Лекция 7
Многоклассовая классификация. Отбор признаков и методы снижения размерности

Кантонистова Е.О.

ВШЭ, 2022

ПЛАН ЛЕКЦИИ

1. Задачи многоклассовой классификации
2. Методы отбора признаков
3. Линейные методы снижения размерности

МНОГОКЛАССОВАЯ КЛАССИФИКАЦИЯ

ПОДХОД ONE-VS-ALL

Решаем задачу классификации на K классов.

- Обучим K бинарных классификаторов $b_1(x), \dots, b_K(x)$, каждый из которых решает задачу: ***принадлежит объект x к классу k_i или не принадлежит?***

Например, линейные классификаторы будут иметь вид

$$b_k(x) = sign((w_k, x))$$

ПОДХОД ONE-VS-ALL

Решаем задачу классификации на K классов.

- Обучим K бинарных классификаторов $b_1(x), \dots, b_K(x)$, каждый из которых решает задачу: *принадлежит объект x к классу k_i или не принадлежит?*

Например, линейные классификаторы будут иметь вид

$$b_k(x) = sign((w_k, x))$$

- Тогда в качестве итогового предсказания будем выдавать **класс самого уверенного классификатора:**

$$a(x) = \underset{k \in \{1,\dots,K\}}{arg\max}((w_k, x))$$

ПОДХОД ONE-VS-ALL

Решаем задачу классификации на K классов.

- Обучим K бинарных классификаторов $b_1(x), \dots, b_k(x)$, каждый из которых решает задачу: *принадлежит объект x к классу k_i или не принадлежит?*

Например, линейные классификаторы будут иметь вид

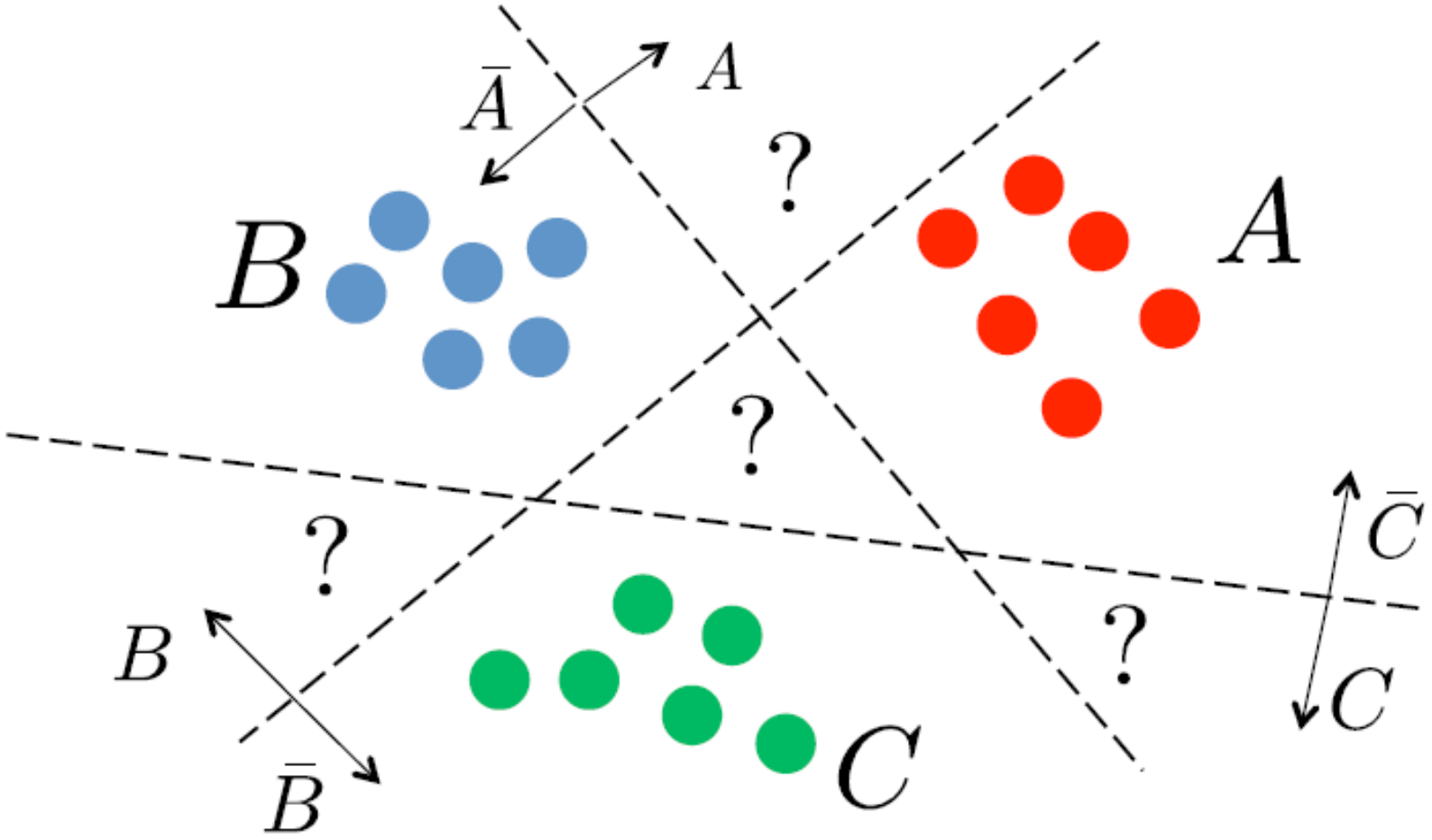
$$b_k(x) = sign((w_k, x))$$

- Тогда в качестве итогового предсказания будем выдавать класс самого уверенного классификатора:

$$a(x) = \underset{k \in \{1,\dots,K\}}{arg\max}((w_k, x))$$

- Предсказания классификаторов могут иметь разные масштабы, поэтому сравнивать их некорректно.

ПОДХОД ONE-VS-ALL



ПОДХОД ALL-VS-ALL

- **Для каждой пары классов i и j обучим бинарный классификатор $a_{ij}(x)$, который будет предсказывать класс i или j**

(если всего K классов, то получим C_K^2 классификаторов).

Каждый такой классификатор будем обучать только на объектах классов i и j.

ПОДХОД ALL-VS-ALL

- Для каждой пары классов i и j обучим бинарный классификатор $a_{ij}(x)$, который будет предсказывать класс i или j

(если всего K классов, то получим C_K^2 классификаторов).

Каждый такой классификатор будем обучать только на объектах классов i и j.

