НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Internet Studies Lab, Department of Applied Mathematics and Business Informatics

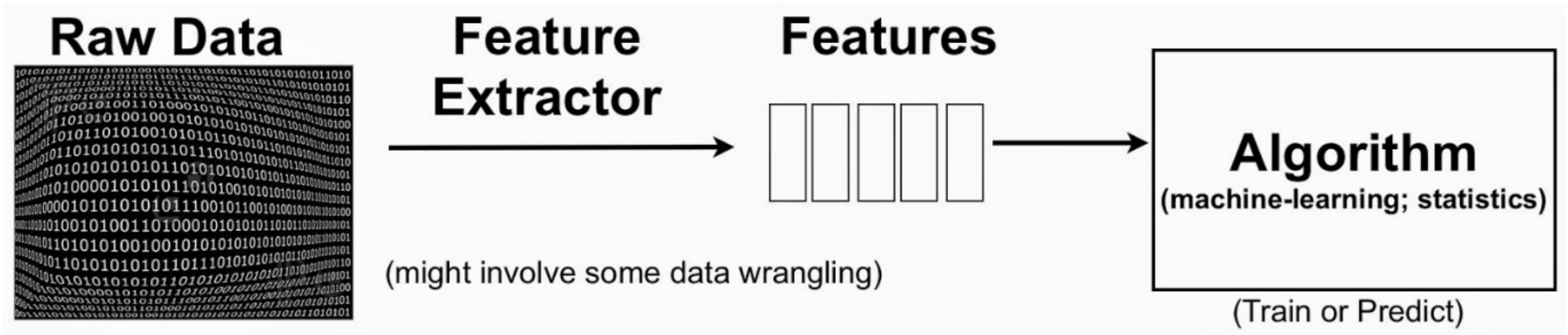
INTRODUCTION TO FEATURE SELECTION

Анализ баз данных в публичном управлении

Кольцов С.Н.

Saint Petersburg, 21.09.2018

FEATURE ENGINEERING



1. Principal Components Analysis.
2. Information gain
3. Хи-квадрат (chi-square)
4. One-way analysis of variance
5. Relief

НЕКОТОРЫЕ ПОНЯТИЯ ИЗ СТАТИСТИКИ

Средняя величина $\bar{x} = \sqrt[k]{\frac{\sum x_i^k}{n}}$

x_i - i-й вариант усредняемого признака, n - число вариантов, к - порядок средней.

Средняя арифметическая: (математическое ожидание) $\bar{x} = \frac{\sum x_i}{n}$

Среднее квадратическое отклонение признака: $\sigma = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n}}$

Дисперсия случайной величины : $D = \frac{\sum (x - \bar{x})^2}{n}$

СОБСТВЕННЫЕ ЧИСЛА И СОБСТВЕННЫЕ ВЕКТОРА

Пусть число λ и вектор x таковы что $Ax = \lambda x.$ $(A - \lambda E)X = 0, \quad X \neq 0.$

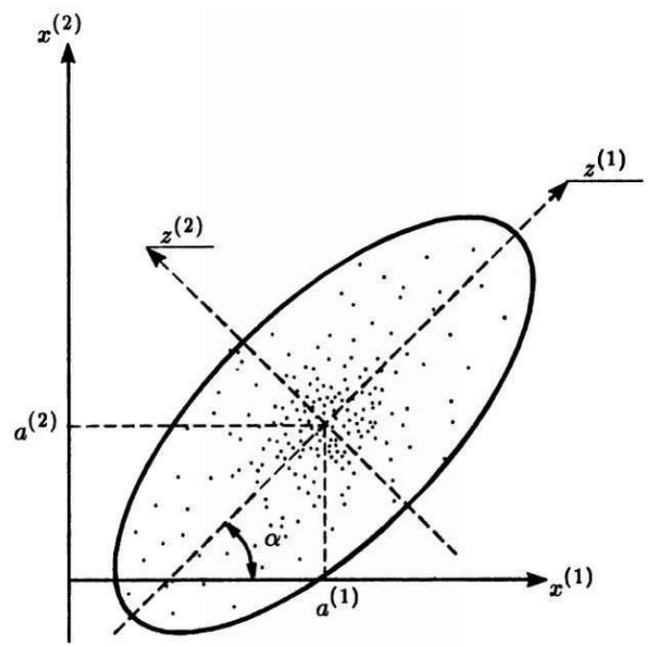
Тогда число λ называется собственным числом линейного оператора A, а вектор x собственным вектором этого оператора, соответствующим собственному числу λ.

