Матричные разложения и их применения
(Часть 1, SVD)

Бурштеин Денис

МОП 162

Задача матричного разложения

- Пусть дана выборка из n объектов с m признаками: $X \in Mat_{n \times m}$
- В случае, когда размерность пространства признаков m велика, хотелось бы понизить эту размерность.
- Решение – попробовать найти базис из k объектов (k < m):

$h_1, ..., h_k \in \mathbb{R}^m$ (эти объекты не обязательно должны входить в выборку).

- Каждый объект из выборки представляется в виде линейной комбинации этих базисных объектов:

$$x_i = \sum_{j=1}^{k} w_{ij} h_j$$

Задача матричного разложения

- В матричном виде: $X_{n \times m} = W_{n \times k} H_{k \times m}$
- Строки матрицы W - это новые, k-размерные признаки.
- Строки матрицы H - это набор наиболее «характерных» объектов, через которые выражается вся выборка.
- Поиск таких двух матриц - задача матричного разложения.

$$||X - WH|| \to min$$

- Как норму можно взять норму Фробениуса, т.е. корень из суммы квадратов элементов матрицы.

Примеры интерпретации разложения

- Каждый объект – фотография лица, число признаков равно числу пикселей, значение каждого признака – интенсивность соответствующего пикселя.
- Если матрицы W и H неотрицательные, то строки матрицы H - это набор «базисных лиц».
- Как правило, «базисные лица» - это изображения носа, губ, глаз и других частей лица.

Примеры интерпретации разложения

- Каждый объект – текстовый документ, значение каждого признака – количество соответствующего этому признаку слова из словаря.
- Также, если обе матрицы будут неотрицательными, а сумма элементов в каждой их строке будет равна единице, то мы получаем так называемое «тематическое разложение».
- Базисные объекты – распределения соответствующих слов в одной из тем, каждая строка матрицы W - распределение тем в соответствующем документе.

Примеры интерпретации разложения

- Пусть есть сервис, в котором пользователи ставят оценки фильмам.
- Каждый объект – пользователь, признаки – оценки фильмам, признаков столько, сколько фильмов.
- Тогда базисные объекты – это жанры, а строки матрицы W - это векторы интересов пользователей к этим жанрам.

$$x_{ij} = < w_i, h^j >$$

< > - скалярное произведение, нижний индекс означает строку, верхний – столбец.

- Это типичный подход к задачам построения рекомендаций.

SVD (Singular value decomposition)

- Разложение матрицы в произведение трёх

$$X_{n\times m} = U\Sigma V^T \;\; U \in Mat_{n\times n} \; V \in Mat_{m\times m} \; \Sigma \in Mat_{n\times m}$$

- Столбцы матрицы U - это собственные векторы матрицы XX^T
- Столбцы матрицы V - это собственные векторы матрицы X^TX
- Σ - диагональная матрица, с сингулярными числами на диагонали

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_r, r = rank(X)$$

- Сингулярные числа – корни из собственных чисел матрицы X^TX
- U, V - ортогональные, т.е. их обратные матрицы равны транспонированным

Визуализация геометрического смысла SVD

Применение SVD

- Очень важное применение SVD – приближение матрицей меньшего ранга.
- Теорема Эккарта-Янга: $X_k = U_k \Sigma_k (V_k)^T$, U_k и V_k - отброшены все столбцы кроме первых k, Σ_k - угловой минор размера k. Это приближение матрицы X матрицей ранга k лучшее в том смысле, что Фробениусова норма разности матриц $X - X_k$ минимальна. (Получили решение задачи матричного разложения)
- Благодаря этому свойству, SVD применяется во многих областях: в обработке сигналов, сжатии данных, методе главных компонент и прочих областях.