- **В качестве итогового предсказания выдадим класс, который предсказало наибольшее число алгоритмов:**

$$a(x) = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}} \sum_{i=1}^{K} \sum_{j \neq i} [a_{ij}(x) = k]$$

ПОДХОД ALL-VS-ALL

- Для каждой пары классов i и j обучим бинарный классификатор $a_{ij}(x)$, который будет предсказывать класс i или j

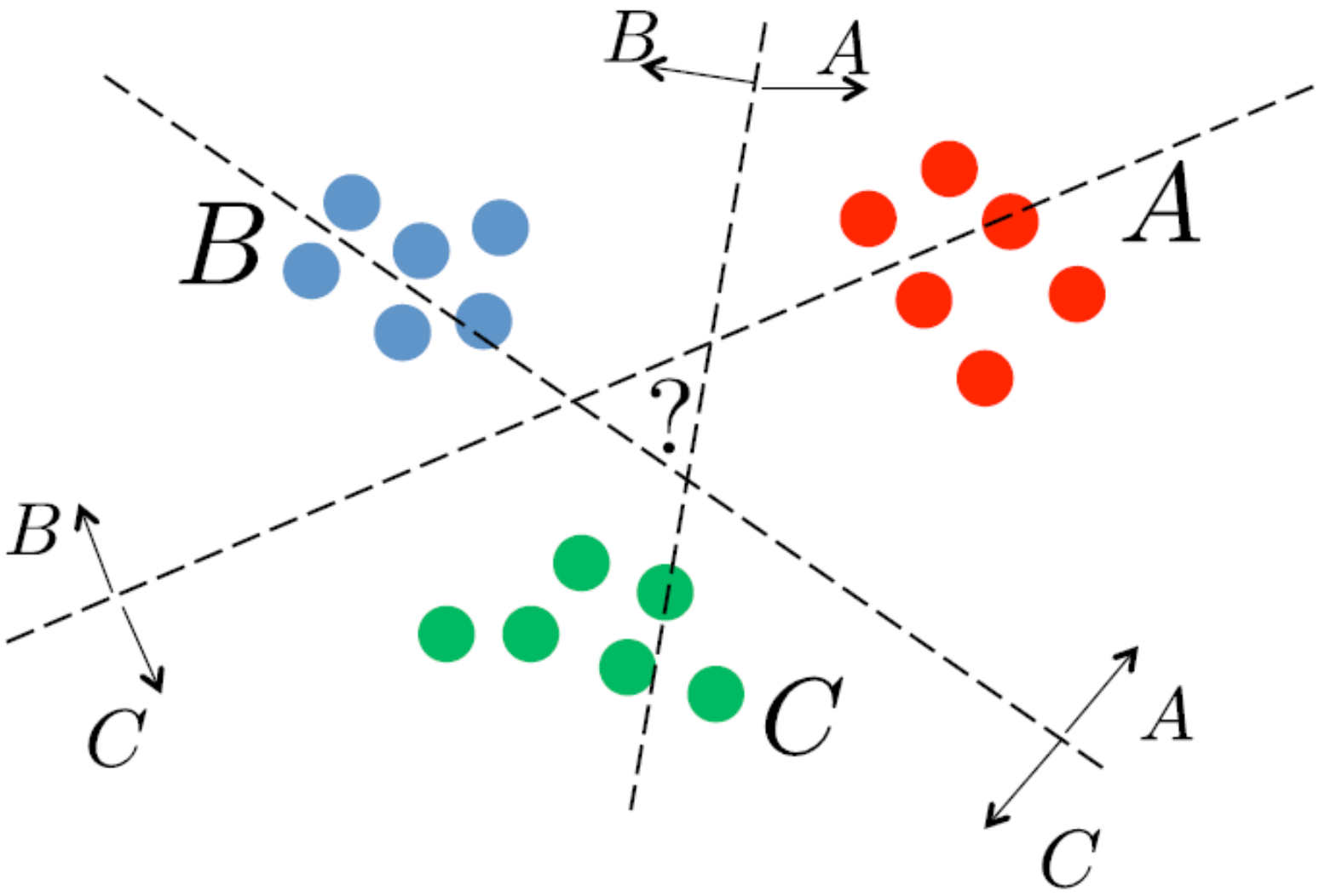
(если всего K классов, то получим **C_K^2 классификаторов**).

Каждый такой классификатор будем обучать только на объектах классов i и j.

- В качестве итогового предсказания выдадим класс, который предсказало наибольшее число алгоритмов:

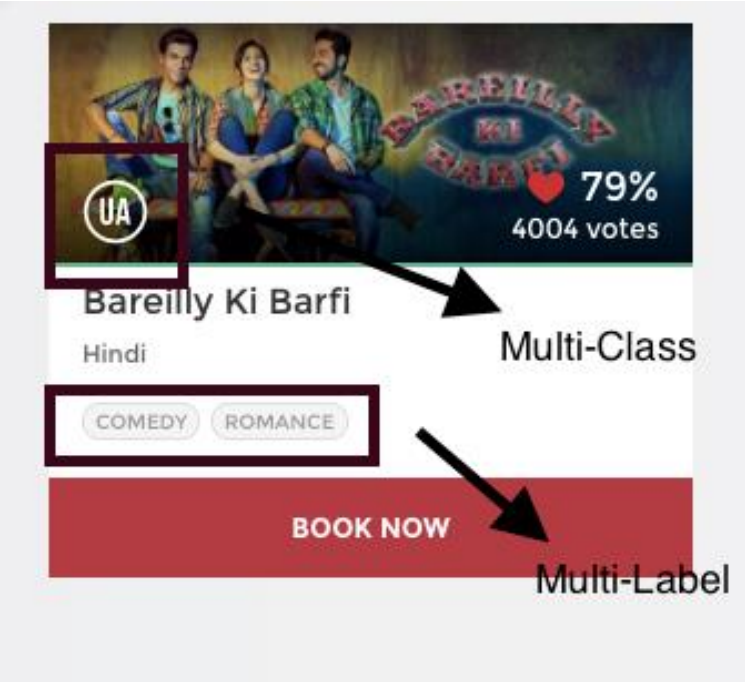
$$a(x) = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}} \sum_{i=1}^{K} \sum_{j \neq i} [a_{ij}(x) = k]$$

ПОДХОД ALL-VS-ALL



MULTICLASS AND MULTI-LABEL CLASSIFICATION

- Если каждый объект может принадлежать только одному классу, то решаем задачу multiclass классификации
- Если каждый объект может принадлежать нескольким классам (задача классификации с пересекающимися классами), то решаем задачу multi-label классификации.



МЕТРИКИ КАЧЕСТВА

Идея: сводим подсчет метрик к бинарному случаю

Подход 1 (микроусреднение, micro average):

- Вычислим для каждого двухклассового классификатора $a^k(x) = [a(x) = k]$ метрики TP_k, FP_k, FN_k, TN_k
- Усредним каждую характеристику по всем классам, например, $TP = \frac{1}{K}\sum_{k=1}^{K} TP_k$.

Тогда точность в многоклассовом случае:

$$precision(a, X) = \frac{TP}{TP + FP}$$

МЕТРИКИ КАЧЕСТВА

Идея: сводим подсчет метрик к бинарному случаю

Подход 2 (макроусреднение, macro average):

- Вычислим для каждого двухклассового классификатора $a^k(x) = [a(x) = k]$ метрики TP_k, FP_k, FN_k, TN_k
- Вычислим итоговую метрику для каждого класса в отдельности: $precision_k(a, X) = \frac{TP_k}{TP_k + FP_k}$

Тогда точность в многоклассовом случае:

$$precision(a, X) = \frac{1}{K} \sum_{k=1}^{K} precision_k(a, X)$$

МЕТРИКИ КАЧЕСТВА (ПРИМЕР)

Результаты некоторого классификатора:

		True/Actual		
		Cat ()	Fish ()	Hen ()
Predicted	Cat ()	4	6	3
	Fish ()	1	2	0
	Hen ()	1	2	6

МЕТРИКИ КАЧЕСТВА (ПРИМЕР)

		True/Actual		
		Cat ()	Fish ()	Hen ()
Predicted	Cat ()	4	6	3
	Fish ()	1	2	0
	Hen ()	1	2	6

	precision	recall	f1-score	support
Cat	0.308	0.667	0.421	6
Fish	0.667	0.200	0.308	10
Hen	0.667	0.667	0.667	9
micro avg	0.480	0.480	0.480	25
macro avg	0.547	0.511	0.465	25
weighted avg	0.581	0.480	0.464	25

2. ОТБОР ПРИЗНАКОВ

VARIANCE THRESHOLD

- Можем удалить признаки, которые имеют очень маленькую дисперсию, т.е. практически константы.

