Санкт-Петербургский политехнический университет
Петра Великого

Институт прикладной математики и механики
Высшая школа прикладной математики и вычислительной физики

Отчет по лабораторной работе № 2:
«Метод главных компонент»
по дисциплине: теория принятия экономических решений.

Выполнила студентка:
Заболотских Екатерина Дмитриевна
группа: 3630102/70301

Проверила:
Павлова Людмила Владимировна

Санкт-Петербург
2020 г.

Оглавление

Постановка задачи 2
Теория 3
Постановка задачи 3
Идея метода 3
Решение 5
Работа с модельными данными 5
Двумерное пространство 5
Трехмерное пространство 5
Работа с данными из репозитория German 5
Работа с данными Iris 5
Результаты 6
Работа с модельными данными 6
Двумерное пространство 6
Трехмерное пространство 8
Работа с данными из репозитория German 10
Работа с данными Iris 11
Выводы 12
Работа с модельными данными 12
Работа с данными репозитория German 12
Работа с данными Iris 12
Приложение 13

Список иллюстраций

Рисунок 1: 2D «хорошо» коррелированные данные 6
Рисунок 2: 2D «плохо» коррелированные данные 7
Рисунок 3: 3D «хорошо» коррелированные данные 8
Рисунок 4: 3D «хорошо» коррелированные данные 9
Рисунок 5: визуализация данных Iris 11

Постановка задачи

Реализовать метод главных компонент. Проверить работу метода на «хорошо» и «плохо» коррелированных модельных данных.

Визуализировать работу метода для двумерного и трехмерного случая данных.

Проанализировать данные из репозитория German и Iris, применив к ним метод главных компонент.

Теория

Метод главных компонент – один из основных подходов для понижения размерности исходных данных, который позволяет при этом сохранить важную информацию.

Вычисление главных компонент может быть сведено к вычислению сингулярного разложения матрицы данных или к вычислению собственных векторов и собственных значений ковариационной матрицы исходных данных.

Постановка задачи

Для данной многомерной случайной величины построить такое ортогональное преобразование координат, в результате которого корреляция между отдельными координатами обратятся в нуль, а дисперсия(ковариации) станет максимальной. (Своеобразное проецирование на вектор (плоскость).)

Идея метода

Пусть мы имеем набор векторов данных: $x_i \in \mathbb{R}^n,\ \ (i \in 1, \dots m)$. Для минимизации потерь данных при выборе признаков для рассмотрения (после работы метода), вектор(плоскость), на который мы будем проецировать, должен проходить через центр выборки. Поэтому лучше отцентрировать выборку – линейно сдвинуть ее так, чтобы средние значения признаков были равны 0.

Как нам известно математическое ожидание – «центр тяжести» величины, а дисперсия – ее «размеры» (разброс). Ковариационная матрица является обобщением дисперсии на случай многомерных случайных величин – она так описывает форму (разброс) случайной величины, как и дисперсия.

Находим такое ортогональное преобразование в новую систему координат, для которого были бы верны условия:

- Выборочная дисперсия данных вдоль первой координаты максимальна (главная компонента);
- Выборочная дисперсия данных вдоль второй координаты максимальна при условии ортогональности первой координате (вторая главная компонента);
- …

$$z = Lx$$

Где L – ортогональное преобразование.

То есть надо найти такой вектор, при котором максимизировался бы размер проекции (дисперсия) нашей выборки на него.

Выполняются условия: $cov(x_i, x_j) = 0$ при $i, j \in 1, \dots m;\quad i \neq j;$

$$\sum_{j=1}^{m} D[z_j] = \sum_{j=1}^{m} D[x_j] = \sum_{j=1}^{m} \sigma_{jj}\ .$$

Возьмем единичный вектор, на который будем проецировать наш случайный вектор X. Тогда проекция на него будет равна: $v^T X$. Дисперсия проекции на вектор будет соответственно равна: $D[v^T X] = E[(v^T X) * (v^T X)^T] = E[v^T X * X^T v] = v^T E[X * X^T] v = v^T \Sigma v$.

