МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное   
учреждение высшего образования   
«Южный федеральный университет»

Институт математики, механики   
и компьютерных наук им. И. И. Воровича

Кафедра информатики и вычислительного эксперимента

Григорян Георгий Зоргевич

**Использование стресс-функции   
для определения размерности пространства характеристик   
в задачах DATA MINING**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
по направлению подготовки  
02.03.02 – Фундаментальная информатика и информационные технологии

**Научный руководитель** –   
доц., к. ф.-м. н. Нестеренко Виктор Александрович

Допущено к защите:  
заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Михалкович С.С.

**Оглавление**

Оглавление

[Введение 3](#_Toc105517484)

[1.Постановка задачи 3](#_Toc105517485)

[2. Принцип работы алгоритма 4](#_Toc105517486)

[3. Реализация алгоритма 4](#_Toc105517487)

[3.1 Подготовка данных 5](#_Toc105517488)

[3.2 Вычисление ковариационной матрицы 6](#_Toc105517489)

[3.3 Извлечение собственных векторов и собственных чисел 8](#_Toc105517490)

[3.4 Формирование нового вектора фич. 8](#_Toc105517491)

[3.5 Получение нового набора данных 8](#_Toc105517492)

[3.6 Восстановление данных. 9](#_Toc105517493)

[4. Применение Метода Главных Компонент 10](#_Toc105517494)

[5. Использование Стресс-функции для оценки размерности пространства данных. 14](#_Toc105517495)

# Введение

На протяжении многих лет объемы данных, предоставленных людям, стремительно увеличиваются. Из-за этого перед человечеством встает проблема извлечения полезной информации из них. Поэтому на помощь к людям приходит такая технология как Data Mining. Data Mining — это интеллектуальный анализ данных. Данная технология пришла на замену прикладной статистике, следовательно отсюда проистекает изобилие методов и алгоритмов. Сам же термин “Data Mining” часто переводится как добыча данных, извлечение информации. Одной из важных задач в Data Mining является уменьшение размерности.

Для чего же нужна редукция размерности пространства признаков? Во-первых, большое количество признаков требуют большего времени для вычислений. Во-вторых, большие вычисления более ресурсоемкие. В-третьих, в любых данных есть шум, который негативно влияет на обучение какой-либо модели. Кроме того, информацию, представленную в двумерном или трехмерном измерениях, можно легко визуализировать, чем при боле высоких измерениях. Есть множество методов, позволяющих сделать редукцию пространства признаков данных, но в своей работе я опишу и реализую Метод Главных Компонент либо же Анализ Основных Компонент (PCA, Principal Component Analysis).

# 1.Постановка задачи

В выпускной работе требуется выполнить следующее

Основные этапы выполнения задания:

1. Оценка и выбор метода уменьшения размерности.
2. Программная реализация метода Главных Компонент
3. Применение предлагаемого метода
4. Анализ полученных резуельтатов

# 2. Принцип работы алгоритма

Это способ выявления закономерностей в данных и выражения данных таким образом, чтобы подчеркнуть их сходства и различия. Поскольку закономерности может быть трудно найти в данных большой размерности, где роскошь графического представления недоступна, PCA является мощным инструментом для анализа данных. Суть алгоритма состоит в том, что уменьшение количества фич происходит за счет точности новых данных, так как главные компоненты являются линейной комбинацией признаков.

С геометрической точки зрения, главные компоненты представляют собой [Векторы](https://www.helenkapatsa.ru/viektor/) данных, которые объясняют максимальное количество отклонений. Главные компоненты – новые оси, которые обеспечивают лучший угол для оценки данных, чтобы различия между наблюдениями были лучше видны. Поскольку существует столько главных компонент, сколько переменных в наборе, главные компоненты строятся таким образом, что первый из них учитывает наибольшую возможную дисперсию в наборе данных.