ОТБОР ПРИЗНАКОВ ПО КОРРЕЛЯЦИИ С ЦЕЛЕВОЙ ПЕРЕМЕННОЙ

- Для каждого признака вычислим его корреляцию с целевой переменной. Будем выкидывать признаки, имеющие маленькую корреляцию.

БОЛЕЕ СЛОЖНЫЕ МЕТОДЫ

- Filtration methods (фильтрационные методы)
- Wrapping methods (оберточные методы)
- Model selection (встроенный в модель отбор признаков)

1. ФИЛЬТРАЦИОННЫЕ МЕТОДЫ

- **Фильтрационные методы - это отбор признаков по различным статистическим тестам.** Идея метода состоит в вычислении влияния каждого признака в отдельности на целевую переменную (с помощью вычисления некоторой статистики).

Очевидный плюс метода: скорость, так как мы вычисляем значения N статистик, где N - количество признаков.

1. ФИЛЬТРАЦИОННЫЕ МЕТОДЫ

В sklearn есть сразу несколько методов, использующих отбор по статистическим критериям. Среди них выделим следующие:

- **SelectKBest** - оставляет k признаков с наибольшим значением выбранной статистики
- **SelectPercentile** - оставляет признаки со значениями выбранной статистики, попавшими в заданную пользователем квантиль

1. СТАТИСТИЧЕСКИЕ ТЕСТЫ ДЛЯ ОТБОРА ПРИЗНАКОВ (ПРИМЕР)

- **Тест χ^2** используется в статистике для проверки независимости двух событий.
- Поскольку χ^2 проверяет степень независимости между двумя переменными, а мы хотим сохранить только признаки, наиболее зависимые от метки, то будем вычислять χ^2 между каждым признаком и меткой, сохраняя только признаки с наибольшими значениями.
- Критерий χ^2 можем применять только для бинарных или порядковых признаков.

1. СТАТИСТИЧЕСКИЕ ТЕСТЫ ДЛЯ ОТБОРА ПРИЗНАКОВ (ПРИМЕР)

- Статистика χ^2 вычисляется по формуле

$$\chi^2(X;Y) = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}},$$

где O_{ij} - наблюдаемая частота, E_{ij} - ожидаемая частота.

Пример: хотим выявить влияние курения на гипертонию:

	Артериальная гипертония есть (1)	Артериальной гипертонии нет (0)	**Всего**
Курящие (1)	40	30	**70**
Некурящие (0)	32	48	**80**
Всего	**72**	**78**	**150**

Вычисляем χ^2: χ^2 = (40-33.6)^2/33.6 + (30-36.4)^2/36.4 + (32-38.4)^2/38.4 + (48-41.6)^2/41.6 = 4.396.

Подробно про вычисление χ^2 почитать здесь

1. СТАТИСТИЧЕСКИЕ ТЕСТЫ ДЛЯ ОТБОРА ПРИЗНАКОВ

- Статистика χ^2 вычисляется по формуле

$$\chi^2(X;Y) = \sum_{i,j} \frac{\left(O_{ij} - E_{ij}\right)^2}{E_{ij}},$$

где O_{ij} - наблюдаемая частота, E_{ij} - ожидаемая частота.

Пример: хотим выявить влияние курения на гипертонию:

	Артериальная гипертония есть (1)	Артериальной гипертонии нет (0)	**Всего**
Курящие (1)	40	30	**70**
Некурящие (0)	32	48	**80**
Всего	**72**	**78**	**150**

Вычисляем χ^2: χ^2 = (40-33.6)^2/33.6 + (30-36.4)^2/36.4 + (32-38.4)^2/38.4 + (48-41.6)^2/41.6 = 4.396.

При отборе признаков оставляем k (или заданную квантиль) признаков с наибольшим значением χ^2.

1. СТАТИСТИЧЕСКИЕ ТЕСТЫ ДЛЯ ОТБОРА ПРИЗНАКОВ

- mutual information:

для векторов X и Y статистика вычисляется по формуле

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

- хи-квадрат:

$$\chi^2(X;Y) = \sum_{i,j} \frac{\left(O_{ij} - E_{ij}\right)^2}{E_{ij}},$$

где O_{ij} - наблюдаемая частота, E_{ij} - ожидаемая частота.

2. ОБЕРТОЧНЫЕ МЕТОДЫ

Оберточные методы используют ***жадный отбор признаков***, т.е. последовательно выкидывают наименее подходящие по мнению методов признаки.

В sklearn есть оберточный метод - Recursive Feature Elimination (RFE).

Параметры метода:

a) алгоритм, используемый для отбора признаков (например, RandomForest)

b) число признаков, которое мы хотим оставить.

2. ЖАДНЫЙ ОТБОР ПРИЗНАКОВ

1шаг: Перебираем все признаки и убираем тот, удаление которого сильнее всего уменьшает ошибку

2 шаг: Из оставшихся признаков убираем тот, удаление которого сильнее всего уменьшает ошибку

И т.д.

3. ВСТРОЕННЫЕ В МОДЕЛЬ МЕТОДЫ

Напоминание: L_1-регуляризация умеет отбирать признаки.

$$Q(w) + \alpha \sum_{i=1}^{d} |w_j| \to \min_w$$

3. ВСТРОЕННЫЕ В МОДЕЛЬ МЕТОДЫ

Напоминание: L_1-регуляризация умеет отбирать признаки.

$$Q(w) + \alpha \sum_{i=1}^{d} |w_j| \to \min_w$$

Рассмотрим другой вариант регуляризации, которая тоже умеет отбирать признаки (L_0-регуляризация):

$$Q(w) + \alpha \sum_{i=1}^{d} [w_j \neq 0] \to \min_w$$

3. ИНФОРМАЦИОННЫЕ КРИТЕРИИ

- **Информационный критерий -** мера качества модели**,** учитывающая степень «подгонки» модели под данные с корректировкой (штрафом) на используемое количество параметров**.**
- Информационные критерии основаны на **компромиссе между точностью и сложностью модели.** Критерии различаются тем**,** как они обеспечивают этот баланс**.**

3. КРИТЕРИЙ AIC

Критерий Акаике (AIC, Akaike Information Criterion)

- Дополнительно предполагаем, что модель a – линейная.

$$\boldsymbol{AIC(a,X) = Q(a,X) + \frac{2\hat{\sigma}^2}{l} n \to min}$$

Q – функционал ошибки

$\hat{\sigma}^2$ - оценка дисперсии ошибки $D(y_i - a(x_i))$

n – количество используемых признаков

l – число объектов

3. КРИТЕРИЙ AIC

Критерий Акаике (AIC, Akaike Information Criterion)

- Дополнительно предполагаем, что модель a – линейная.

$$\boldsymbol{AIC(a, X) = Q(a, X) + \frac{2\hat{\sigma}^2}{l} n \to min}$$

Q – функционал ошибки

$\hat{\sigma}^2$ - оценка дисперсии ошибки $D(y_i - a(x_i))$

n – количество используемых признаков

l – число объектов

- Если Q – среднеквадратичная ошибка для линейной регрессии, и шумы нормально распределены, то

$$\boldsymbol{AIC = -\ln \Pi + n}$$

3. КРИТЕРИЙ BIC

Критерий Шварца (BIC, Bayesian Information Criterion)

$$BIC(a, X) = \frac{l}{\hat{\sigma}^2}\left(Q(a, X) + \frac{\hat{\sigma}^2 lnl}{l} n\right) \to min$$

