Практическое задание № 7

АНАЛИЗ ГЛАВНЫХ КОМПОНЕНТ

Цель работы: получить практику анализа статистических данных с использованием анализа главных компонент.

Содержание задания

1. Общие сведения

1. Ознакомиться с материалами лекции № 7.

2. Установить необходимое программное обеспечение. При выполнении задания наверняка понадобятся **Python 3**, **NumPy**, **SciPy**, и **Matplotlib**.

3. Ознакомиться с содержимым папки с заданием, которая включает в себя файлы, представленные ниже.

`main.py` – «основной» модуль, необходимый для выполнения задания, который поможет выполнить его поэтапно. Настоящий программный код не требует какой-либо коррекции!

`data1.mat` – база данных для выполнения первой части задания.

`data2.mat` – база данных для выполнения второй части задания.

`displayData.py` – модуль, содержащий функцию `displayData`, которая необходима для визуализации данных. Данный модуль не требует коррекции!

`featureNormalize.py` – модуль, содержащий функцию `featureNormalize`, которая необходима для нормализации признаков. Данный модуль не требует коррекции!

`pca.py` – модуль, содержащий функцию `pca`, которая необходима для вычисления главных компонент данных, содержащихся в матрице объекты-признаки.

`projectData.py` – модуль, содержащий функцию `projectData`, которая необходима для вычисления проекций нормализованных входов матрицы объекты-признаки.

`recoverData.py` – модуль, содержащий функцию `recoverData`, которая необходима для восстановления аппроксимации исходных данных, размерность которых была уменьшена.

4. Поэтапно выполнить задание, связанное с реализацией и исследованием анализа главных компонент.

5. Ответить на вопросы, необходимые для составления отчета по данному практическому заданию. Отчет сдается на проверку в печатной или письменной форме в указанные сроки.

2. Анализ главных компонент

При выполнении данного задания требуется заполнить пустые места программного кода в блоках с комментарием «Ваш код здесь». Данную процедуру необходимо выполнить для следующих функций: `pca`, `projectData`, `recoverData`.

1. Загрузите базу данных, находящуюся в файле **`data1.mat`**. Она представляет собой некоторое искусственное множество объектов, описываемых двумерным вектором признаков. База данных не является размеченной, а анализ главных компонент является разновидностью методов обучения без учителя. Визуализация объектов из рассматриваемой базы данных представлена на рис. 1.

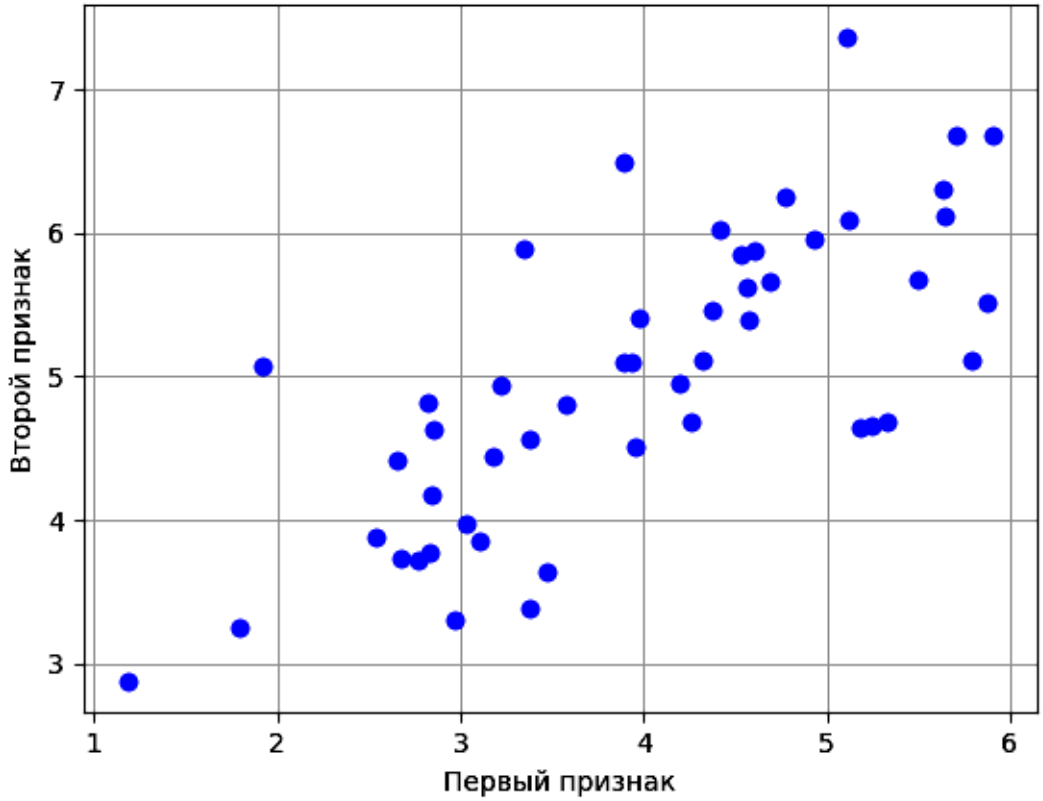

Рис. 1. Анализируемая искусственная база данных, для объектов которой наблюдается сильная корреляционная взаимосвязь между признаками

2. Завершите код в модуле **`pca.py`**, который позволит выполнить поиск ближайших средних для объектов, заключенных в матрице объекты-признаки. Выполнение данной процедуры было представлено в лекции № 7. При выполнении данной части задания могут понадобиться функции из библиотеки **NumPy**, представленные ниже.

`dot` – позволяет вычислить матричное произведение для двумерных массивов и скалярное произведение для одномерных массивов (без комплексного сопряжения).

`svd` – позволяет выполнить разложение по сингулярным числам матрицы.

