**МИНОБРНАУКИ РОССИИ**
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**
**Кафедра МО ЭВМ**

# ОТЧЕТ

**по лабораторной работе №2**

**по дисциплине «Машинное обучение»**

**Тема: Метод главных компонент и факторный анализ**

Выполнил: Сергеев А.Д.

Факультет: ФКТИ

Группа: 8304

Преподаватель: Жангиров Т.Р.

Санкт-Петербург

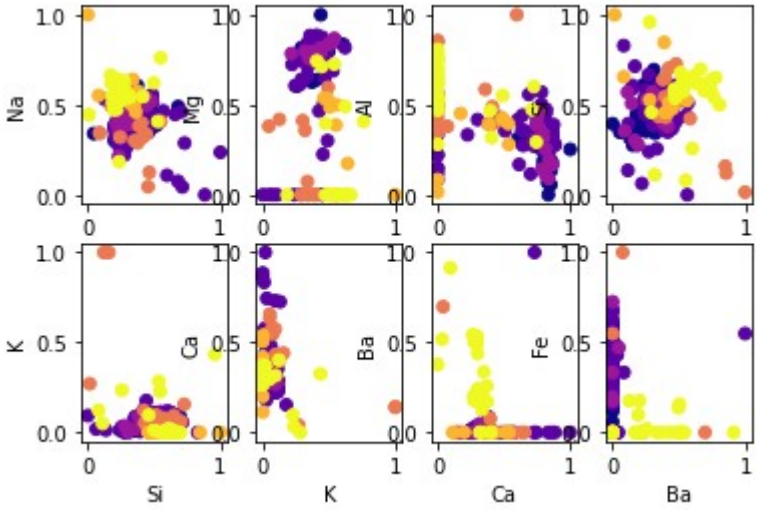
2021

## 1) Загрузка данных

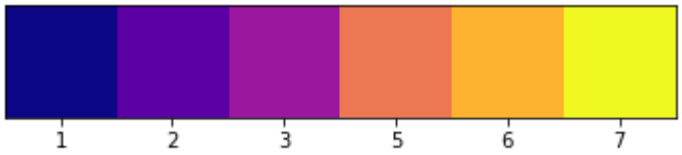
1 Данные были загружены в датафрейм и разделены на описательные признаки (*Ri*, *Na*, *Mg*, *Al*, *Si*, *K*, *Ca*, *Ba*, *Fe*) и признак, отображающий класс (*Type*).

2 Данные были нормированы к интервалу *[0; 1]* при помощи *minmax_scale*.

3 Были построены диаграммы рассеяния для пар признаков.




4 Было установлено соответствие между цветом точки и классом признака (для однозначности использована цветовая схема *plasma*).



## 1) Метод главных компонент

Метод главных компонент представлякт из себя способ линейного уменьшения размерности и проецирования их в пространство более низкой размерности. Входные данные центрируются (вычитанием из среднего), но не масштабируются для каждого объекта перед применением *SVD*.

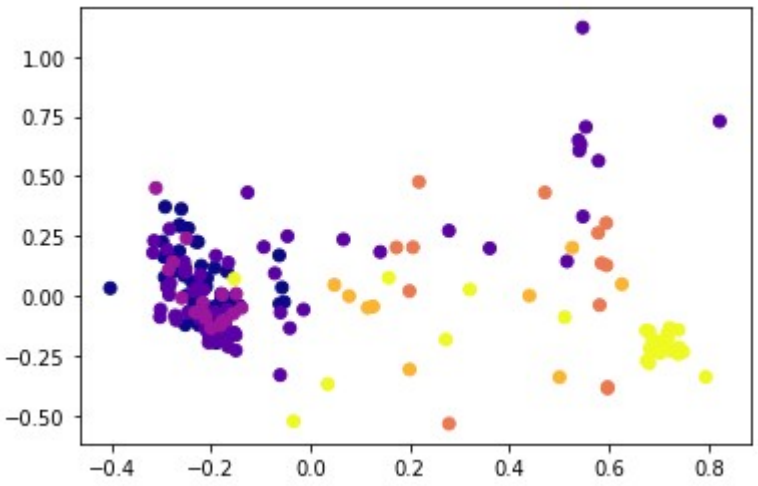
PCA линейно преобразует данные в новые признаки, которые не коррелируют друг с другом.

1 С использованием метода главных компонент (*PCA*) было проведено понижение размерности пространства до размерности 2.