Заметим, что дисперсия максимизируется при максимальном значении $v^T \Sigma v$. Из отношения Релея: $M\vec{x} = \lambda\vec{x}$, где $\vec{x}$ – собственный вектор, λ – собственное значение. Количество собственных векторов и значений равны размеру матрицы (и значения могут повторяться).

Таким образом, направление максимальной дисперсии у проекции всегда совпадает с собственным вектором, имеющим максимальное собственное значение, равное величине этой дисперсии.

И это справедливо также для проекций на большее количество измерений – дисперсия (ковариационная матрица) проекции на m-мерное пространство будет максимальна в направлении m собственных векторов, имеющих максимальные собственные значения. Диагональные элементы ковариационной матрицы показывают дисперсии по изначальному базису, а ее собственные значения – по новому (по главным компонентам).

Часто требуется оценить объем потерянной (и сохраненной) информации. Удобнее всего представить в процентах. Мы берем дисперсии по каждой из осей и делим на общую сумму дисперсий по осям (т.е. сумму всех собственных чисел ковариационной матрицы).

Восстановление данных

$X' = X^t v^T + m$, где m – вектор средних.

Потерянная информация не восстанавливается. Тем не менее, если простота важнее точности, восстановленное значение отлично аппроксимирует исходное.

Решение

Работа с модельными данными

Двумерное пространство

Модельные данные составляем из двумерных нормально распределенных случайных векторов:

$$\mu = (0, 0)$$

Матрица ковариаций для «хорошо» коррелированных данных: $\Sigma = \begin{pmatrix} 10 & 9 \\ 9 & 10 \end{pmatrix}$.

Матрица ковариаций для «плохо» коррелированных данных: $\Sigma = \begin{pmatrix} 10 & 2 \\ 2 & 10 \end{pmatrix}$.

Трехмерное пространство

Модельные данные составляем из трехмерных нормально распределенных случайных векторов:

$$\mu = (0, 0, 0)$$

Матрица ковариаций для «хорошо» коррелированных данных: $\Sigma = \begin{pmatrix} 22 & 20 & 1 \\ 20 & 22 & 1 \\ 1 & 1 & 22 \end{pmatrix}$.

Матрица ковариаций для «плохо» коррелированных данных: $\Sigma = \begin{pmatrix} 22 & 3 & 3 \\ 3 & 22 & 3 \\ 3 & 3 & 22 \end{pmatrix}$.

Получаем новые вектора значений Z, и смотрим по матрице ковариаций на дисперсии векторов признаков, оцениваем, какой из них можно убрать из рассмотрения без потери важной информации.

Работа с данными из репозитория German

Составляем матрицу «объект-свойство» из всех 24 признаков репозитория, размерности $(100, 24)$.

Получаем преобразованные вектора Z и строим их ковариационную матрицу. Получим проценты «значимости» признаков по формуле:

$$P_i = \frac{D[z_i]}{\sum_{i=1}^{m} D[z_i]} * 100\%$$

Работа с данными Iris

Считываем данные из базы данных.
Получаем преобразованные вектора Z и строим ковариационную матрицу. Получаем проценты «значимости» признаков по формуле.

Замечание: перед тем, как сделать преобразования, исходные признаки стандартизируются.

