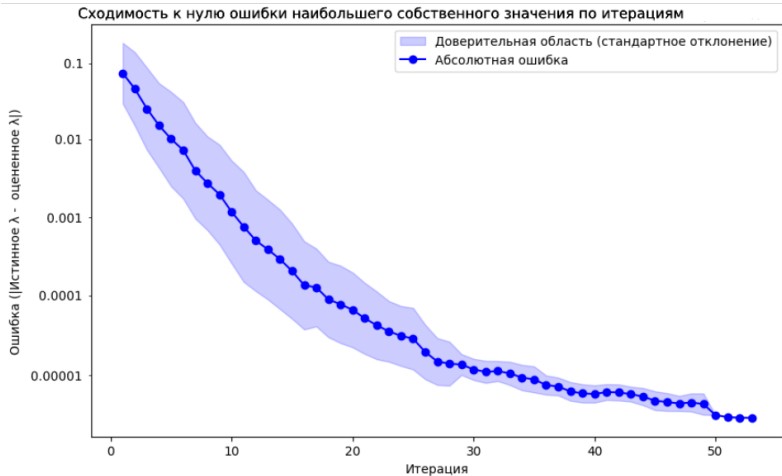


Рис.: Ошибка вычисления собственного числа

# Моделирование для матрицы $20 \times 20$ .

График показывает погрешности

$$\hat{\lambda}_{ik} = (\lambda - \lambda_k), \quad k \in [1, 100], \quad i \in [10, 100],$$

где  $\lambda$  — истинное значение,  $\lambda_k$  — средние значения по  $k$  запускам алгоритма для разных  $i$ .

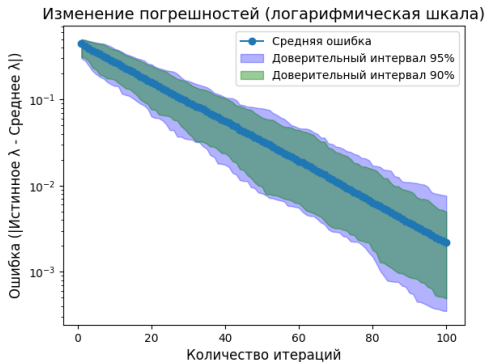


Рис.: Средняя погрешность

# Моделирование для матрицы $3 \times 3$ .

Анализ влияния разницы между первым и вторым собственными числами на скорость сходимости алгоритма

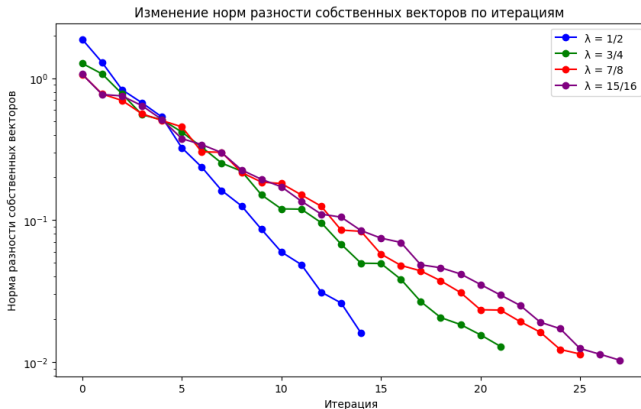


Рис.: Влияние значений вторых собственных чисел

# Моделирование для матрицы $6 \times 6$ .

Анализ влияния разницы между первым и вторым собственными числами на скорость сходимости алгоритма

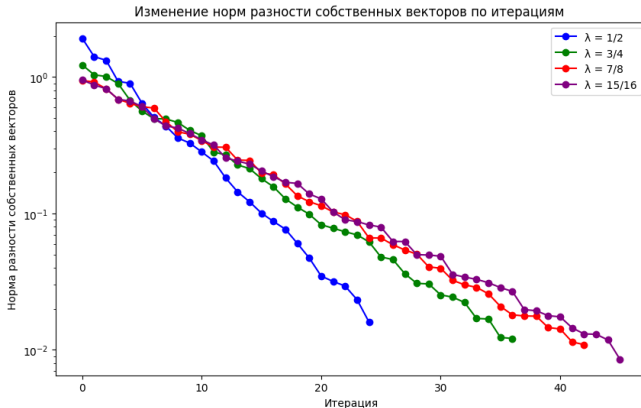


Рис.: Влияние значений вторых собственных чисел



# Моделирование для матрицы $11 \times 11$ .

Анализ влияния разницы между первым и вторым собственными числами на скорость сходимости алгоритма

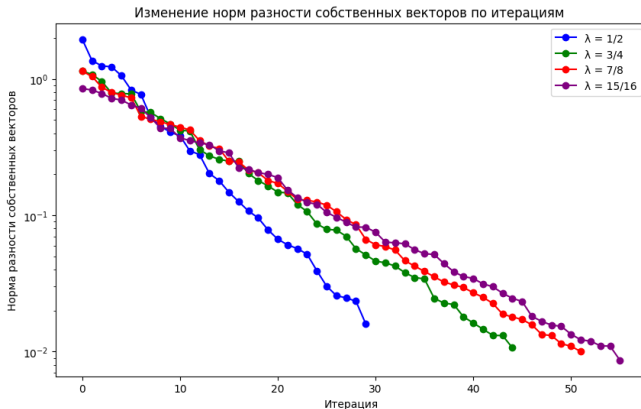


Рис.: Влияние значений вторых собственных чисел

# Моделирование для матрицы $20 \times 20$ .

Анализ влияния разницы между первым и вторым собственными числами на скорость сходимости алгоритма

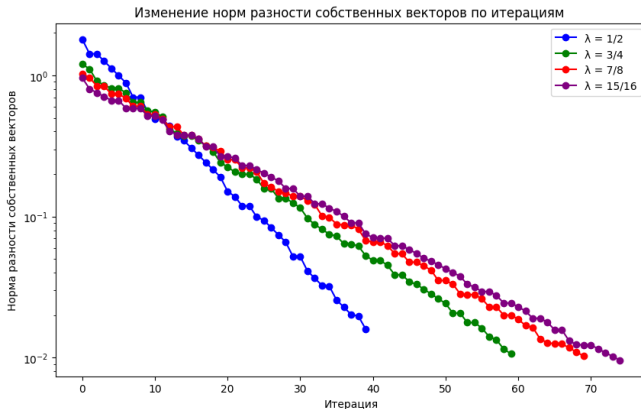


Рис.: Влияние значений вторых собственных чисел

Количество итераций  $k$  для достижения точности  $\varepsilon$ :

$$k \approx \left\lceil \frac{\log(\varepsilon/\delta_0)}{\log \left( \left| \frac{\lambda_2}{\lambda_1} \right| \right)} \right\rceil,$$

где:

- $\varepsilon$  — заданная точность,
- $\delta_0$  — начальная ошибка,
- $\left| \frac{\lambda_2}{\lambda_1} \right|$  — отношение двух наибольших собственных чисел матрицы  $\mathbf{A}$ .

- Разработан и реализован новый простой алгоритм для поиска наибольшего собственного числа и соответствующего собственного вектора.
- Исследованы его сходимость, вычислительная сложность и количество необходимых итераций.
- Проведено моделирование работы алгоритма на различных наборах данных

Дальнейшее развитие алгоритма видится в его расширении на высокие размерности и в значительном улучшении скорости сходимости посредством метода Монте-Карло.