Существующие реализации

- SVD реализован в большинстве математических пакетов. Например, в нашем любимом NumPy: numpy.linalg.svd

Метод главных компонент (PCA - principal component analysis)

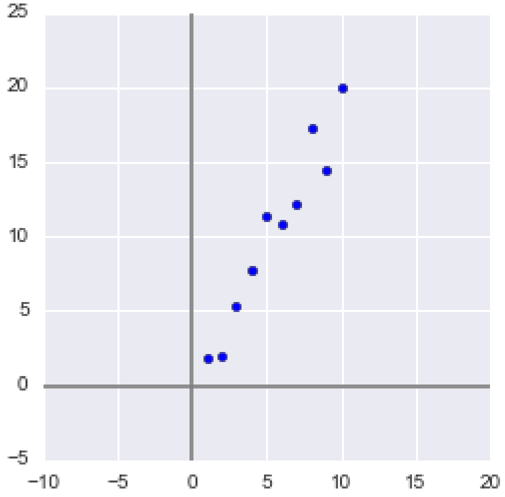
- Метод главных компонент – один из основных методов в анализе данных и машинном обучении.
- Позволяет уменьшить размерность данных, потеряв наименьшее количество информации.
- Иногда метод главных компонент называют преобразованием Кархунена — Лоэва или преобразованием Хотеллинга.

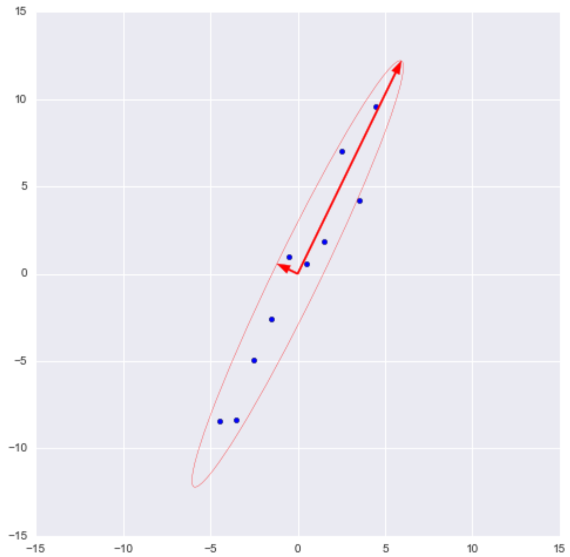
Постановка задачи PCA

- Найти подпространство меньшей размерности, такое, что в ортогональной проекции на него разброс данных максимален, т.е. проекция сохраняет максимально возможную информацию о дисперсии.
- Также, метод позволяет переходить к такой системе координат, что корреляции между отдельными координатами равны нулю.