- Если Q – среднеквадратичная ошибка для линейной регрессии, и шумы нормально распределены, то

$$BIC = -\ln П + \frac{n}{2} lnl$$

3. ОТБОР ПРИЗНАКОВ С ПОМОЩЬЮ ИНФОРМАЦИОННЫХ КРИТЕРИЕВ

- Если в модели k признаков (регрессоров), то существует 2^k всевозможных моделей
- В идеале необходимо построить все 2^k моделей, для каждой посчитать значение критерия качества (AIC, BIC) и выбрать модель, лучшую по этому критерию
- При большом количестве регрессоров используют метод включений-исключений (жадный отбор признаков)

3. ПРИМЕР

Задача предсказания уровня преступности в разных штатах по следующим признакам:

Регрессор
Нулевой коэффициент
Возраст
Южный штат(да/нет)
Образование
Расходы
Труд
Количество мужчин
Численность населения
Безработные (14-24)
Безработные (25-39)
Доход

3. ПРИМЕР: ОТБОР ПРИЗНАКОВ ПО AIC

- Мы решаем задачу линейной регрессии с предположением, что ошибки нормально распределены, поэтому $AIC = \ln \Pi(a, X) - n \to max$.

В модели с полным набором регрессоров AIC = -310.37. В порядке убывания AIC при удалении каждой из переменных равен:

Численность населения (AIC = -308), Труд (AIC = -309), Южный штат (AIC = -309), Доход (AIC = -309), Количество мужчин (AIC = -310), Безработные I (AIC = -310), Образование (AIC = -312), Безработные II (AIC = -314), Возраст (AIC = -315), Расходы (AIC = -324).

Таким образом, имеет смысл удалить переменную “Население”.

3. ПРИМЕР: ОТБОР ПРИЗНАКОВ ПО AIC

Южный штат (AIC = -308), Труд (AIC = -308), Доход (AIC = -308), Количество мужчин (AIC = -309), Безработные I (AIC = -309), Образование (AIC = -310), Безработные II (AIC = -313), Возраст (AIC = -313), Расходы (AIC = -329).
Удаляем переменные до тех пор, пока не удастся больше получить увеличения AIC.

Уровень преступности = 1.2 Возраст + 0.75 Образование + 0.87 Расходы + 0.34 Количество мужчин − 0.86 Безработные I + 2.31 Безработные II.

МЕТОД ГЛАВНЫХ КОМПОНЕНТ (PRINCIPAL COMPONENT ANALYSIS, PCA)

МЕТОД ГЛАВНЫХ КОМПОНЕНТ

Цель: *хотим придумать новые признаки, каким-то образом выражающиеся через старые, причем новых признаков хочется получить меньше, чем старых*.

Сегодня будем рассматривать только случай, когда новые признаки **линейно** выражаются через старые.

МЕТОД ГЛАВНЫХ КОМПОНЕНТ

Постановка задачи:

- $x_1, \dots, x_n$ - исходные числовые признаки
- $z_1, \dots, z_d$ – новые числовые признаки, $d \leq n$

Хотим:

1. чтобы новые числовые признаки z_j линейно выражались через исходные признаки x_i

2. чтобы при переходе к новым признакам было потеряно наименьшее количество исходной информации

МЕТОД ГЛАВНЫХ КОМПОНЕНТ

1. чтобы новые числовые признаки z_j линейно выражались через исходные признаки x_i

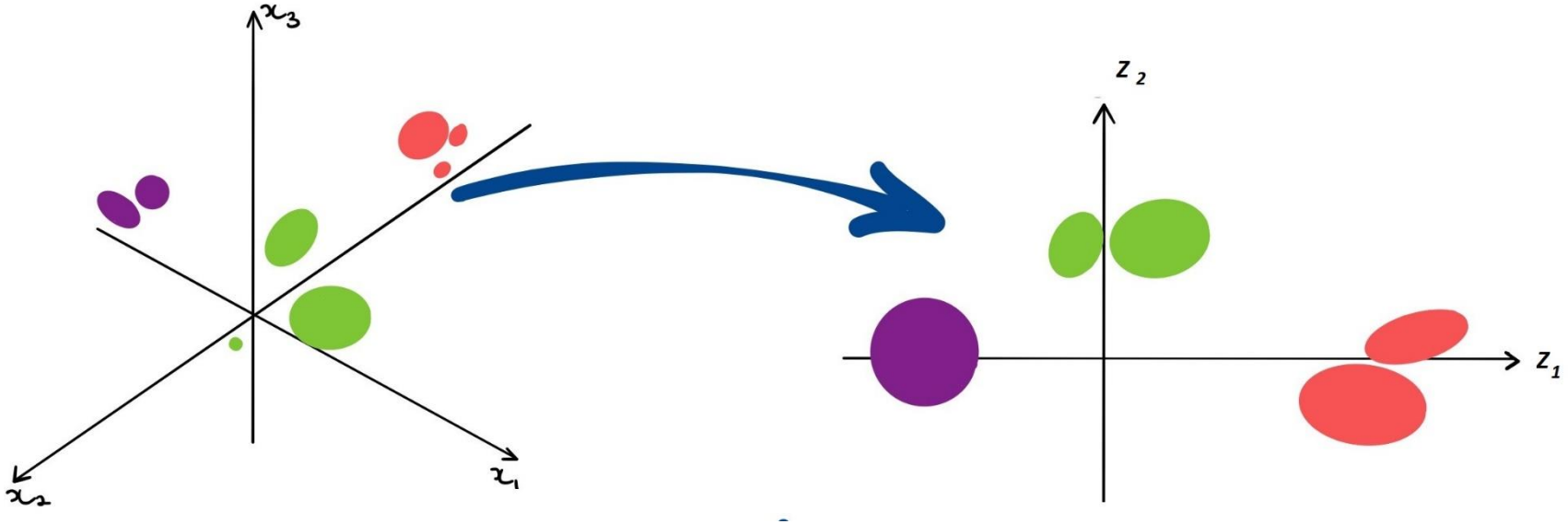
$$\begin{cases} z_1 = u_{11}x_1 + \cdots + u_{1n}x_n \\ z_2 = u_{21}x_1 + \cdots + u_{2n}x_n \\ \qquad\qquad \cdots \\ z_d = u_{d1}x_1 + \cdots + u_{dn}x_n \end{cases}$$

Геометрическая интерпретация: новые признаки z_i – это проекции исходных признаков x_i на некоторые векторы (компоненты) u.