Результаты

Работа с модельными данными

Двумерное пространство

Хорошо коррелированные данные

$\mu = (0, 0), \Sigma = \begin{pmatrix} 10 & 9 \\ 9 & 10 \end{pmatrix}$

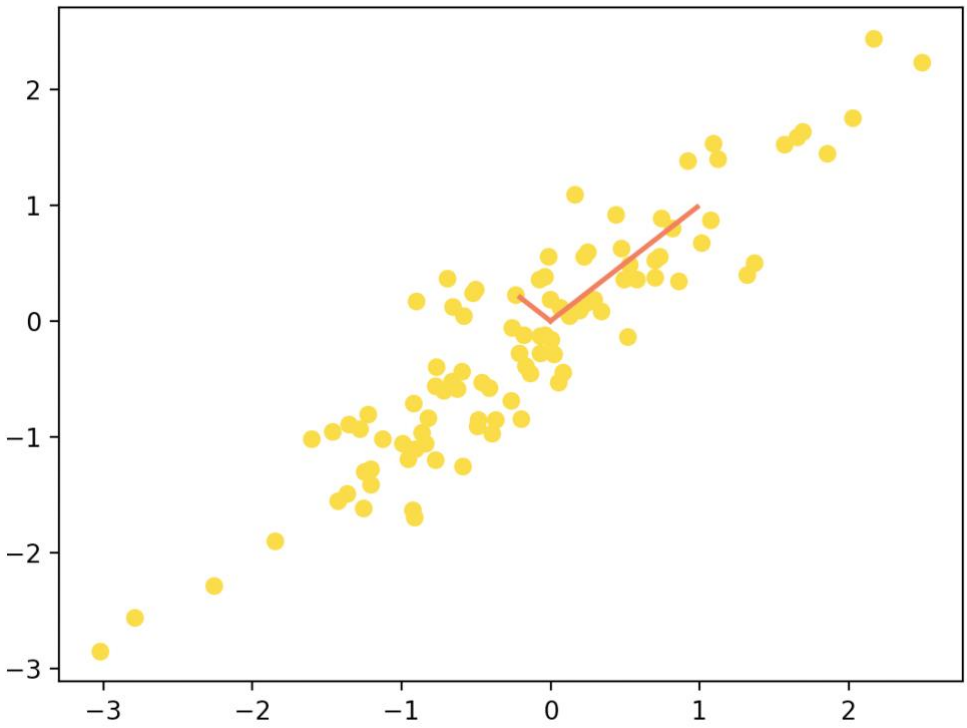

Рисунок 1: 2D «хорошо» коррелированные данные

Заметим, что данные были смоделированы так, что признаки сильно коррелированы. Поэтому по матрице ковариаций можем заметить, что дисперсия $D[z_2]$ мала относительно $D[z_1]$. Следовательно, мы можем убрать из рассмотрения второй признак и не потерять важную информацию.

$$\text{cov}(z) = \begin{pmatrix} 1.9 & 5.18e-16 \\ 5.18e-16 & 0.1148 \end{pmatrix}$$