# 3. Реализация алгоритма

Вычисление главных компонент может быть сведено к вычислению [сингулярного разложения](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D0%B3%D1%83%D0%BB%D1%8F%D1%80%D0%BD%D0%BE%D0%B5_%D1%80%D0%B0%D0%B7%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) матрицы данных или к вычислению [собственных векторов](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B1%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B2%D0%B5%D0%BA%D1%82%D0%BE%D1%80) и [собственных значений](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B1%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) [ковариационной матрицы](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B2%D0%B0%D1%80%D0%B8%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%82%D1%80%D0%B8%D1%86%D0%B0) исходных данных.  В моем случае алгоритм реализован через нахождение собственных значений и векторов ковариационной матрицы/

## Подготовка данных

Для демонстрации работы алгоритма был сгенерирован data set состоящий из вещественных чисел ([Рис. 1](#_Подготовка_данных)). Размерность пространства данных равнялось 4 признакам, такая размерность взята для проверки редукции к разному количеству главных компонент.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 1.0 | 2.73446908 | 11.01792737 | 98.47615116 |
| 2.0 | 4.35122722 | 12.38619263 | 20.46282367 |
| 3.0 | 7.21132988 | 11.48804931 | 67.17093415 |
| 4.0 | 11.24872601 | 24.10099224 | 77.66304739 |
| 5.0 | 9.58103444 | 30.21117481 | 142.25100662 |
| 6.0 | 12.09865079 | 32.83617975 | 127.60618803 |
| 7.0 | 13.78706794 | 42.4237342 | 199.82062948 |
| 8.0 | 13.85301221 | 51.14366228 | 110.97707692 |
| 9.0 | 15.29003911 | 47.1998583 | 218.21847896 |
| 10.0 | 18.0998018 | 60.19540684 | 269.12150528 |

Рис. 1. Таблица сгенерированных данных.

Для корректной работы PCA необходимо центрировать данные, так как метод очень чувствителен к дисперсиям, т.е. вычесть из каждого значения столбца среднее арифметическое этого столбца (Рис. 2). Таким образом среднее арифметическое центрированных данных будет равняться нулю.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| -4,50000000 | -8,09106677 | -21,28239040 | -34,70063301 |
| -3,50000000 | -6,47430863 | -19,91412514 | -112,71396050 |
| -2,50000000 | -3,61420597 | -20,81226846 | -66,00585002 |
| -1,50000000 | 0,42319016 | -8,19932553 | -55,51373678 |
| -0,50000000 | -1,24450141 | -2,08914296 | 9,07422245 |
| 0,50000000 | 1,27311494 | 0,53586198 | -5,57059614 |
| 1,50000000 | 2,96153209 | 10,12341643 | 66,64384531 |
| 2,50000000 | 3,02747636 | 18,84334451 | -22,19970725 |
| 3,50000000 | 4,46450326 | 14,89954053 | 85,04169479 |
| 4,50000000 | 7,27426595 | 27,89508907 | 135,94472111 |

Рис. 2. Таблица центрированных данных.

## Вычисление ковариационной матрицы

Для начала стоит разобраться что такое ковариация. Ковариация или корреляционный момент — мера зависимости одной случайной величины от другой. В нашем случае формула, по которой вычисляется ковариация значений двух измерений будет выглядеть так:

Рис.3. Формула ковариации двух векторов.

где , среднее арифметическое значений измерений X, Y.

Ковариационная матрица для n-мерного пространства признаков будет выглядеть так:

Рис.4 Ковариационная матрица

где одно из возможных измерений.

Рис.5 Пример ковариационной матрицы для данных с размерностью пространства.

В матрице коэффициентов ковариационной матрицы имеет значение знаки этих коэффициентов. Если знак – это:

* положительное число, то две переменные прямо пропорциональны, то есть второй увеличивается или уменьшается вместе с первым.
* отрицательное число, то переменные обратно пропорциональны, то есть второй увеличивается, когда первый уменьшается, и наоборот.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | 9,16666666 | 14,34448508 | 52,34245948 | 193,94202309 |
|  | 14,34448508 | 23,84180375 | 81,40260965 | 300,65742902 |
|  | 52,34245947 | 81,40260965 | 312,47012667 | 1122,85505928 |
|  | 193,94202309 | 300,65742902 | 1122,85505928 | 5789,75302839 |

Рис. 5. Ковариационная матрица сгенерированных данных.