3. Завершите код в модуле **`projectData.py`**, который позволит выполнить вычисление проекций нормализованных входов матрицы объекты-признаки. Выполнение данной процедуры было представлено в лекции № 7. При выполнении данной части задания могут понадобиться следующие функции из библиотеки **NumPy**: `dot` и `transpose`.

`transpose` – позволяет выполнить транспонирование массива. Для одномерного массива данная функция не оказывает никакого действия, а для двумерного массива использование функции соответствует обычному матричному транспонированию.

4. Завершите код в модуле **`recoverData.py`**, который позволит выполнить восстановление аппроксимации исходных данных, размерность которых была уменьшена. Выполнение данной процедуры было представлено в лекции № 7. При выполнении данной части задания может понадобиться функция `dot` из библиотеки **NumPy**.

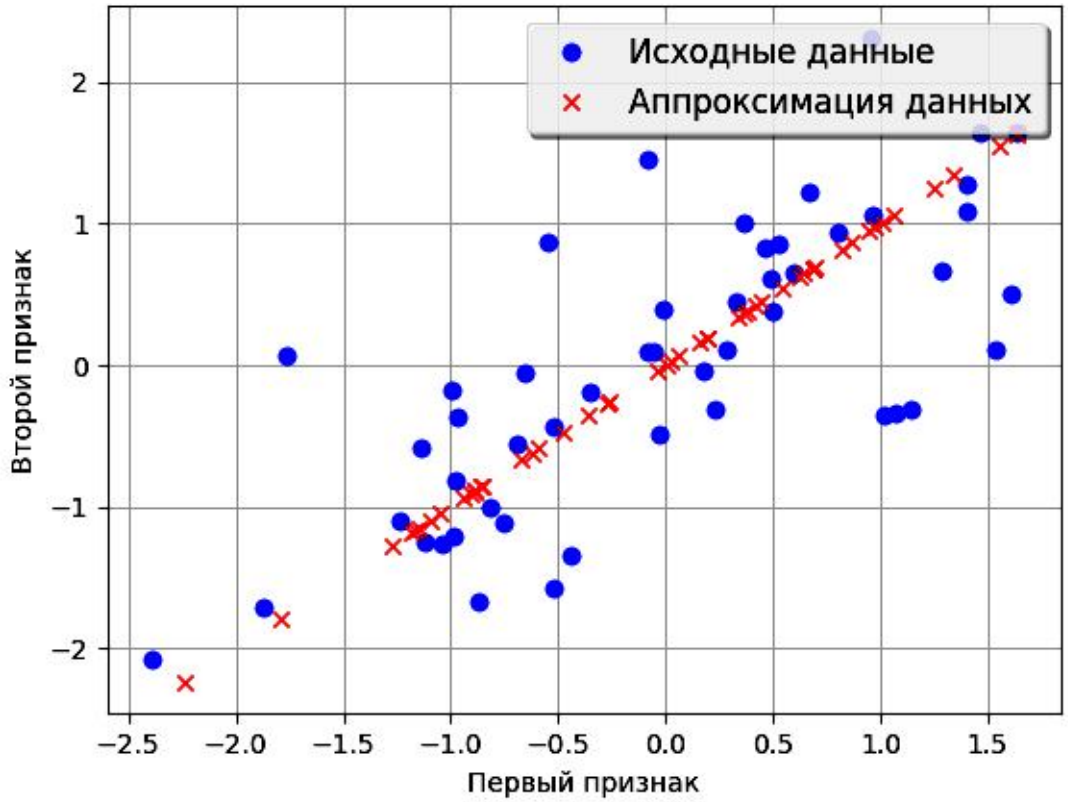

Рис. 2. Анализируемая искусственная база данных из рис. 1 и приближенное представление объектов этой базы данных, полученное с использованием анализа главных компонент

5. После завершения вышеуказанных пунктов выполните файл **main.py** для анализа результата вычисления аппроксимации анализируемых данных. Результат должен быть идентичным тому, что представлен на рис. 2, если на этапе сокращения размерности данных произошло ее уменьшение на 1.

Рис. 3. Анализируемая база данных лиц (слева) и приближенное представление объектов этой базы данных, полученное с использованием анализа главных компонент (справа). Число сохраненных главных компонент равно 100

Рис. 4. Визуализация 36 главных компонент (собственных лиц), рассчитанных для примеров из базы данных на рис. 3

6. Загрузите базу данных, находящуюся в файле **data2.mat**. Она представляет собой базу данных лиц, представленных в виде изображений с разрешением 32x32 пикселя. База данных не является размеченной. Визуализация объектов из рассматриваемой базы данных представлена на рис. 3 (слева).

7. Выполните сокращение размерности для рассматриваемой базы данных лиц, ограничившись 0, 10, 50 и 100 главными компонентами. Проанализируйте полученный результат. Пример аппроксимации объектов рассматриваемой базы данных представлен на рис. 3 (справа). Число сохраненных главных компонент в этом случае равнялось 100. Дополнительно на рис. 4 визуализированы 36 из 1024 собственных лиц, в структуре которых прослеживаются очертания лиц людей.

3. Вопросы для составления отчета

1. Как выглядят главные компоненты (собственные вектора) рассчитанные для объектов, содержащихся в базе данных **data1.mat** (**50 баллов**)?

2. Выполните аппроксимацию данных из базы в файле **data1.mat**, выполняя сокращение размерности на 1. Как выглядят аппроксимации первых трех примеров из базы данных (**50 баллов**)?

3. Сколько главных компонент при формировании аппроксимации данных, содержащихся в базе изображений лиц, Вам достаточно для того, чтобы лица были узнаваемы (**50 баллов**)? Настоящий вопрос является необязательным, но позволяет заработать дополнительные баллы!