2 Были выведены значения объясненной дисперсии в процентах и собственные числа, соответствующие компонентам.

| | component 1 | component 2 |
|---|---|---|
| **explained variance** | 0.454296 | 0.179901 |
| **eigenvalues** | 5.104931 | 3.212457 |

3 Была построена диаграмма рассеяния данных после применения метода главных компонент.

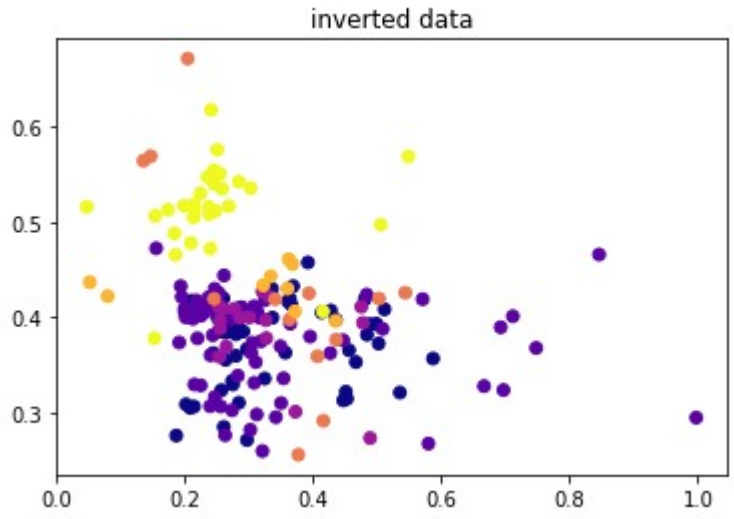



4 По диаграмме рассеивания видно, что линейной зависимости данных нет.

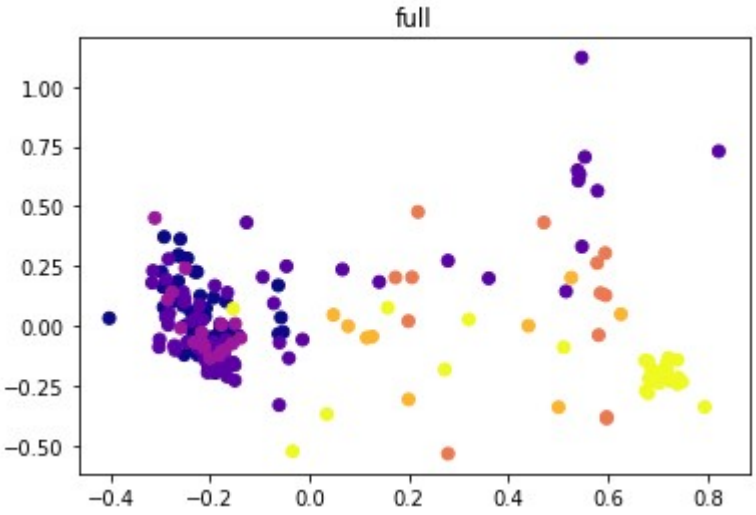
5 При помощи изменения количества компонент, было определено количество, при котором компоненты объясняют не менее 85% дисперсии данных (4).

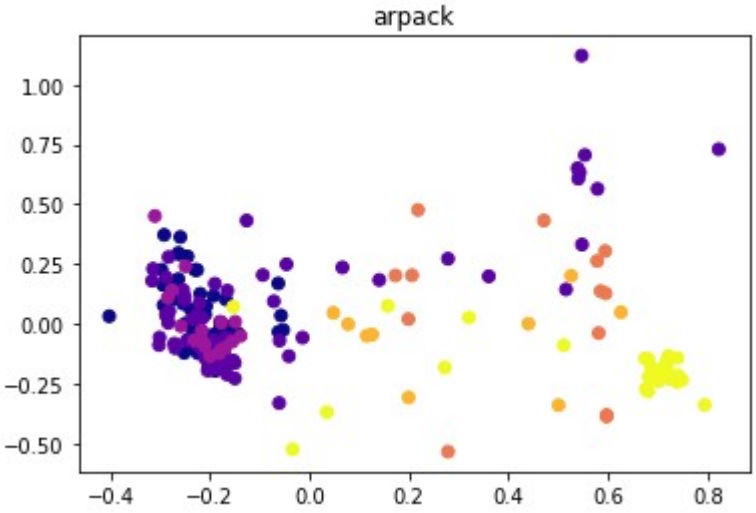
| | explained variance % |
|---|---|
| **0** | 0.000000 |
| **1** | 0.454296 |
| **2** | 0.634197 |
| **3** | 0.760691 |
| **4** | 0.858670 |
| **5** | 0.927294 |
| **6** | 0.969435 |
| **7** | 0.995533 |
| **8** | 0.999861 |
| **9** | 1.000000 |