## 3.3 Извлечение собственных векторов и собственных чисел

Что бы определить Главные Компоненты необходимо извлечь собственные векторы (eigenvectors) и собственные числа (eigenvalues) из ковариационной матрицы. Главная Компонента ­­­­­­­– это новая переменная, комбинированная таким образом, что новые переменные не коррелированы между собой, и основная информация об исходных переменных помещается в первых компонентах. Для извлечения собственных чисел и собственных векторов был применен Метод Якоби – итерационный алгоритм применимый к вещественной симметричной матрице. Собственные векторы дают нам представление о направлении осей, где наблюдается наибольшая дисперсия т.е. большая часть информации, а собственные числа — значения, показывающие величину этой дисперсии у каждого собственного вектора. Таким образом отсортировав собственные векторы по их собственным числам, мы получаем Главные Компоненты в порядке их насыщенности информацией.

## Формирование нового вектора фич.

Для построения новой матрицы фич необходимо взять собственные векторы, обладающие наибольшей дисперсией и сформировать матрицу из этих векторов в столбцах.

где – собственный вектор.

## Получение нового набора данных

Это последний шаг для метода главных компонент. После того, как был сформирован новый вектор фич, необходимо транспонировать его и умножить слева на транспонированный набор исходных данных.

Рис.6. Проекция на ось главных компонент.

Где FeatureVector это транспонированная матрица собственных векторов со значениями в строках отсортированные сверху-вниз по порядку значимости, а MeanedData это транспонированные центрированные начальные данные. Цель этого перемножения переориентировать набор данных с исходной оси, на оси, представленные главными компонентами.

## Восстановление данных.

Проекция данных дает огромные возможности для работы с ней, однако она не дает явного понимания какая информацию она содержит. Для понимания полной картины необходимо восстановить данные. Для этой процедуры все необходимое вычислено: Средние значения векторов признаков, данные спроецированный на оси главных компонент, собственные векторы.

Рис.7. Восстановление данных.

В данном случае восстановленные данные получаются умножением транспонированной матрицы собственных векторов на спроецированные данные справа. После умножения следует построчно прибавить вектор средних (MeanVector).

# Применение Метода Главных Компонент

Для проверки реализации алгоритма было принято решение воспользоваться дата сетом Iris, который представляет себя данные о трех видах Ириса.

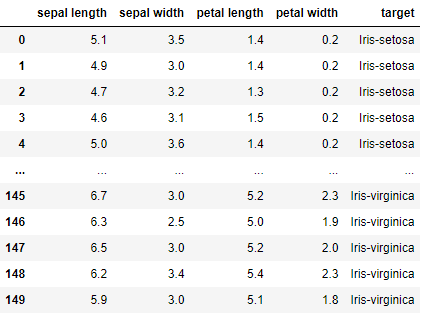


Рис.8. Ирисы Фишера.

Первым делом было принято решение визуализировать данные.

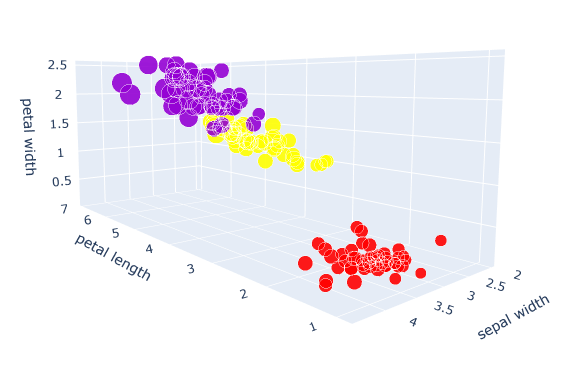


Рис.9. Представление Ирисов Фишера в размерности пространства 4.