Плохо коррелированные данные

$\mu = (0, 0), \Sigma = \begin{pmatrix} 10 & 2 \\ 2 & 10 \end{pmatrix}$

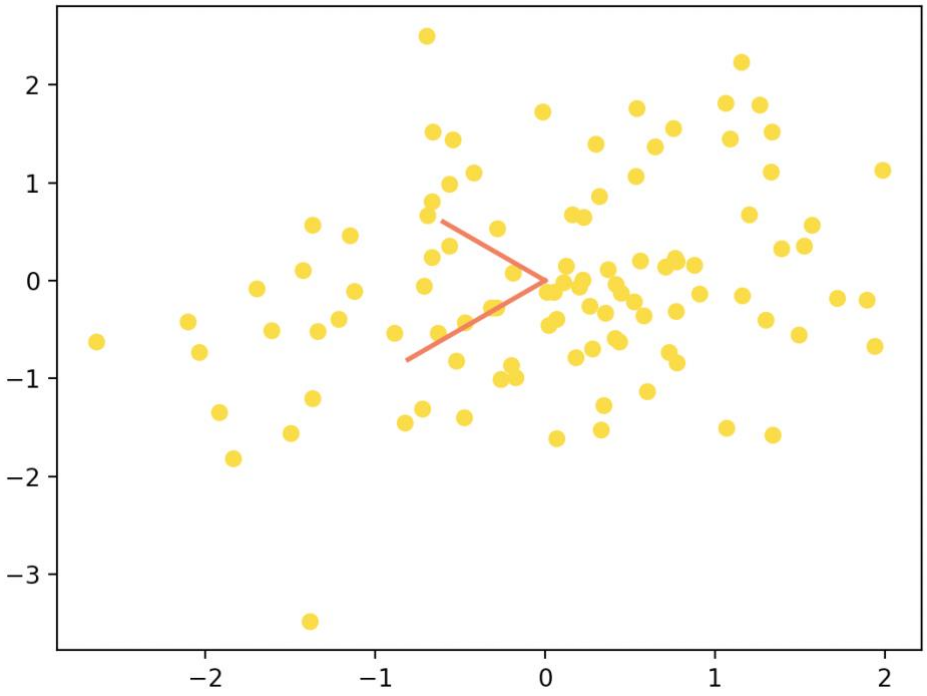

Рисунок 2: 2D «плохо» коррелированные данные

В данном случае данные уже имеют большой разброс, и не имеют сильной зависимости друг от друга. Дисперсии признаков примерно одинаковы. Никакой признак без потери важной информации не убрать.

$$\mathrm{cov}(z) = \begin{pmatrix} 1.002 & 1.012e-16 \\ 1.012e-16 & 1.0179 \end{pmatrix}$$

Трехмерное пространство

Хорошо коррелированные данные

$$\mu = (0,0,0), \Sigma = \begin{pmatrix} 22 & 20 & 1 \\ 20 & 22 & 1 \\ 1 & 1 & 22 \end{pmatrix}$$

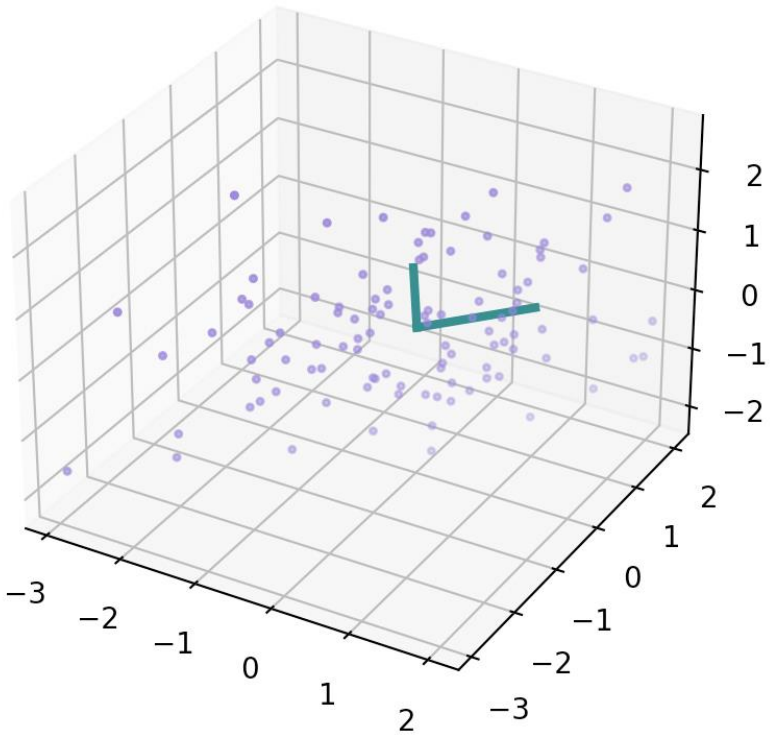

Рисунок 3: 3D «хорошо» коррелированные данные

Мы задали большую корреляцию между первым и вторым признаками. Заметим, что данные образовали плоскость, и видно всего два собственных вектора. Также в матрице ковариаций имеем малую дисперсию у второго признака, следовательно, можем его не рассматривать.

$$cov(z) = \begin{pmatrix} 2.027 & -4.29e-16 & 4.189e-17 \\ -4.296e-16 & 1.05e-16 & -9.77e-18 \\ 4.189e-17 & -9.77e-18 & 1.003 \end{pmatrix}$$

Плохо коррелированные данные

$$\mu = (0,0,0), \Sigma = \begin{pmatrix} 22 & 3 & 3 \\ 3 & 22 & 3 \\ 3 & 3 & 22 \end{pmatrix}$$