PRINCIPAL COMPONENTS ANALYSIS ИЛИ МЕТОД ГЛАВНЫХ КОМПОНЕНТ

Метод главных компонент - это один из способов понижения размерности, состоящий в переходе к новому ортогональному базису, оси которого ориентированы по направлениям максимальной дисперсии набора входных данных. Вдоль первой оси нового базиса дисперсия максимальна, вторая ось максимизирует дисперсию при условии ортогональности первой оси, и т.д., последняя ось имеет минимальную дисперсию из всех возможных. Такое преобразование позволяет понижать информацию путем отбрасывания координат, соответствующих направлениям с минимальной дисперсией. Предполагается, что если нам надо отказаться от одного из базисных векторов, то лучше, если это будет тот вектор, вдоль которого набор входных данных меняется менее значительно.

В основе метода главных компонент лежат следующие допущения:

- Допущение о том, что размерность данных может быть понижена за счет линейного преобразования.
- Допущение о том, что больше всего информации несут те направления, в которых дисперсия входных данных максимальна.



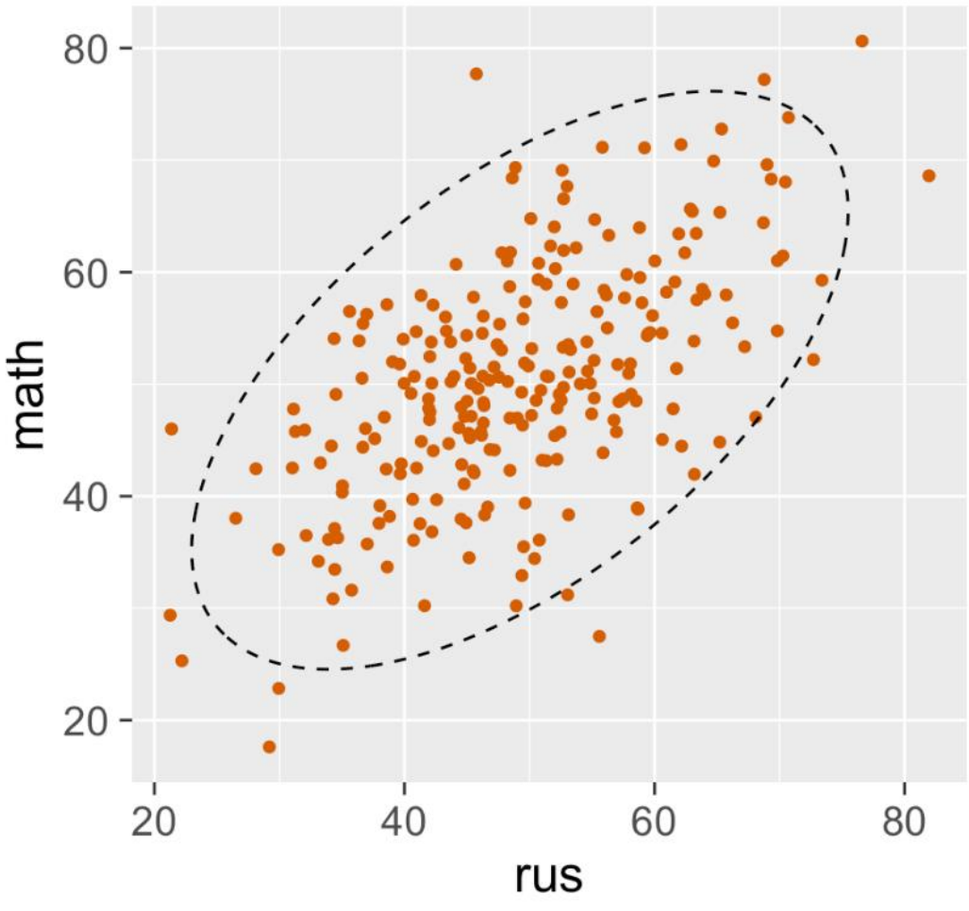
PRINCIPAL COMPONENTS ANALYSIS ИЛИ МЕТОД ГЛАВНЫХ КОМПОНЕНТ

Простой пример

Пусть у нас есть список школьников с оценками по русскому языку и математике:

```
##        rus     math
## 1 38.62011 33.67848
## 2 46.22913 54.53733
## 3 46.40963 38.32976
## 4 53.17011 51.07601
## 5 62.86754 65.64322
```

Эти оценки можно визуализировать на плоскости:



Оценки по математике и русскому языку могут коррелировать между собой.