На данном графике цветами представлены виды ирисов. Оси X,Y,Z заняты такими признаками как ‘petal length’, ‘sepal width’, ‘petal width’. Признак ‘sepal length’ показывает размер окружности каждой элемента. Как видно из этого графика 4 признака избыточны для визуализации, в добавок они приносят сложность в восприятии графика. Попробуем уменьшить размерность. Нормируем данные.

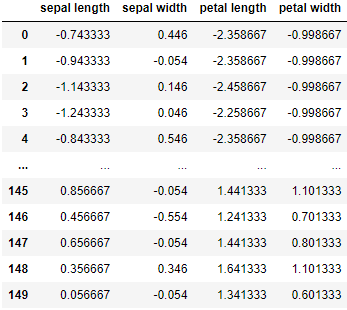


Рис.10. Нормированные данные.

Посчитаем ковариационную матрицу.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | sepal length | sepal width | petal length | petal width |
| sepal length | 0.68569351 | -0.03926846 | 1.27368233 | 0.5169038 |
| sepal width | -0.03926846 | 0.18800403 | -0.32171275 | -0.11798121 |
| petal length | 1.27368233 | -0.32171275 | 3.11317942 | 1.29638747 |
| petal width | 0.5169038 | -0.11798121 | 1.29638747 | 0.58241432 |

Рис.9.Ковариационная матрица.

На ковариационной матрице видны зависимости между признаками, например, при росте параметра ‘sepal length’ уменьшается параметр ‘sepal width’ и наоборот.

Далее на основе ковариационной матрицы следует вычислить собственные векторы и собственные числа. Получаем следующий результат:

Собственные числа:

Собственные векторы:

Основываясь на значениях собственных чисел, мы видим, что 4 собственный вектор несет в себе наибольшую информацию о данных. Отсортируем их в порядке убывания значений собственных чисел. Попробуем уменьшить размерность пространства с 4 до меньших.

Для размерности пространства 3 график будет выглядеть следующим образом:

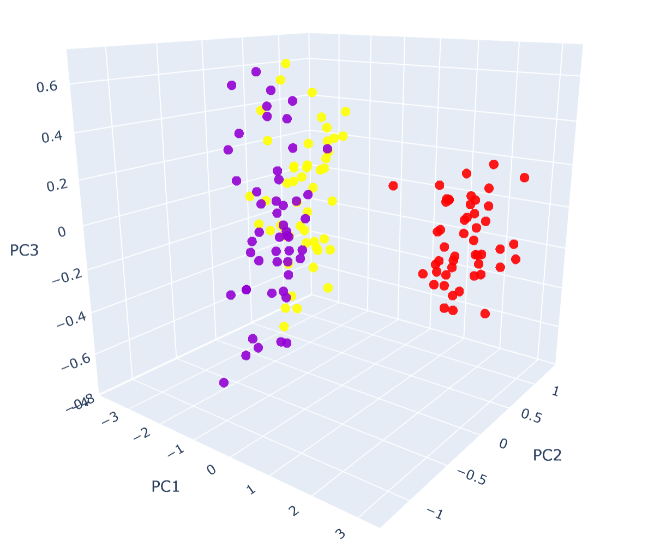


Рис.11. График размерности пространства 3.

Данные представлены в трехмерном пространстве, где осями являются 3 главные компоненты, а цвет отвечает за вид ирисов. Данные стало воспринимать проще, однако до конца не ясны границы между видами. Уменьшим размерность пространства до 2х.