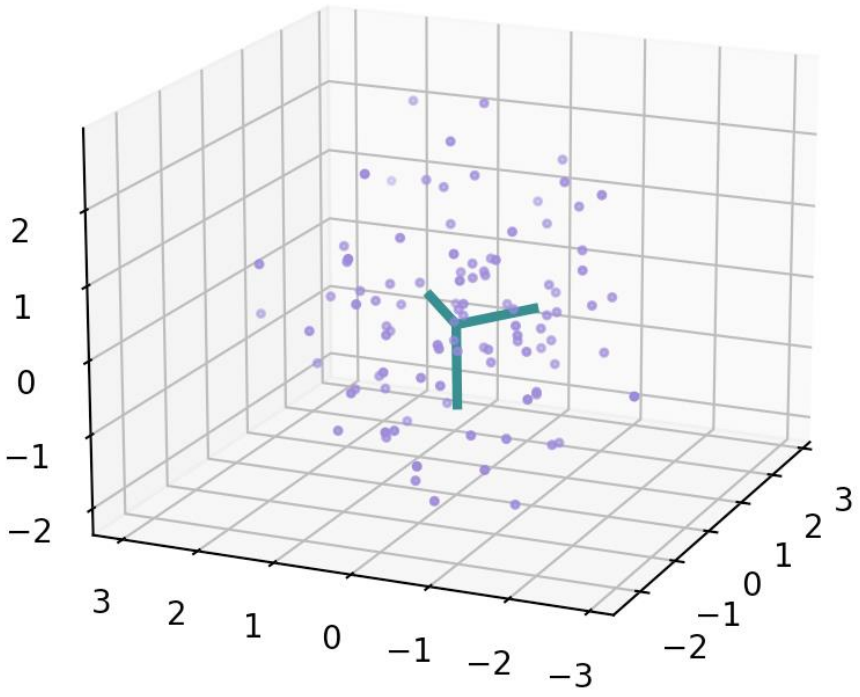

Рисунок 4: 3D «хорошо» коррелированные данные

В данном случае связи между признаками слабые, поэтому никакой признак убрать нельзя.

$$cov(z) = \begin{pmatrix} 1.28 & -4.7e-16 & 2.26e-16 \\ -4.7e-16 & 0.77 & 4.008e-16 \\ 2.26e-16 & 4.008e-16 & 0.966 \end{pmatrix}$$

Работа с данными из репозитория German

Процент «значимости»	Дисперсия
1. 11 %	2.72
2. 10 %	2.35
3. 9 %	2.13
4. 8.6 %	2.08
5. 6 %	1.55
6. 6 %	1.47
7. 5.5 %	1.36
8. 5 %	1.25
9. 4.5 %	1.11
10. 4.5 %	1.09
11. 0.5 %	0.11
12. 0.6 %	0.15
13. 0.9 %	0.22
14. 0.95 %	0.23
15. 1.4 %	0.33
16. 3.8 %	0.92
17. 1.9 %	0.46
18. 2 %	0.52
19. 2 %	0.57
20. 2.5 %	0.62
21. 3 %	0.80
22. 2.9 %	0.71
23. 3 %	0.75
24. 3 %	0.74

Можно сделать вывод, что данные плохо коррелированы, но тем не менее важные признаки: № 1, № 2, № 3, № 4.

Работа с данными Iris

Рассмотрим матрицу ковариаций исходных данных (стандартизированных):

$$cov(x) = \begin{pmatrix} 1.006 & -0.12 & 0.88 & 0.82 \\ -0.12 & 1.006 & -0.43 & -0.369 \\ 0.88 & -0.43 & 1.006 & 0.969 \\ 0.82 & -0.369 & 0.969 & 1.006 \end{pmatrix}$$

Можно заметить, что это четырехмерные данные, причем у всех признаков одинаковая дисперсия; первый признак хорошо коррелирован с 3 и 4 признаками; третий хорошо коррелирован с 1 и 4; а четвертый с 1 и 3. Второй признак достаточно независим. Из чего можно предположить, что один из 1, 3, 4 признаков можно будет убрать из рассмотрения.
Математическое ожидание нормированных данных: $\mu \approx (4.458)$.