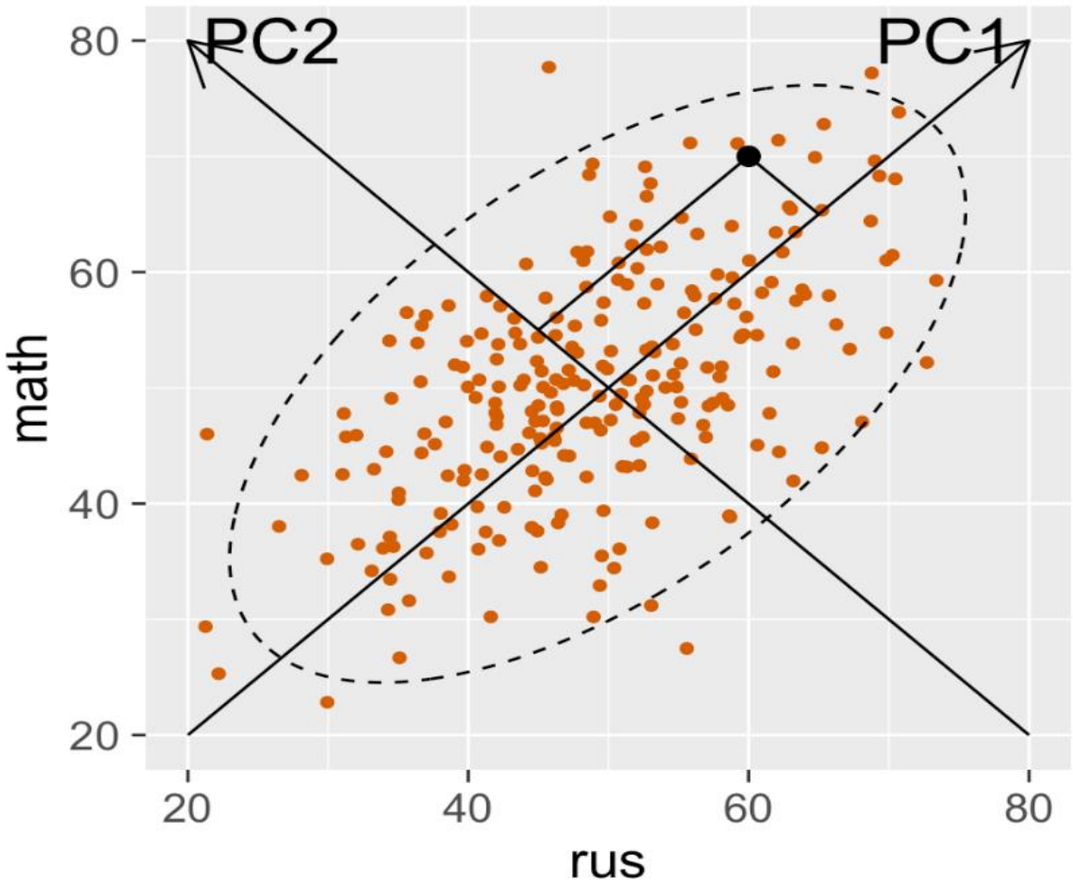
PRINCIPAL COMPONENTS ANALYSIS ИЛИ МЕТОД ГЛАВНЫХ КОМПОНЕНТ

Мы можем ввести новые оси на этой же плоскости как линейные комбинации старых осей.