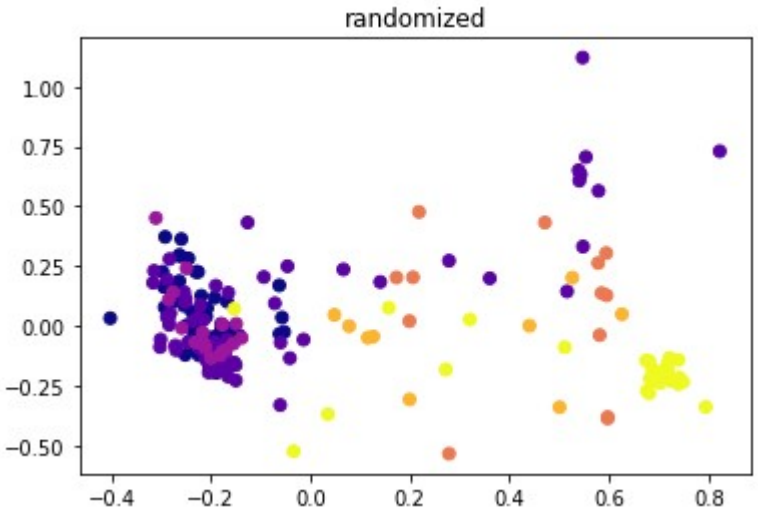
6 С использованием метода *inverse_transform* данные были восстановлены данные, а также выполнено сравнение с исходными данными. Была построена диаграмма рассеяния после восстановления данных. Можно сделать вывод о том, что, при размере объясненной дисперсии в 85%, различия между исходными и восстановленными данными достаточно малы.



7 Было выполнено исследование метода *PCA* при различных значениях *svd_solver*. Из построенных диаграмм рассеивания видно, что на данном наборе данных значение *svd_solver* почти не влияет на конечный результат. При значении ‘*full*’ выполняется разложение алгоритмом *LAPACK*, а затем определяется количество компонент. При значении ‘*arpack*’ выполняется разложение алгоритмом *ARPACK* для заданного числа компонент. При значении ‘*randomized*’ выполняется разложение по методу Халко.