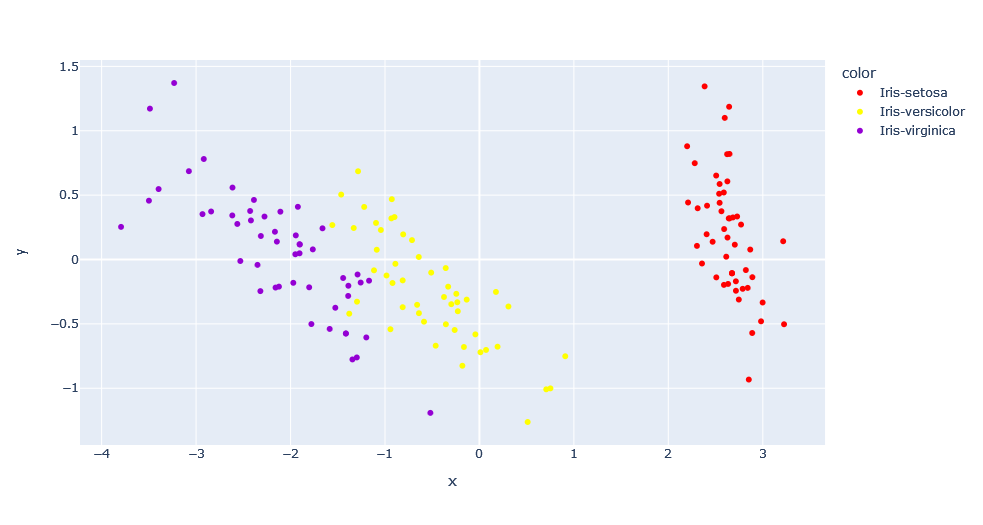


Рис.12. График размерности пространства 2.

Различия между видами ирисов стали более наглядны. Попробуем уменьшить размерность до 1.

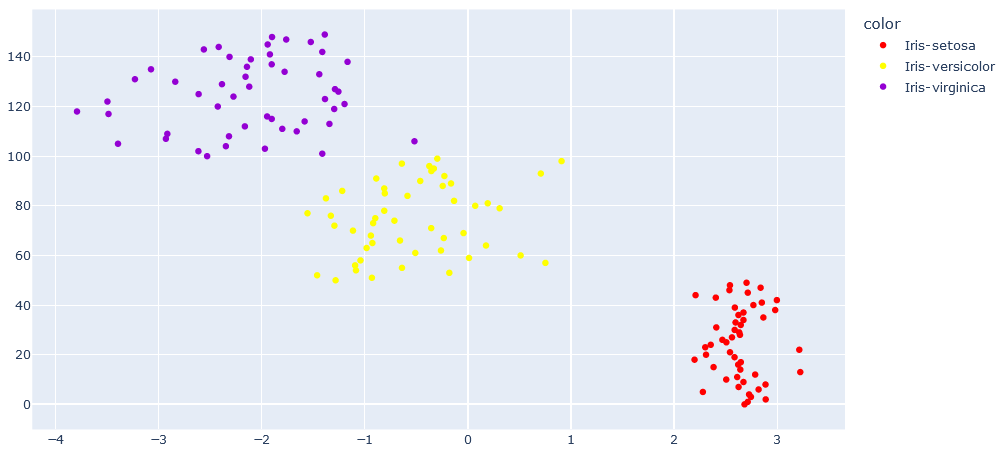
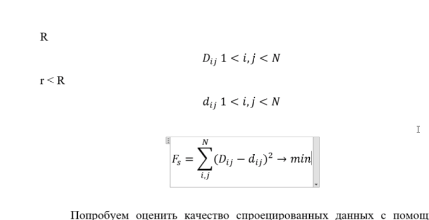


Рис.13. График размерности пространства 1.

В отличии от рисунка 12 на рисунке 13 расстояния между точками намного меньше. И различия между тремя кластерами выражены лучшим образом.