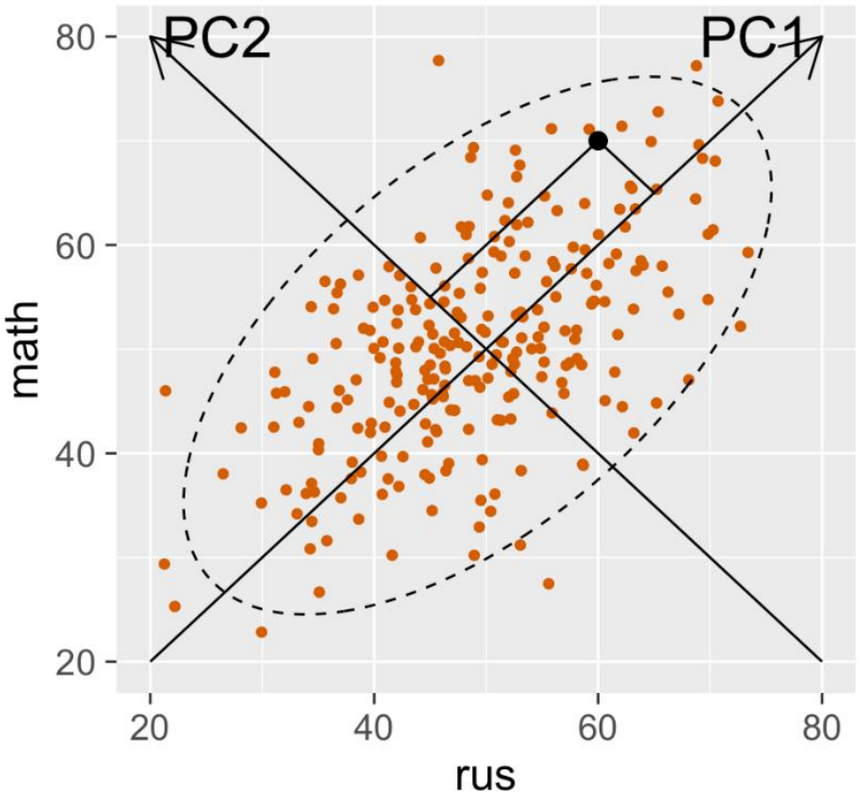
$$PC_1 = rus + math$$

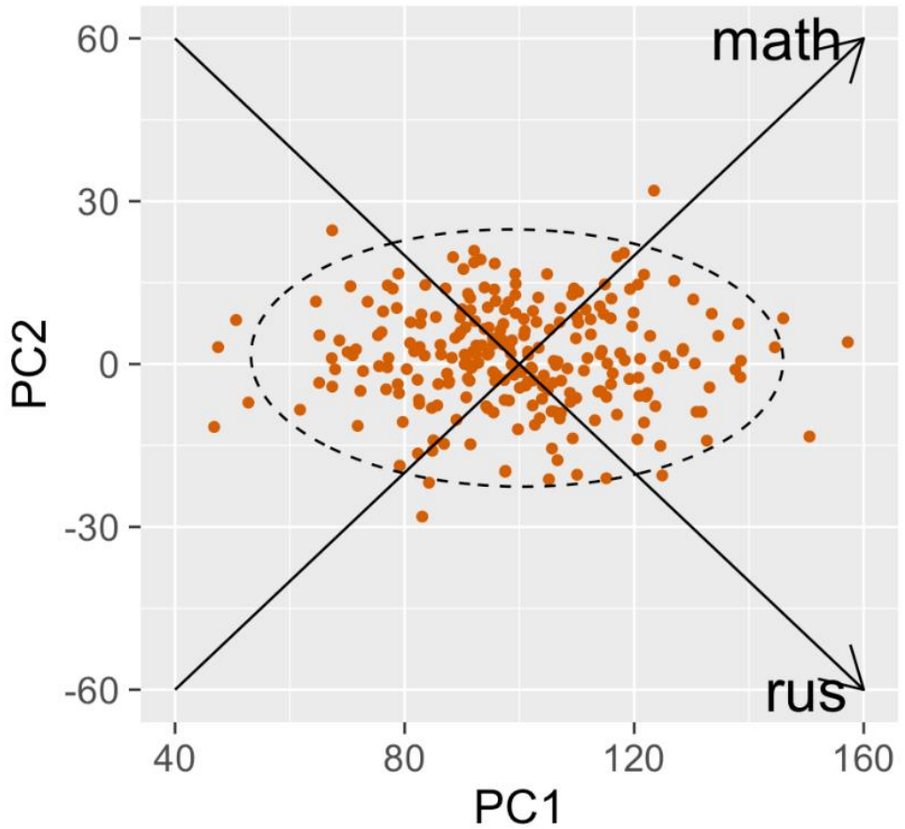
$$PC_2 = math - rus$$

В этом случае наши оси будут выглядеть вот так:
Новые оси, так же перпендикулярны друг другу. Однако первая ось проходит через область максимального разброса оценок, вторая ось проходит через оставшуюся часть максимального разброса оценок.