Центрируем признаки и получаем новые вектора значений Z. Строим по ним матрицу ковариаций:

$$cov(z) = \begin{pmatrix} 2.94 & 7.695e-16 & 8.4e-16 & 2.94e-16 \\ 7.695e-16 & 0.92 & -3.99e-16 & 2.3e-16 \\ 8.4e-16 & -3.99e-16 & 0.15 & -2.7e-16 \\ 2.94e-16 & 2.3e-16 & -2.7e-16 & 0.02 \end{pmatrix}$$

Уже можно заметить, что четвертый признак можно убрать из рассмотрения и при этом не потерять значимой информации. Рассмотрим проценты «важности»:

1. 73 %
2. 22,5 %
3. 4 %
4. 0.5 %

Вклад 4 признака составил всего 0,5 % от суммарной дисперсии, следовательно он точно сольется либо с 3, либо с 1 признаком.
Визуализируем данные:

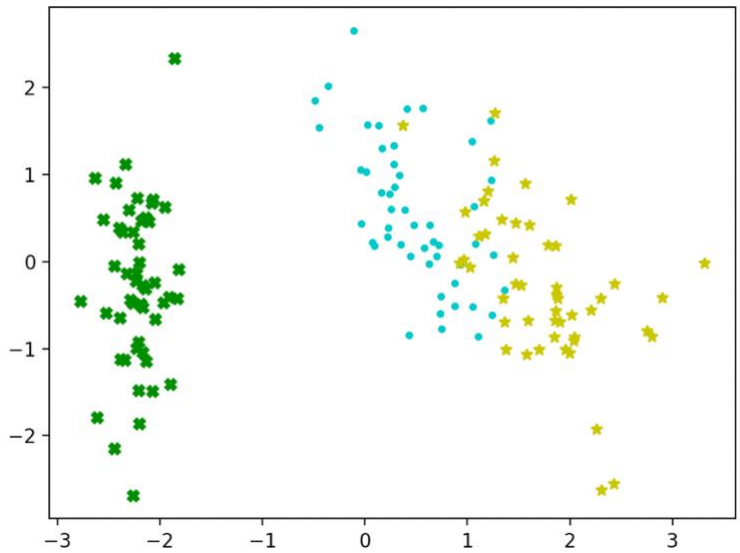

Рисунок 5: визуализация данных Iris

Все вышесказанное подтвердилось.

Выводы

Работа с модельными данными

Метод главных компонент позволяет уменьшить количество рассматриваемых данных в случае «хорошей» коррелированности данных, так как потери информации будут минимальны.

Однако при «плохой» коррелированности данных нет смысла переходить в новую систему координат (проецировать исходные данные).

Работа с данными репозитория German

После нормировки данных и применения метода PCA удалось выделить первые четыре признака, как наиболее значимые. Однако, сложно сказать, что данные хорошо коррелированы, так как процент «значимости» не больше 11%. Это говорит о том, что признаки несут разную информацию, и никакой из них убрать из рассмотрения нельзя.

Работа с данными Iris

Среди четырех признаков удалось выделить три. Второй признак практически не коррелирован с остальными; первый, третий и четвертый имели сильную корреляцию. После проецирования оказалось, что четвертый признак и вовсе не несет новой информации, и его можно выразить через 1 и 3 признаки.

Приложение

Код работы: https://github.com/KateZabolotskih/EconomicDecisionMaking

```
import numpy as np
class PCA:
    def __init__(self, x: '[[],[], ...]'):
        self.x = x
        self.m = len(x)

        self.Xcentered = self._center(x)
        self.covmat = np.cov(x)
        self.eigenvalue, self.eigenvectors = np.linalg.eig(self.covmat)

    def _center(self, x: '[[],[], ...]') -> '[[], [], ...]':
        Xcentered = []
        for i in range(self.m):
            mean = np.mean(x[i])
            Xcentered.append(x[i] - mean)
        return Xcentered

    def Z(self) -> '[[],[], ...]':
        return self.eigenvectors.transpose().dot(self.Xcentered)
```