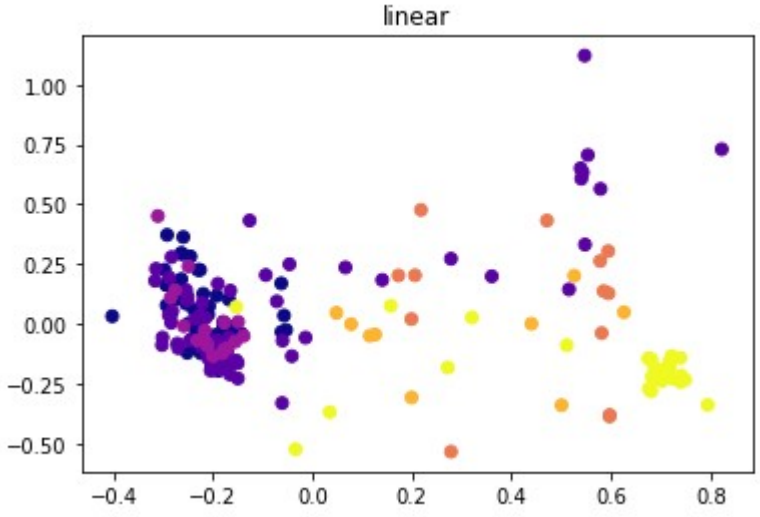


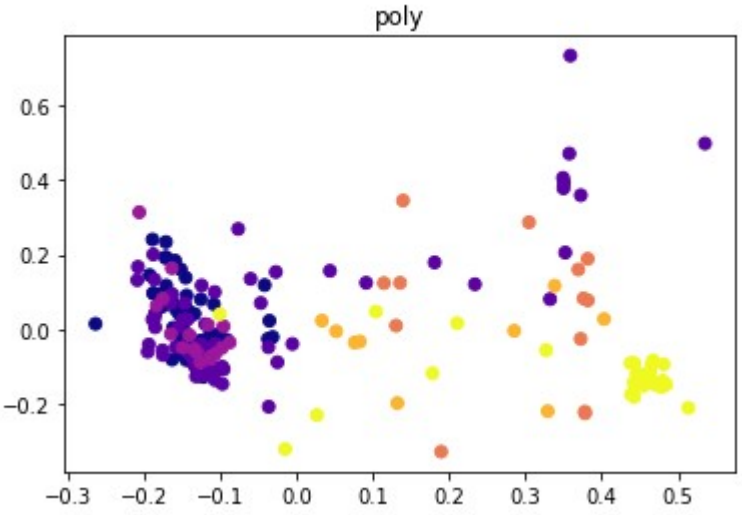

**2) Модификации метода главных компонент**

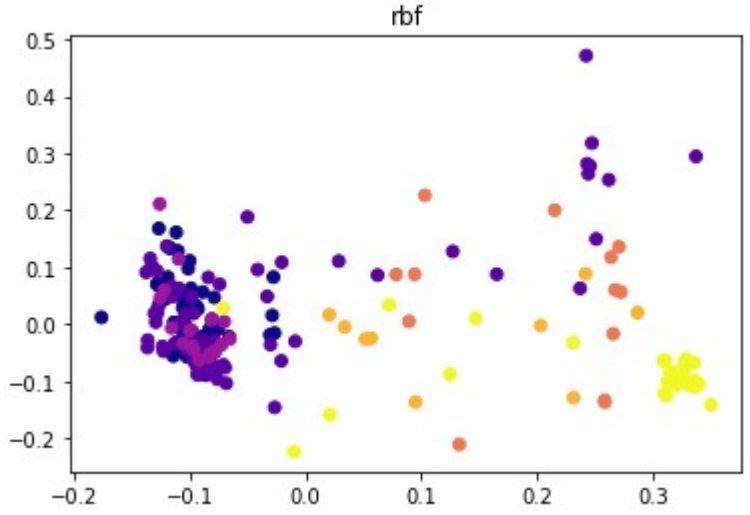
Алгоритм *KernelPCA* производит уменьшение нелинейной размерности за счет использования ядер. Ядро может быть '*линейное*', '*poly*', '*rbf*', '*сигмоидное*', '*косинусное*', '*предварительно вычисленное*', по умолчанию используется *линейное* ядро.

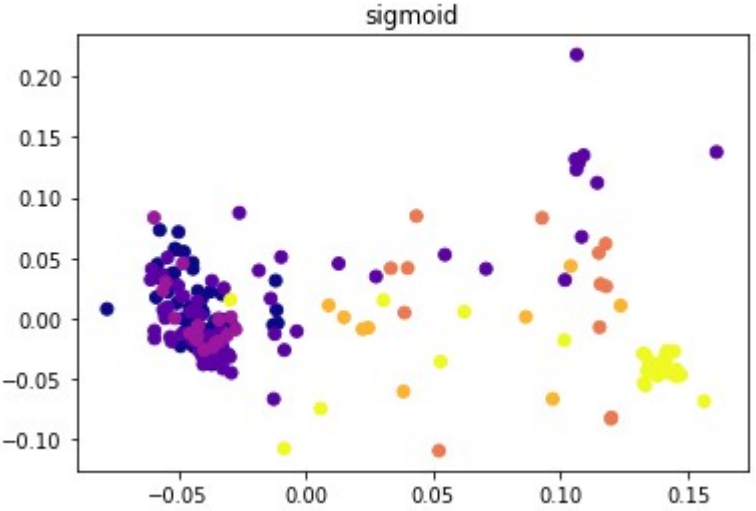
1 По аналогии с *PCA* было проведено исследование *KernelPCA* при различных параметрах *kernel*. В результате исследования можно сделать вывод о том, что собственные числа и количество компонент отличаются из-за разных функций ядра, однако значения объясненной дисперсии различаются в небольших пределах от 1 до 2%.