# Использование Стресс-функции для оценки размерности пространства данных.



Попробуем оценить качество спроецированных данных с помощью кластеризации. Для этого используем один из методов кластеризации K-means. Изначально данные (MainData) были кластеризованы по видам ирисов. Применим K-means. Начальная кластеризация отличается от полученной в 16 пунктах. Назовем ее Origin. Кластеризуем данные, которые были получены использованием метода главных компонент. Сравним их со значением origin:

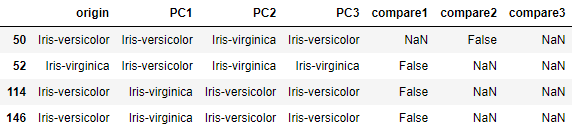


Рис.15. Точки, принадлежащие разным кластерам.

Compare1, compare2, compare3 это значения полученные при сравнении кластеров origin с проекциями данных на оси Главных компонент, размерность которых соответственно равна 1,2,3. Как видно из таблицы между Origin и PC1 3 промаха, а между Origin и PC2 промах 1.

Оценим данные, полученные проецированием на оси главных компонент. Для этого используем индекс Rand. Индекс оценивает, насколько много из тех пар элементов, которые находились в одном классе, и тех пар элементов, которые находились в разных классах, сохранили это состояние после кластеризации алгоритмом.

Элементы принадлежат одному кластеру и одному классу — TP

Элементы принадлежат одному кластеру, но разным классам — FP

Элементы принадлежат разным кластерам, но одному классу — FN

Элементы принадлежат разным кластерам и разным классам — TN

Отсюда получим различные значения для разных пар кластеризаций:

Rand (Origin, PC1) = 0.9739597315436241

Rand (Origin, PC2) = 0.9911409395973154

Rand (Origin, PC3) = 1.0

Rand (MainData, PC1) = 0.8987919463087248

Rand (MainData, PC2) = 0.8737360178970918

Rand (MainData, PC3) = 0.8797315436241611

Как видно из значений наибольшее сходство между Origin и PC3, а также MainData и PC1. Визуализируем кластеры полученные на основе PC1, PC2 и PC3. Построим график для 3х компонент.

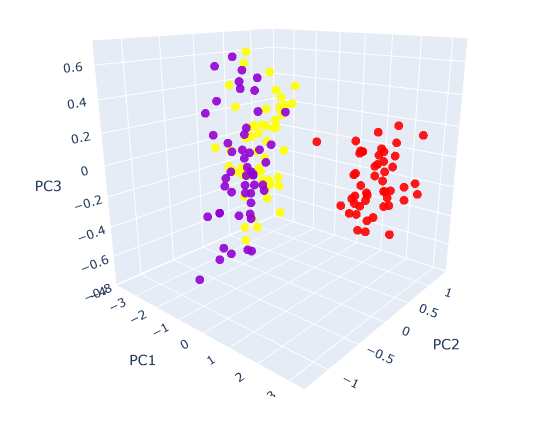


Рис.16. Кластеры для размерности пространства 3.

Разница между рисунком 16 и 11 не видна, однако разница между начальным кластером и полученным 12%.

Построим график для 2х компонент:

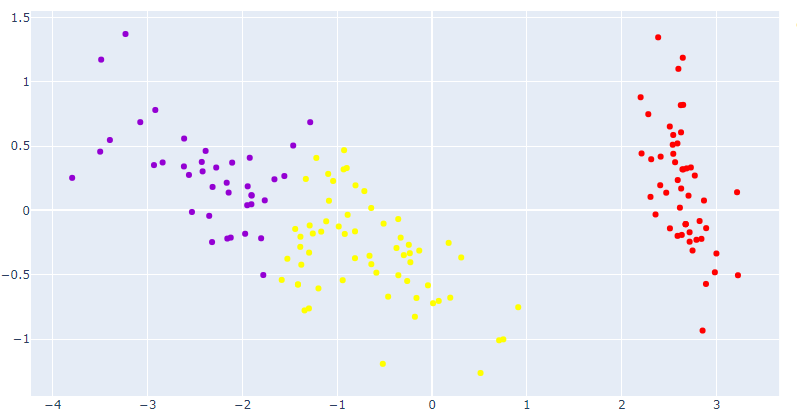


Рис.17. Кластеры для размерности пространства 2.