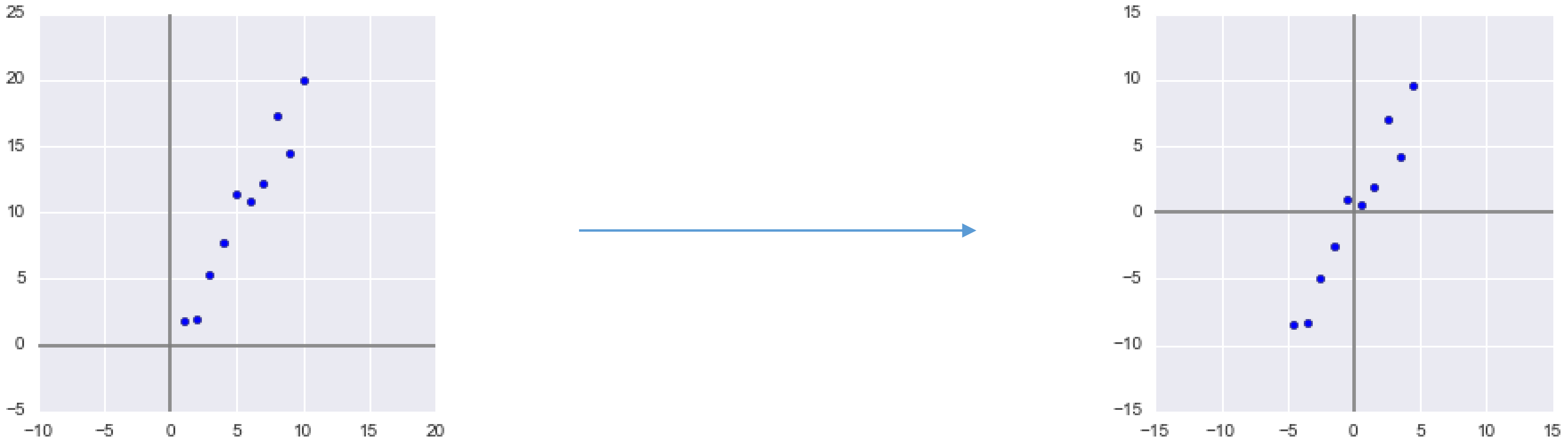
Что хотим, наглядно

- Пусть есть матрица X объекты-признаки с n объектов (в примере ниже n = 10) и m признаков (в примере ниже m = 2).
- Хотим перейти к такой системе координат, что корреляции между различными признаками равны нулю, а проекция на новые оси сохраняла как можно большую информацию о рассеянии (дисперсии).





- Для начала, вычтем средние значения, чтобы центрировать наше распределение.

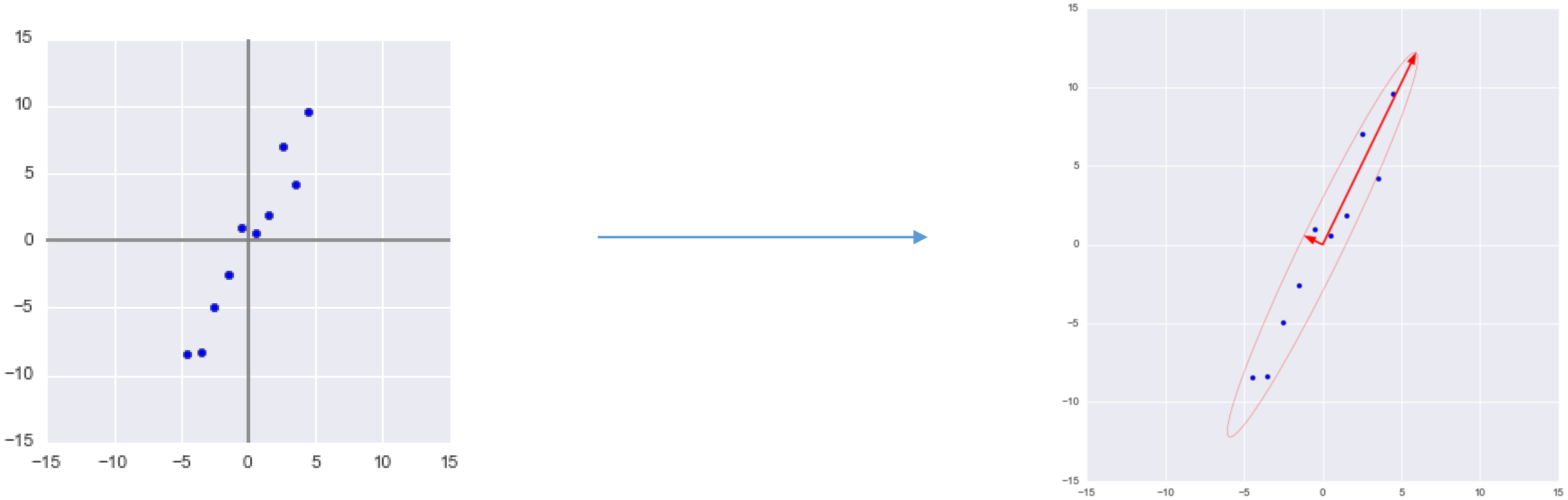


- Далее, нам потребуется ковариационная матрица наших данных.