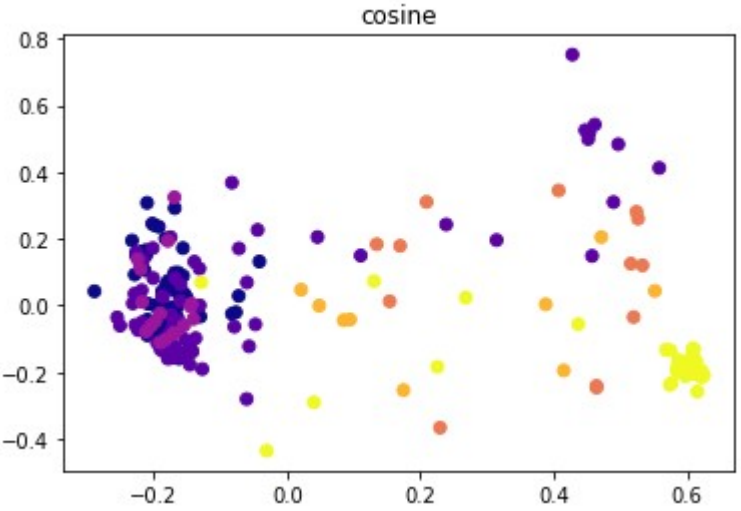
| | components | inverse delta |
|---|---|---|
| **linear** | 9.0 | 4.577652e-17 |
| **poly** | 177.0 | 3.051952e-01 |
| **rbf** | 201.0 | 3.051932e-01 |
| **sigmoid** | NaN | NaN |
| **cosine** | 9.0 | 5.897440e-02 |


linear
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8


poly
0.6
0.4
0.2
0.0
−0.2
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5


rbf
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.2
−0.1
0.0
0.1
0.2
0.3


sigmoid
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.05
0.00
0.05
0.10
0.15

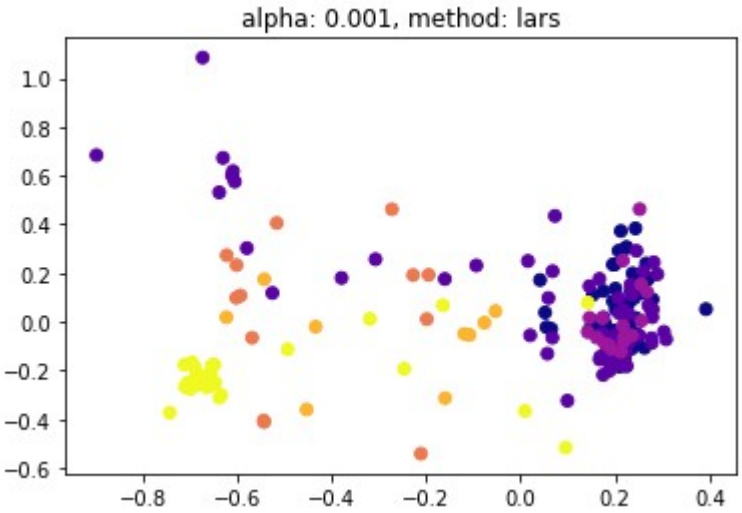

cosine
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.2
0.0
0.2
0.4
0.6

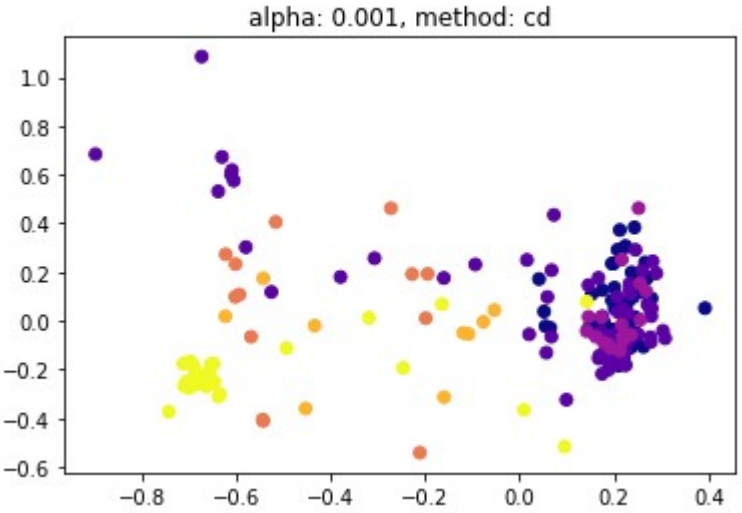
2 *KernelPCA* ведет себя также, как *PCA* при использовании *linear* ядра.