МЕТОД ГЛАВНЫХ КОМПОНЕНТ

1. чтобы новые числовые признаки z_j линейно выражались через исходные признаки x_i

Геометрически это означает, что мы проецируем пространство признаков размерности n на некоторое линейное подпространство размерности d:



ПОЯСНЕНИЕ: ПРОЕКЦИЯ

- Проекция вектора x на вектор (компоненту) u_i:

$$(x, u_i)$$

- Проекция выборки X на компоненту u_i:

$$Xu_i$$

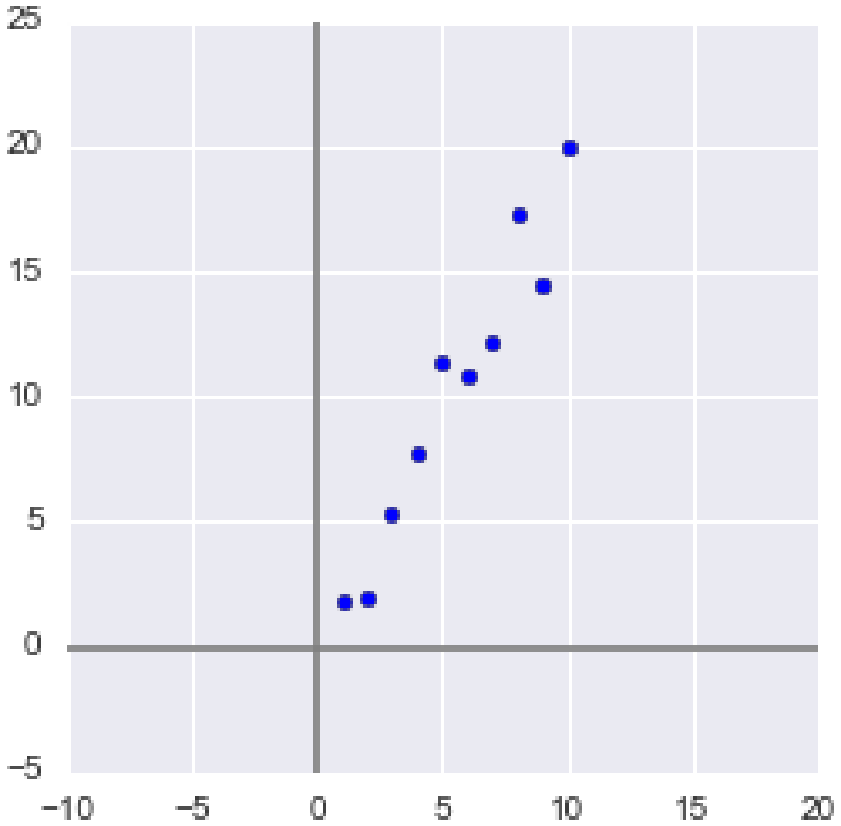
МЕТОД ГЛАВНЫХ КОМПОНЕНТ

2. чтобы при переходе к новым признакам было потеряно наименьшее количество исходной информации.

Дисперсия выборки, посчитанная в новых признаках, показывает, как много информации нам удалось сохранить после понижения размерности, поэтому дисперсия в новых признаках должна быть максимальной.

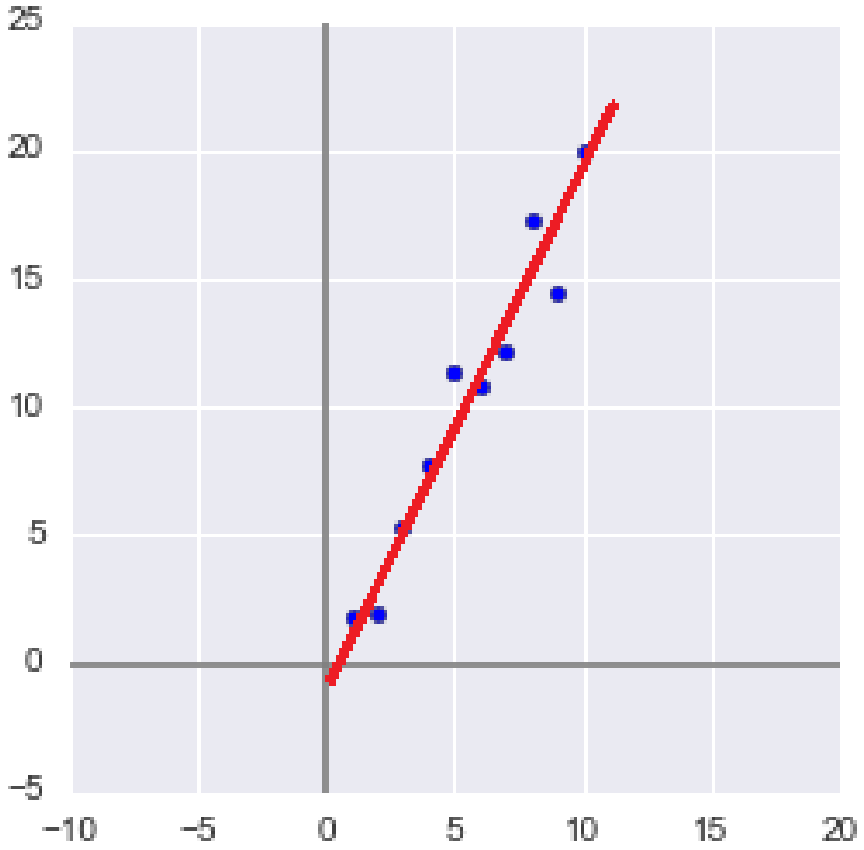
ПРИМЕР

Хотим спроецировать двумерные данные X на одномерный вектор u так, чтобы дисперсия проекции Xu была максимальной:



ПРИМЕР

Хотим спроецировать двумерные данные X на одномерный вектор u так, чтобы дисперсия проекции Xu была максимальной:



ПОСТАНОВКА ЗАДАЧИ

Будем искать такие компоненты $u_1, u_2, \dots, u_d$, что:

1) Они ортогональны, т.е. $(u_i, u_j) = 0$
2) Они нормированы, т.е. $||u_i|| = 1$
3) дисперсия проекции выборки на них максимальна:

$$D(Xu_i) \to \max_{u_i}, \qquad i = 1, \dots, d$$

ВАЖНОЕ ДЕЙСТВИЕ

Центрируем исходные данные, то есть вычтем из каждого признака его среднее значение.

ДИСПЕРСИЯ ПРОЕКЦИИ

- Мы уже выяснили, что проекция выборки X на компоненту u_i:

$$Xu_i$$

- Тогда проекция выборки на первые d компонент, задаваемых столбцами матрицы U_d:

$$XU_d$$

ДИСПЕРСИЯ ПРОЕКЦИИ

- Мы уже выяснили, что проекция выборки X на компоненту u_i:

$$Xu_i$$

- Тогда проекция выборки на первые d компонент, задаваемых столбцами матрицы U_d:

$$XU_d$$

- Тогда дисперсия проекции – это след ковариационной матрицы:

$$tr\big((XU_d)^T(XU_d)\big) = \sum_{i=1}^{d} \big||Xu_i|\big|^2 \to \max_u$$

ПЕРВЫЙ ШАГ

- Будем искать первую компоненту, u_1:

$$\begin{cases} \left|\left|Xu_1\right|\right|^2 \to \max\limits_{u_1} \\ \left|\left|u_1\right|\right|^2 = 1 \end{cases}$$

ПЕРВЫЙ ШАГ

- Будем искать первую компоненту, u_1:

$$\begin{cases} \left|\left|Xu_1\right|\right|^2 \to \max\limits_{u_1} \\ \left|\left|u_1\right|\right|^2 = 1 \end{cases}$$

Решение:

- Запишем лагранжиан

$$L(u_1, \lambda) = \left|\left|Xu_1\right|\right|^2 + \lambda(\left|\left|u_1\right|\right|^2 - 1)$$

ПЕРВЫЙ ШАГ

- Будем искать первую компоненту, u_1:

$$\begin{cases} \left|\left|Xu_1\right|\right|^2 \to \max\limits_{u_1} \\ \left|\left|u_1\right|\right|^2 = 1 \end{cases}$$

Решение:

- Запишем лагранжиан

$$L(u_1, \lambda) = \left|\left|Xu_1\right|\right|^2 + \lambda(\left|\left|u_1\right|\right|^2 - 1)$$

- $\frac{\partial L}{\partial u_1} = ?$

ПЕРВЫЙ ШАГ

- Будем искать первую компоненту, u_1:

$$\begin{cases} \left|\left|Xu_1\right|\right|^2 \to \max\limits_{u_1} \\ \left|\left|u_1\right|\right|^2 = 1 \end{cases}$$

Решение:

- Запишем лагранжиан

$$L(u_1, \lambda) = \left|\left|Xu_1\right|\right|^2 + \lambda(\left|\left|u_1\right|\right|^2 - 1)$$

- $\frac{\partial L}{\partial u_1} = 2X^T Xu_1 + 2\lambda u_1 = 0 \Rightarrow X^T Xu_1 = -\lambda u_1$ - собств.в-р.