PRINCIPAL COMPONENTS ANALYSIS ИЛИ МЕТОД ГЛАВНЫХ КОМПОНЕНТ





В рамках старых осей, мы могли судить про корреляцию между оценками math и rus, то в новых осях это сделать уже сложно. Координата PC_1 называется ***первой главной компонентой***, а PC_2 — ***второй***. Заметим, что «главных компонент» получилось столько же, сколько изначально было переменных, но, и что PC_1 «главнее» (содержит больше информации), чем PC_2. **Таким образом, идея метода заключается в преобразовании старых компонент в новые, при этом происходит ранжирования новых компонент по степени объяснения дисперсии.**

АЛГОРИТМ МОДЕЛИ

$f_1(x), \ldots, f_n(x)$ — исходные числовые признаки;
$g_1(x), \ldots, g_m(x)$ — новые числовые признаки, $m \leqslant n$;

Требование: старые признаки должны линейно восстанавливаться по новым:

$$\hat{f}_j(x) = \sum_{s=1}^{m} g_s(x) u_{js}, \quad j = 1, \ldots, n, \quad \forall x \in X,$$

как можно точнее на обучающей выборке $x_1, \ldots, x_\ell$:

$$\sum_{i=1}^{\ell} \sum_{j=1}^{n} \left(\hat{f}_j(x_i) - f_j(x_i)\right)^2 \to \min_{\{g_s(x_i)\},\{u_{js}\}}$$

АЛГОРИТМ МОДЕЛИ

Матрицы «объекты–признаки», старая и новая:

$$\underset{\ell\times n}{F} = \begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_n(x_\ell) \end{pmatrix}; \quad \underset{\ell\times m}{G} = \begin{pmatrix} g_1(x_1) & \dots & g_m(x_1) \\ \dots & \dots & \dots \\ g_1(x_\ell) & \dots & g_m(x_\ell) \end{pmatrix}.$$

Матрица линейного преобразования новых признаков в старые:

$$\underset{n\times m}{U} = \begin{pmatrix} u_{11} & \dots & u_{1m} \\ \dots & \dots & \dots \\ u_{n1} & \dots & u_{nm} \end{pmatrix}; \qquad \hat{F} = GU^{\mathsf{T}} \overset{\text{хотим}}{\approx} F.$$

Найти: и новые признаки G, и преобразование U:

$$\sum_{i=1}^{\ell}\sum_{j=1}^{n}\left(\hat{f}_j(x_i) - f_j(x_i)\right)^2 = \left\|GU^{\mathsf{T}} - F\right\|^2 \to \min_{G,U},$$

АЛГОРИТМ МОДЕЛИ

$$\sum_{i=1}^{\ell}\sum_{j=1}^{n}\left(\hat{f}_j(x_i)-f_j(x_i)\right)^2=\left\|GU^{\mathsf{T}}-F\right\|^2\to\min_{G,U},$$

Можно показать, что если вычислить собственные вектора матрицы F, потом составить матрицу U из этих векторов у которых максимальные собственные значения. Далее рассчитать матрицу G=FU, то тогда мы получим минимум. И таким образом получим решение задачи.

Подробности можно найти в видеолекции Воронцова К.

https://www.youtube.com/watch?v=wcJ0nSUr7ws

или вот тут
http://www.machinelearning.ru/wiki/images/archive/a/a2/20150509140209%21Voron-ML-regression-slides.pdf

PRINCIPAL COMPONENTS ANALYSIS ИЛИ МЕТОД ГЛАВНЫХ КОМПОНЕНТ

Проблема интерпретации новых осей.

В рассмотренном примере старые оси имели смысл оценок по двум разным предметам. Новые оси это разности и суммы оценок.

В случае множества осей, новые оси выражаются через линейную комбинацию старых осей, где вклад каждой оси определяется коэффициентом.