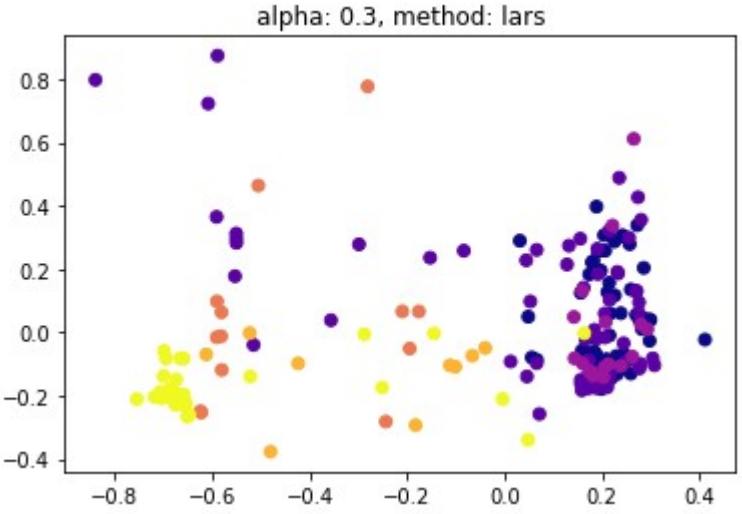
3 Аналогично был исследован *SparsePCA*.

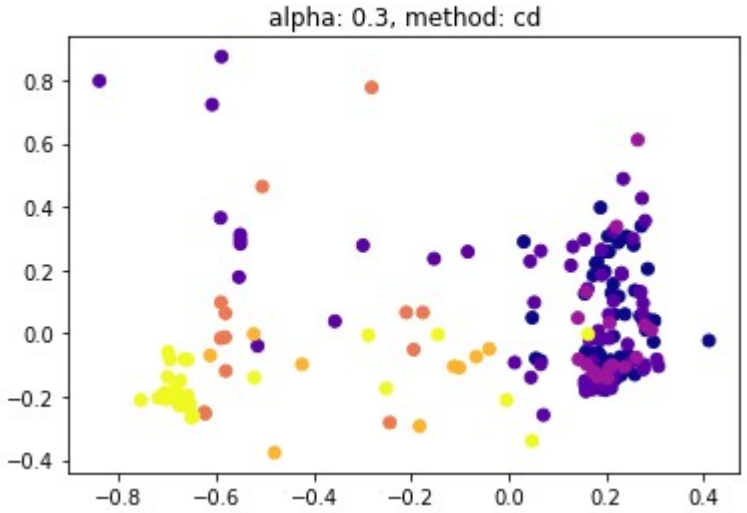
Этот алгоритм находит набор разреженных компонентов, которые могут оптимально реконструировать данные.

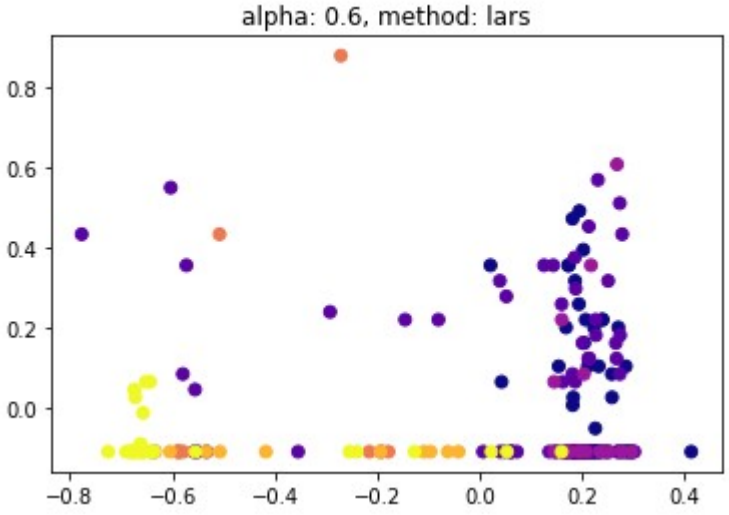
| | components | iterations |
|---|---|---|
| **a: 0.001, m: lars** | 9 | 1000 |
| **a: 0.001, m: cd** | 9 | 1000 |
| **a: 0.3, m: lars** | 9 | 15 |
| **a: 0.3, m: cd** | 9 | 15 |
| **a: 0.6, m: lars** | 9 | 6 |
| **a: 0.6, m: cd** | 9 | 8 |
| **a: 1, m: lars** | 9 | 5 |
| **a: 1, m: cd** | 9 | 7 |