ПЕРВЫЙ ШАГ

- Будем искать первую компоненту, u_1:

$$\begin{cases} \left|\left|Xu_1\right|\right|^2 \to \max\limits_{u_1} \\ \left|\left|u_1\right|\right|^2 = 1 \end{cases}$$

Решение:

- Запишем лагранжиан

$$L(u_1, \lambda) = \left|\left|Xu_1\right|\right|^2 + \lambda(\left|\left|u_1\right|\right|^2 - 1)$$

- $\frac{\partial L}{\partial u_1} = 2X^T Xu_1 + 2\lambda u_1 = 0 \Rightarrow X^T Xu_1 = -\lambda u_1$ - собств.в-р.
- $\left|\left|Xu_1\right|\right|^2 = u_1^T X^T Xu_1 = \lambda u_1^T u_1 = \lambda \to \max_{u_1}$ - max собств. значение.

ПЕРВЫЙ ШАГ

- Будем искать первую компоненту, u_1:

$$\begin{cases} \left|\left|Xu_1\right|\right|^2 \to \max\limits_{u_1} \\ \left|\left|u_1\right|\right|^2 = 1 \end{cases}$$

Ответ:

u_1 - собственный вектор матрицы ковариаций $X^T X$ с максимальным собственным значением.

ПРОЕКЦИИ МЕТОДА ГЛАВНЫХ КОМПОНЕНТ

- Пусть X – матрица объект-признак для исходных признаков.
- Метод главных компонент делает проекцию исходных объектов на гиперплоскость некоторой размерности d.

Теорема. Базисные векторы этой гиперплоскости – это собственные векторы матрицы $\mathrm{X}^{\mathrm{T}}\mathrm{X}$ (матрица ковариаций), соответствующие d её наибольшим собственным значениям.

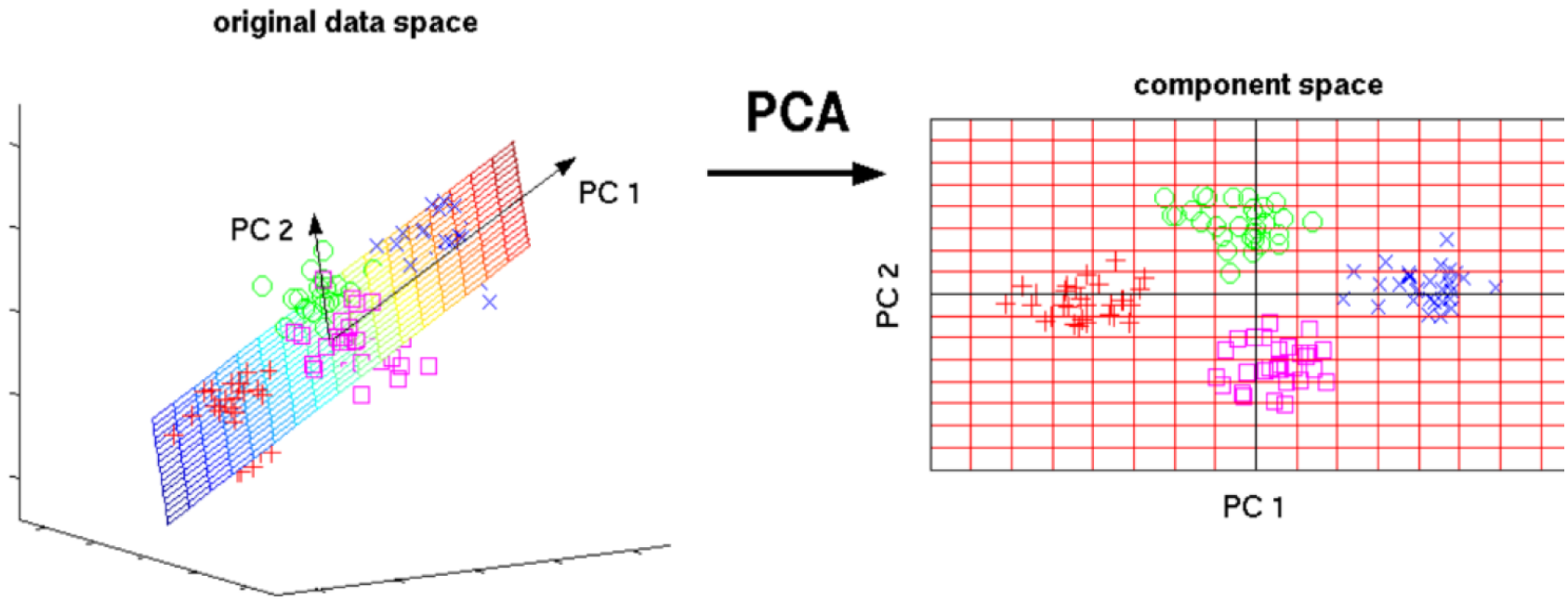
КОНСТРУКТИВНОЕ ПОСТРОЕНИЕ БАЗИСА В PCA

- Находим вектор $u_1 = argmax_u\big(D(Xu)\big)$ и нормируем его: $u_1 \to \frac{u_1}{||u_1||}$
- Находим вектор $u_2 = argmax_u\big(D(Xu)\big)$ такой, что $(u_1, u_2) = 0$ и нормируем его: $u_2 \to \frac{u_2}{||u_2||}$
- Находим вектор $u_3 = argmax_u\big(D(Xu)\big)$ такой, что $(u_1, u_3) = (u_2, u_3) = 0$ и нормируем его: $u_3 \to \frac{u_3}{||u_3||}$.

И т.д.

Получаем ортонормированный базис $\{u_1, u_2, \dots, u_d\}$.

ПРОЕКЦИЯ НА ГИПЕРПЛОСКОСТЬ



ПРИМЕНЕНИЕ МЕТОДА

- Когда главные компоненты найдены, можно проецировать на них и новые данные:

$$Z' = X'U_d.$$

ДОЛЯ ОБЪЯСНЕННОЙ ДИСПЕРСИИ

- Упорядочим собственные значения матрицы $X^T X$ по убыванию: $\lambda_1 \geq \lambda_2 \geq \cdots > \lambda_n \geq 0$.