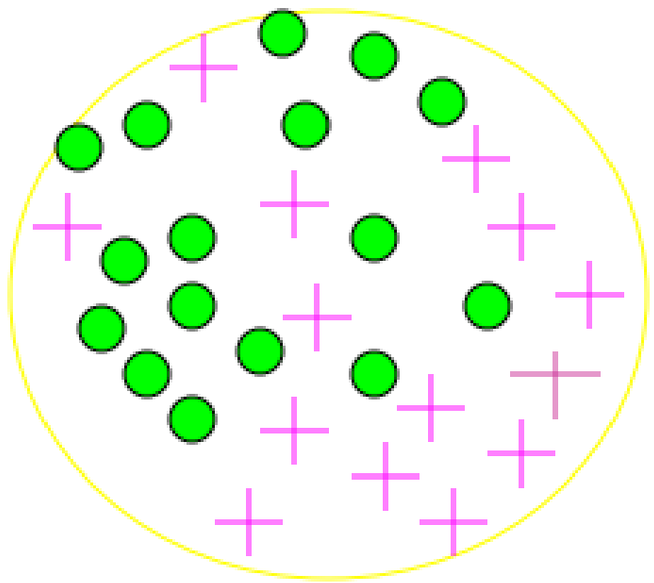
X – «старые объекты-признаки», G – « новые объекты-признаки»,
U – матрица перехода.

Таким образом, новые оси представляют собой смесь старых факторов (осей), соответственно, интерпретация новых осей вызывает проблему.

https://ru.coursera.org/learn/vvedenie-mashinnoe-obuchenie/lecture/Ey95r/mietod-ghlavnykh-komponient

ENTROPY

- Entropy = $\sum_i - p_i \log_2 p_i$



p_i is the probability of class i

Compute it as the proportion of class i in the set.

16/30 are green circles; 14/30 are pink crosses

$\log_2(16/30) = -.9$; $\log_2(14/30) = -1.1$

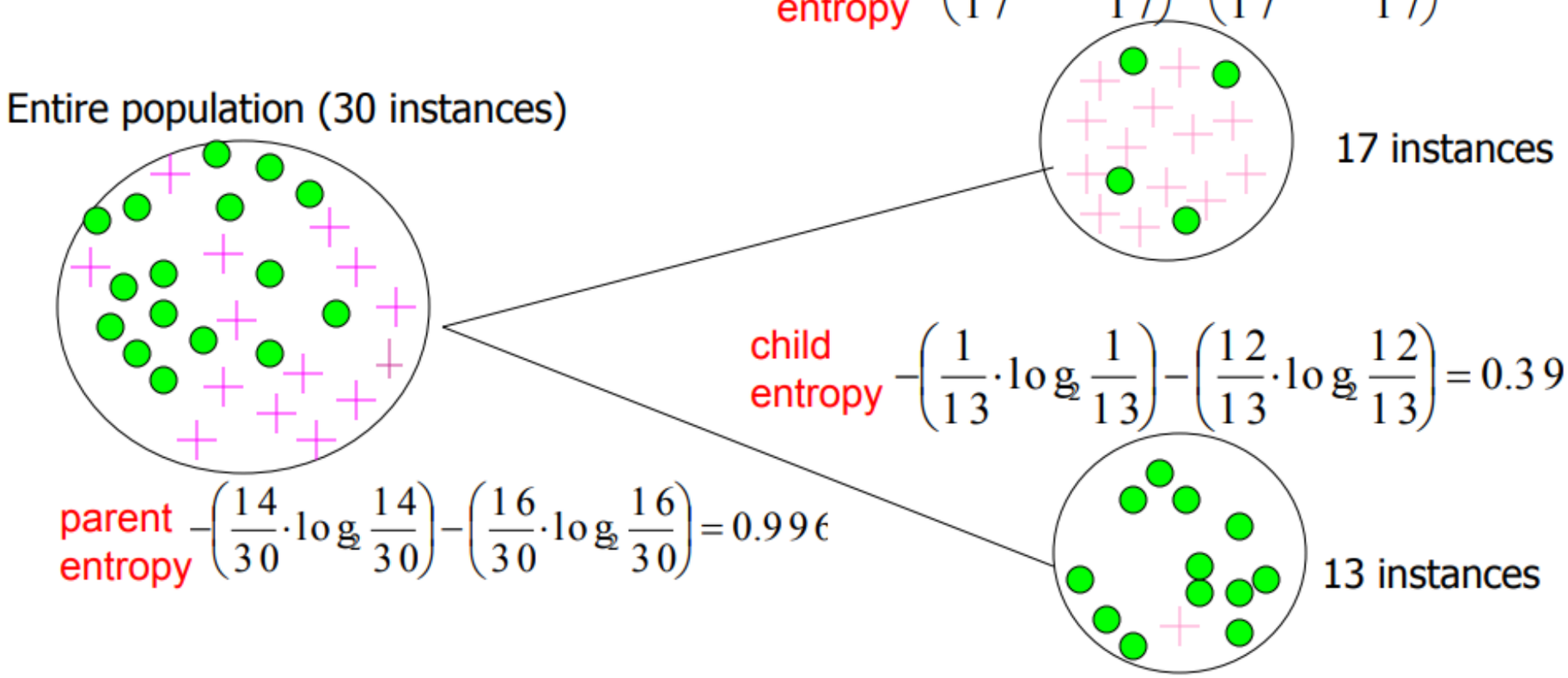
Entropy = -(16/30)(-.9) –(14/30)(-1.1) = .99