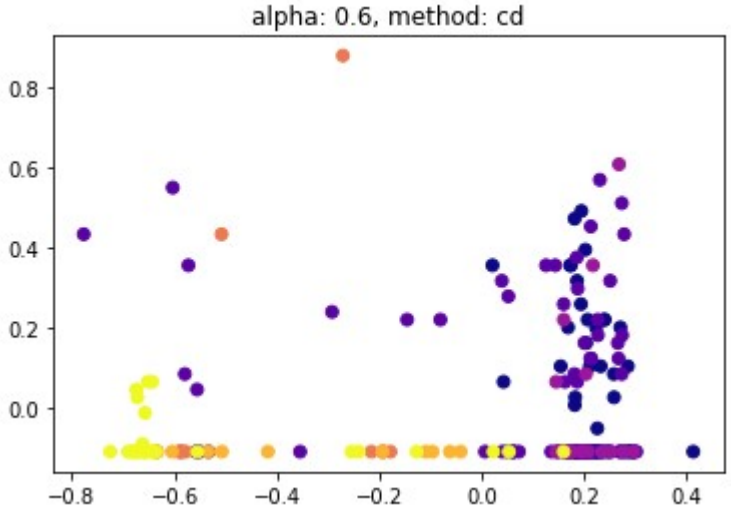


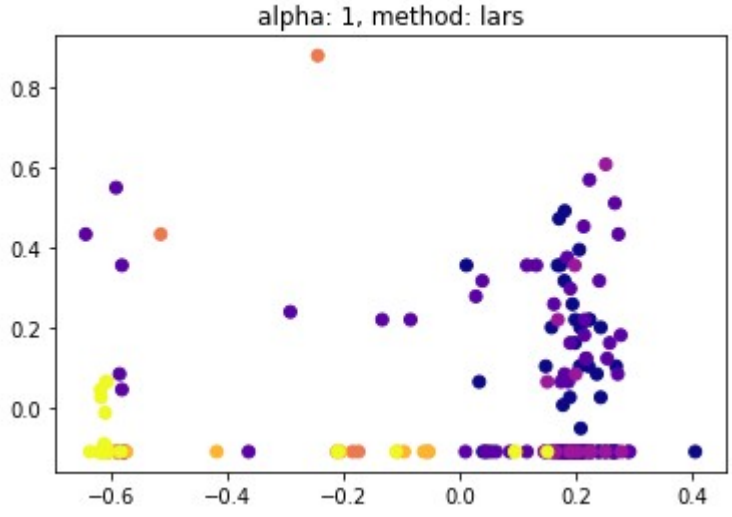

alpha: 0.001, method: cd
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4

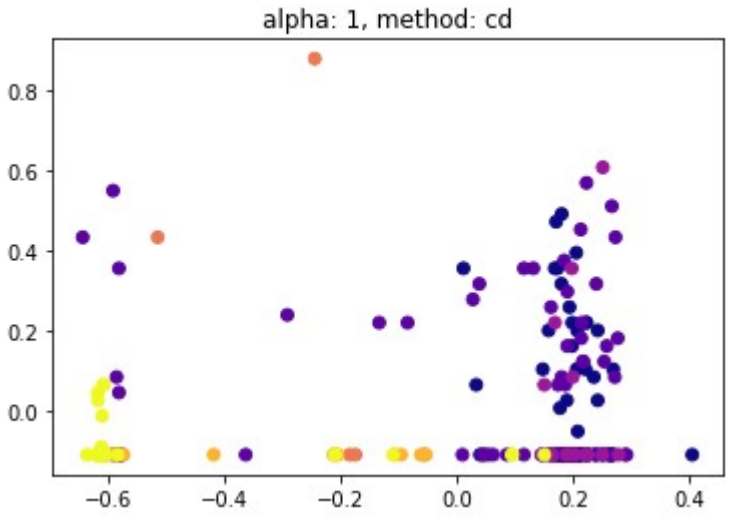

alpha: 0.3, method: lars
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4


alpha: 0.3, method: cd
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4


alpha: 0.6, method: lars
0.8
0.6
0.4
0.2
0.0
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4


alpha: 0.6, method: cd
0.8
0.6
0.4
0.2
0.0
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4


alpha: 1, method: lars
0.8
0.6
0.4
0.2
0.0
−0.6
−0.4
−0.2
0.0
0.2
0.4




4 Параметр *alpha* контролирует разреженность данных, а параметр *method* — метод работы алгоритма. Понижая *alpha* до 0, мы приблизим результат работы алгоритма к результату работы обычного *PCA*. Влияние параметра *method* на результат работы алгоритма для данного габора данных несущественно.