- Доля дисперсии, объяснённой j-й компонентой (explained variance ratio):

$$\boldsymbol{\delta_j = \frac{\lambda_j}{\sum_{i=1}^{n} \lambda_n}}$$

- Доля дисперсии, объясняемой первыми k компонентами:

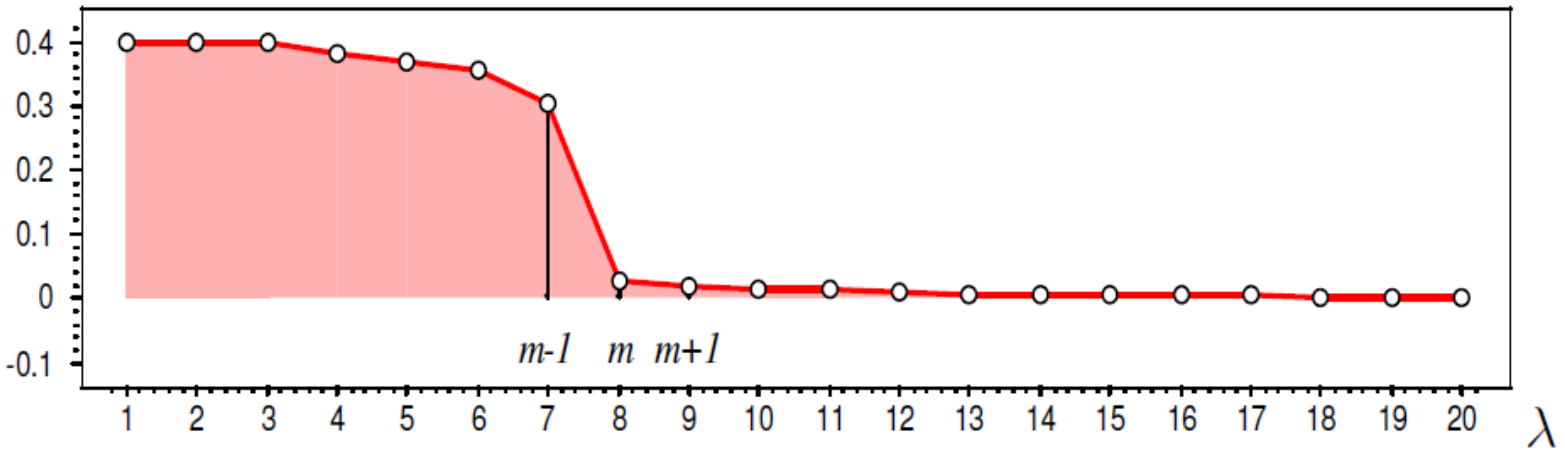
$$\boldsymbol{\delta = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_n} = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_n}}$$

ВЫБОР ЧИСЛА ГЛАВНЫХ КОМПОНЕНТ

- Эффективная размерность выборки – это наименьшее целое m, при котором *доля необъясненной дисперсии*

$$E_m = \frac{||ZU^T - X||^2}{||X||^2} = \frac{\lambda_{m+1} + \cdots + \lambda_n}{\sum_{i=1}^{n} \lambda_i} \leq \varepsilon$$

Критерий крутого склона:



ПРИМЕР: FACES DATASET

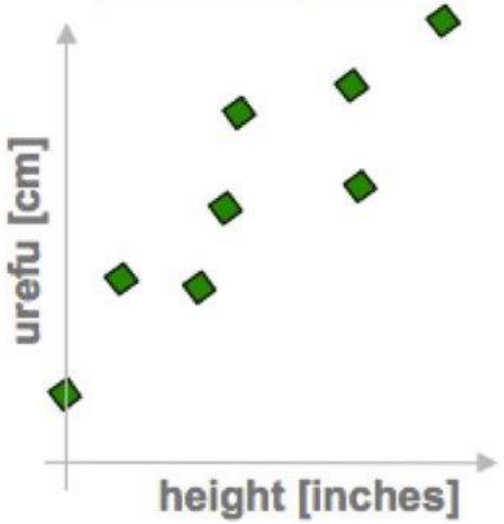
FACES DATASET (ГЛАВНЫЕ КОМПОНЕНТЫ)

ВОССТАНОВЛЕННОЕ ИЗОБРАЖЕНИЕ

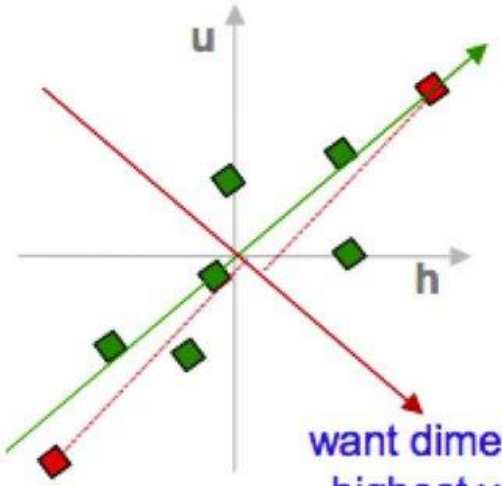
PCA in a nutshell

1. correlated hi-d data

("urefu" means "height" in Swahili)



2. center the points



want dimension of highest variance

3. compute covariance matrix

	h	u
h	2.0	0.8
u	0.8	0.6

$$\text{cov}(h,u) = \frac{1}{n}\sum_{i=1}^{n} h_i u_i$$

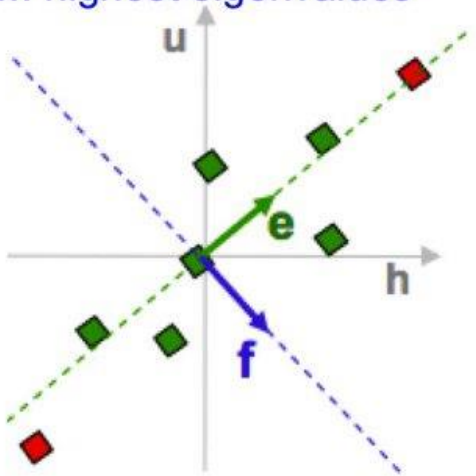
4. eigenvectors + eigenvalues

$$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} e_h \\ e_u \end{pmatrix} = \lambda_e \begin{pmatrix} e_h \\ e_u \end{pmatrix}$$

$$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} f_h \\ f_u \end{pmatrix} = \lambda_f \begin{pmatrix} f_h \\ f_u \end{pmatrix}$$

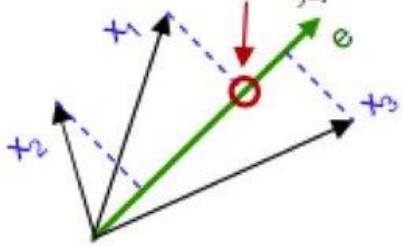
`eig(cov(data))`

5. pick m<d eigenvectors w. highest eigenvalues

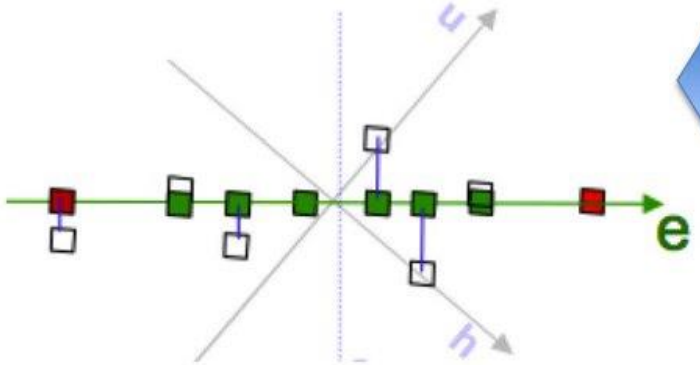


6. project data points to those eigenvectors

$$x_e' = x^T e = \sum_{j=1}^{d} x_{ij} e_j$$



7. uncorrelated low-d data



СИНГУЛЯРНОЕ РАЗЛОЖЕНИЕ МАТРИЦЫ (SINGULAR VALUE DECOMPOSITION, SVD)

Теорема. Матрицу $A \in \mathbb{R}^{m \times n}$ можно представить в виде

$$\boldsymbol{A = U\Sigma V^T},$$

- где $U \in \mathbb{R}^{m \times m}, V \in \mathbb{R}^{n \times n}$ - ортогональные матрицы,
- $\Sigma \in \mathbb{R}^{m \times n}$ - диагональная матрица с ненулевыми элементами $\sigma_i = \sqrt{\lambda_i}$, где λ_i - собственные значения матрицы $A^T A$.