INFORMATION GAIN

Calculating Information Gain

Information Gain = entropy(parent) – [average entropy(children)]

child entropy $-\left(\frac{13}{17}\cdot\log_2\frac{13}{17}\right)-\left(\frac{4}{17}\cdot\log_2\frac{4}{17}\right)=0.78$



child entropy $-\left(\frac{1}{13}\cdot\log_2\frac{1}{13}\right)-\left(\frac{12}{13}\cdot\log_2\frac{12}{13}\right)=0.39$

parent entropy $-\left(\frac{14}{30}\cdot\log_2\frac{14}{30}\right)-\left(\frac{16}{30}\cdot\log_2\frac{16}{30}\right)=0.996$

(Weighted) Average Entropy of Children = $\left(\frac{17}{30}\cdot 0.787\right)+\left(\frac{13}{30}\cdot 0.391\right)=0.615$

Information Gain= 0.996 - 0.615 = 0.38 for this split

GINI INDEX

The Gini index is defined as:

$$\text{Gini} = 1 - \sum_{i=1}^{K} {p_k}^2$$

where p_k denotes the proportion of instances belonging to class k $(K = 1, \dots, k)$.

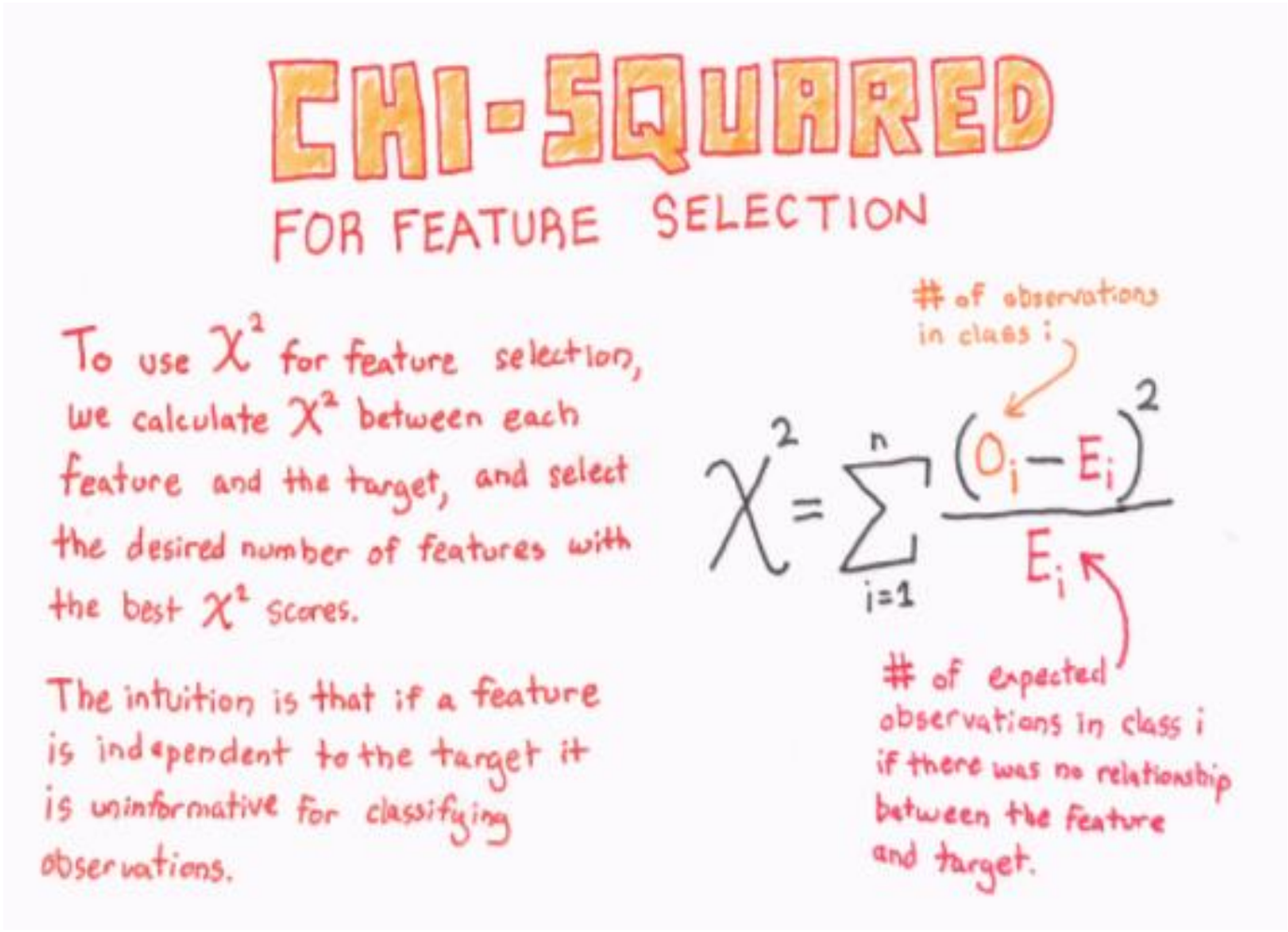
Suppose we want to split on the first variable (x_1):