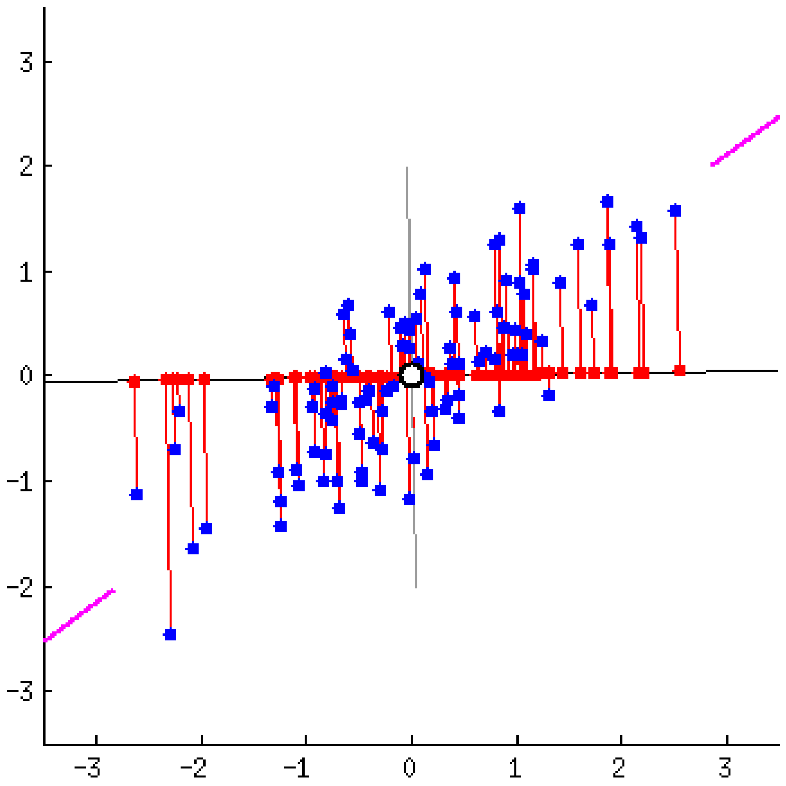
$$C = \frac{1}{n-1} X^T X$$

- Хотим, чтобы дисперсия проекции на первую главную компоненту была максимальной.
- Доказано, что такая главная компонента – это собственный вектор матрицы C , собственное значение которого наибольшее.

- Таким образом, ортонормированный базис из собственных векторов матрицы C, упорядоченный по не возрастанию соответствующих собственных значений, будет искомой системой координат, в которой проекция на первые главные компоненты будет иметь максимальную дисперсию, а также C будет диагональной.



Красные векторы и есть собственные векторы матрицы ковариаций. Тот, который длиннее – первая главная компонента, короткий – вторая. Видно, что спроецировав наши точки на первую компоненту мы сохраняем почти всю информацию о дисперсии, а спроецировав на вторую наоборот теряем почти всю информацию о дисперсии. Также стоит заметить, что собственные значения C - это и есть дисперсии проекций на соответствующие главные компоненты.



3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

Выбор главных компонент

- Для сохранения как можно большего количества информации, надо брать проекцию на какие-то первые k главных компонент.
- k зависит от того, какая точность нам нужна.

Выбор главных компонент

- Как мы уже заметили, собственные значения матрицы ковариаций – это дисперсии проекций.
- Сложив последние m-k значений, мы получаем остаточную дисперсию, т.е. не объяснённую в случае проекции на первые k главных компонент. Сумма первых k – это объяснённая дисперсия.

Выбор главных компонент

- $\delta_k^2 = \dfrac{\lambda_{k+1} + \lambda_{k+2} + ... + \lambda_n}{\lambda_1 + \lambda_2 + ... + \lambda_n}$ - доля необъяснённой дисперсии.

- По относительной ошибке δ_k и определяется k, т.е. на сколько компонент можно спроецировать данные для достижения требуемой точности.