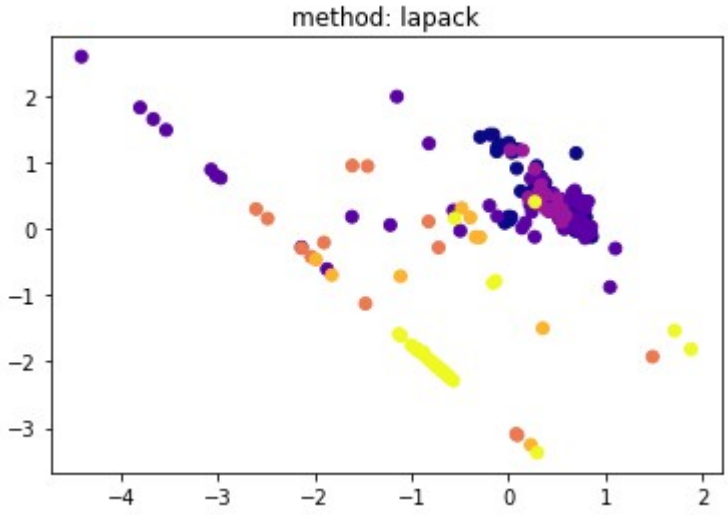
### 3) Факторный анализ

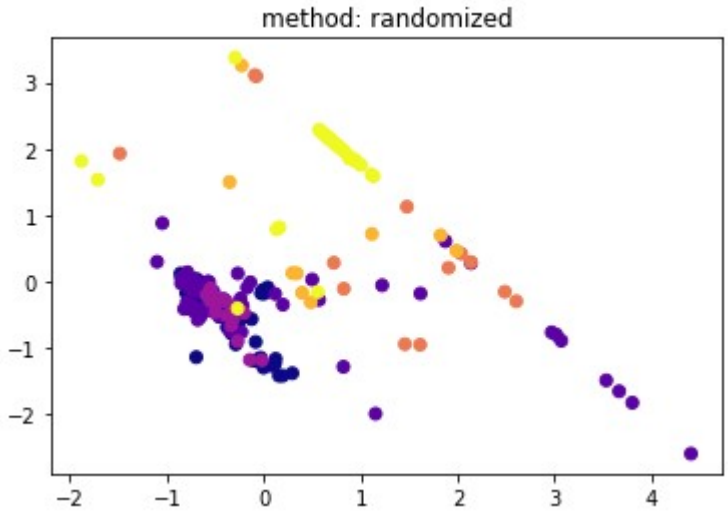
Предполагается, что наблюдения вызваны линейным преобразованием неких факторов и добавлением гауссовского шума. Метод оценивает вероятность преобразования скрытых признаков в наблюдаемые.

1 Было проведено понижение размерности с импользованием факторного анализа *FactorAnalysis*.

| | components | iterations |
|---|---|---|
| **lapack** | 9 | 2 |
| **randomized** | 9 | 2 |

2 Были построены диаграммы рассеяния для двух возможных методов применения факторного анализа: *lapack* и *randomized*.




3 Компонент *PCA* представляет собой линейную комбинацию наблюдаемой переменной. В свою очередь в *FactorAnalysis* наблюдаемые переменные представляют собой линейные комбинации ненаблюдаемых переменных или фактора. Также можно сделать вывод, что, если *PCA* - это метод уменьшения размерности, то *FactorAnalysis* фокусируется на поиске скрытых переменных. Влияние параметра *method* на результат работы алгоритма для данного габора данных несущественно.

## Выводы

В процессе выполнения лабораторной работы были изучены методы понижения размерности и факторного анализа данных из библиотеки *Scikit Learn*.

Для *PCA* (метод главных компонент) был выявлен факт того, что количество компонент напрямую влияет на процент объяснённой дисперсии.

*KernelPCA* применяется для снижения нелинейной размерности за счет использования различных ядер. Может использоваться для поиска нелинейных зависимостей в данных. Для данных, рассмторенных в данной лабораторной работе, смена ядра на результат практически не влияла.

*SparsePCA* находит набор разреженных компонентов, которые могут оптимально восстановить данные. Степень разреженности регулируется коэффициентом *alpha*.

Были изучены отличия факторного анализа от метода главных компонент. Основное преимущество факторного анализа перед ним *PCA* заключается в том, что он может независимо моделировать дисперсию во всех направлениях входного пространства.