СИНГУЛЯРНОЕ РАЗЛОЖЕНИЕ МАТРИЦЫ (SVD)

Теорема. Матрицу $A \in \mathbb{R}^{m \times n}$ можно представить в виде

$$\boldsymbol{A = U\Sigma V^T},$$

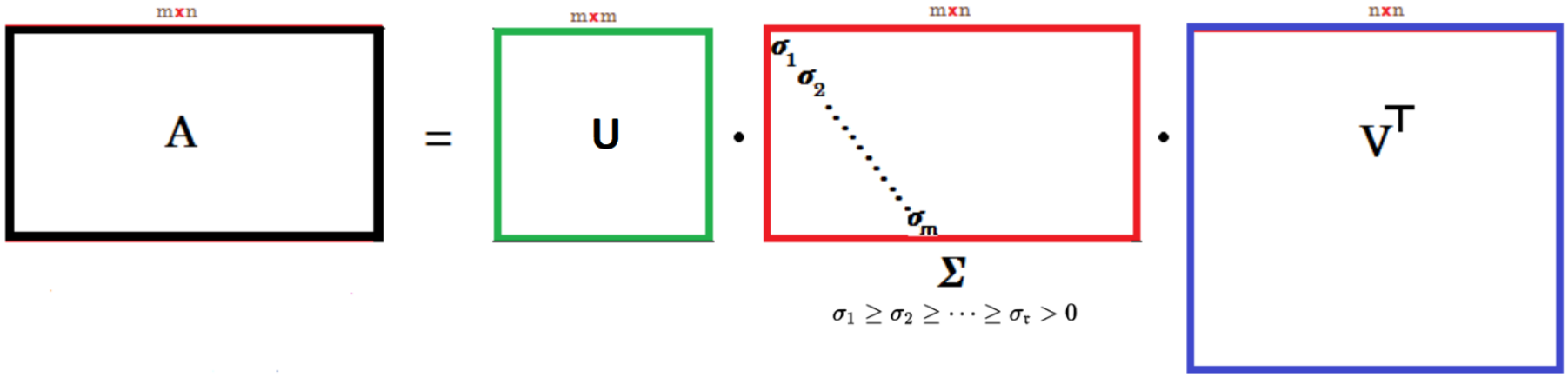
- где $U \in \mathbb{R}^{m \times m}, V \in \mathbb{R}^{n \times n}$ - ортогональные матрицы,
- $\Sigma \in \mathbb{R}^{m \times n}$ - диагональная матрица с ненулевыми элементами $\sigma_i = \sqrt{\lambda_i}$, где λ_i - собственные значения матрицы $A^T A$.

При этом

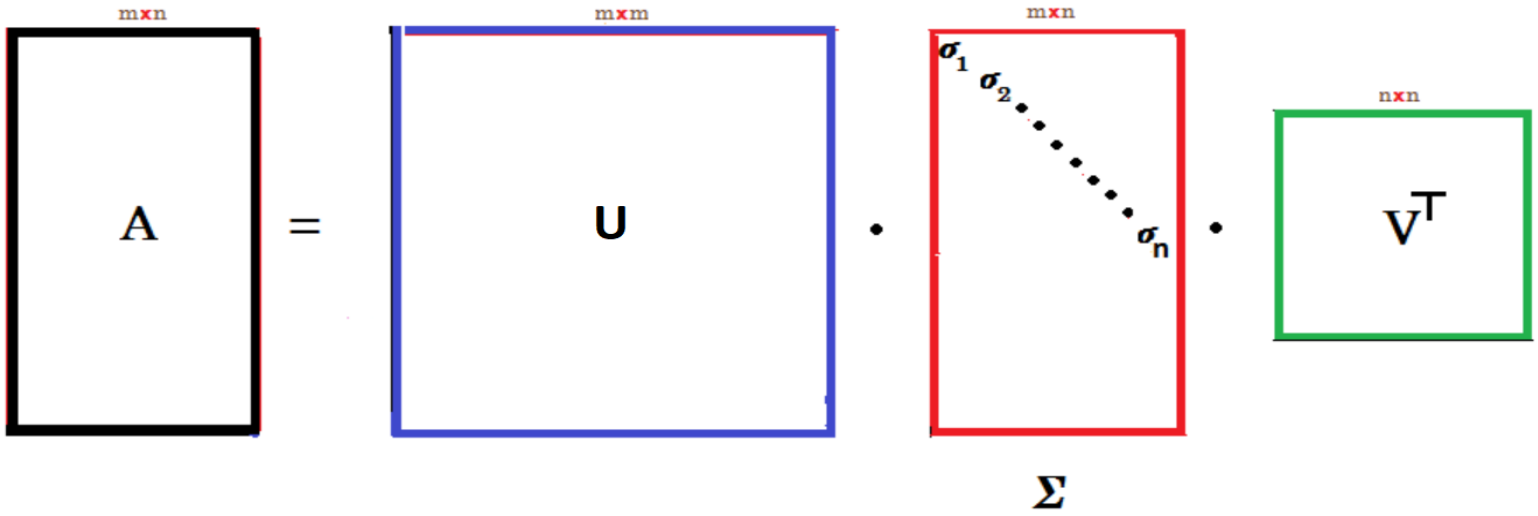
- Столбцы матрицы U являются собственными векторами матрицы AA^T
- Столбцы матрицы V являются собственными векторами матрицы $A^T A$.

SINGULAR VALUE DECOMPOSITION

- При $m \leq n$:



- При $m > n$:



СВЯЗЬ SVD И PCA

Пусть X – матрица объект-признак, для которой мы хотим снизить размерность и $X = U\Sigma V^T$ её SVD-разложение.

Тогда:

- Столбцы матрицы $\boldsymbol{V}$ – это собственные векторы матрицы $X^T X$, т.е. векторы $v_1, \dots, v_n$ – главные компоненты.
- Столбцы матрицы $\boldsymbol{U\Sigma}$ – это новые признаки, то есть, проекции исходных признаков на главные компоненты $Z = Xv$

($X = U\Sigma \mathrm{V}^{\mathrm{T}} \Leftrightarrow U\Sigma = XV$).

- Сингулярные числа матрицы Σ – это корни из собственных чисел матрицы $X^T X$.

СВЯЗЬ SVD И PCA

- Столбцы матрицы V – это собственные векторы матрицы $X^T X$, т.е. векторы $v_1, \dots, v_n$ – главные компоненты.
- Столбцы матрицы $U\Sigma$ – это новые признаки $z = Xv$

($X = U\Sigma \mathrm{V}^{\mathrm{T}} \Leftrightarrow U\Sigma = XV$).

- Сингулярные числа матрицы Σ – это корни из собственных чисел матрицы $X^T X$.

Для снижения размерности берем первые k столбцов матрицы U и верхний $k \times k$-квадрат матрицы Σ, тогда матрица $U_k \Sigma_k$ содержит k новых признаков, соответствующих первым k главным компонентам.

ЧТО ЛУЧШЕ: PCA ИЛИ SVD?

- Существуют вычислительные трудности с нахождением собственных значений, в этом недостаток PCA.
- Существует итерационный алгоритм для нахождения SVD (без нахождения собственных значений) http://www.machinelearning.ru/wiki/index.php?title=Простой_итерационный_алгоритм_сингулярного_разложения.

Поэтому вычислительно эффективнее использовать SVD при прочих равных.