Получение проекции данных

- Пусть V - это матрица, в которой по столбцам записаны найденные главные компоненты.
- Пусть k – это выбранное количество главных компонент.
- Проекция данных на первые k главных компонент:

$X_k = XV_k$, где V_k - это первые k столбцов матрицы V.

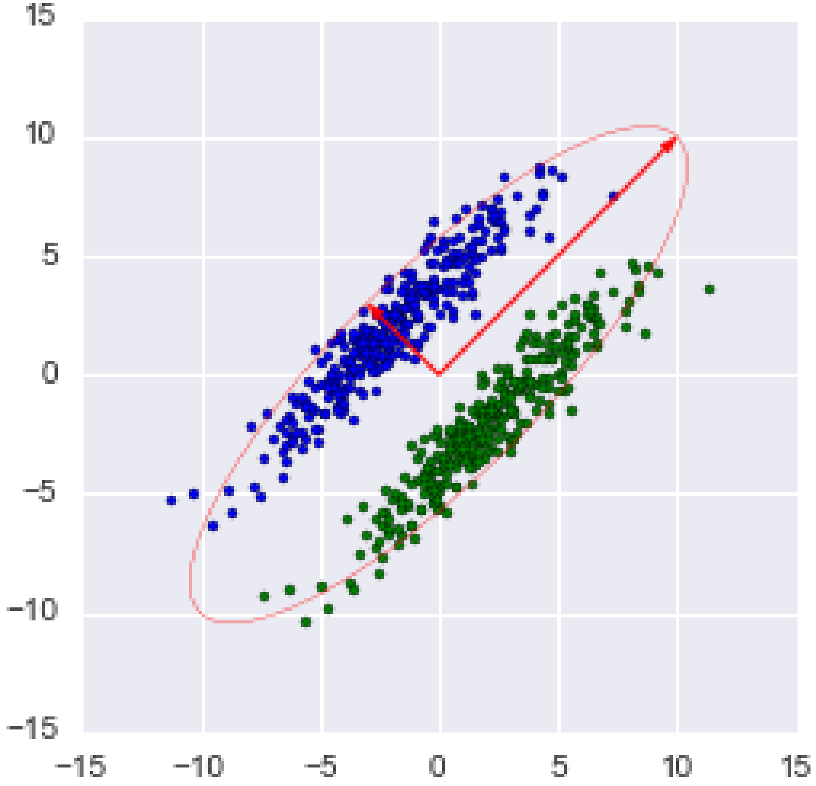
(Понятно, что такого же результата можно добиться, использовав полную матрицу V и оставив у полученной матрицы данных только первые k столбцов)

Решение с помощью SVD

- SVD: $X = U\Sigma V^T$
- Тут V - это базис из собственных векторов матрицы $X^T X$, а это и есть матрица ковариаций с точностью до домножения на константу. (ортонормированность базиса и упорядоченность такие же, как в методе главных компонент, по построению разложения).
- Получаем: $X_{new} = XV = U\Sigma V^T V = U\Sigma$
- Последний переход справедлив, потому что V - ортогональная матрица.
- Таким образом, мы свели PCA к поиску сингулярного разложения матрицы.

Нестандартный случай

- Не всегда требуется сохранить большую дисперсию при проецировании, например в случае задачи классификации:



Некоторые области применения PCA

- Визуализация данных
- Компрессия изображений и видео
- Подавление шума на изображениях
- Эконометрика
- Социология

Плюсы и минусы

- +:
- PCA применим всегда.
- Наличие готовой реализации, например sklearn.decomposition.PCA
- -:
- Не всегда можно добиться требуемой точности, например когда данные распределены как какая-то кривая, сложно расположенная в пространстве.

Источники

- 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
- https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%B3%D0%BB%D0%B0%D0%B2%D0%BD%D1%8B%D1%85_%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82
- https://habr.com/post/304214/
- https://www.coursera.org/lecture/unsupervised-learning/matrichnyie-razlozhieniia-DUBp9
- http://www.machinelearning.ru/wiki/images/b/bb/Sem08_factorizations.pdf