x_1	1	2	3	4	5	6	7	8
y	0	0	0	1	1	1	1	1

$$Gini = 1 - \left({}^{3}/_{8}\right)^2 - \left({}^{5}/_{8}\right)^2 = {}^{15}/_{32}$$

Пусть у нас есть датасет, где одна колонка это значения целевой переменной, а остальные колонки это признаки (фиич). Можно попытаться вычислить насколько целевая переменная зависит от того или иного признака. Как это можно сделать.

Можно рассчитать индивидуальную величину **chi-square statistic,** для разных фич. То есть берем наблюдаемую величину (например, тональную оценку текста или стоимость квартиры) и значение фичи для данной оценки. Считаем **chi-square.** Далее берем следующую фичу и снова считаем статистику.



В итоге перебираем все фичи и получаем набор score для каждой фичи. Сортируем и отбираем нужное количество фич из списка. Выбираем фичи с максимальными значениями Chi2. Основной недостаток заключается в том, что в алгоритме рассматривается каждая фича по отдельности.

ПРИМЕР ХИ – КВАДРАТ (ЭКСПЕРИМЕНТЫ НА МЫШКАХ)

Всего в эксперименте было задействовано 111 мышей, которых мы разделили на две группы, включающие 57 и 54 животных соответственно. Первой группе мышей сделали инъекции патогенных бактерий с последующим введением сыворотки крови, содержащей антитела против этих бактерий. Животные из второй группы служили контролем – им сделали только бактериальные инъекции. После некоторого времени инкубации оказалось, что 38 мышей погибли, а 73 выжили.

Группа	Погибло	Выжило	**Всего**
Бактерии + сыворотка	13	44	**57**
Только бактерии	25	29	**54**
Всего	**38**	**73**	**111**

В эксперименте всего погибло 38 мышей, что составляет 34.2% от общего числа задействованных животных. Если введение антител не влияет на выживаемость мышей, в обеих экспериментальных группах должен наблюдаться одинаковый процент смертности, а именно 34.2%.

Рассчитав, сколько составляет 34.2% от 57 и 54, получим **19.5** и **18.5**. Это и есть ожидаемые величины смертности в наших экспериментальных группах. Аналогичным образом рассчитываются и ожидаемые величины выживаемости: поскольку всего выжили 73 мыши, или 65.8% от общего их числа, то ожидаемые частоты выживаемости составят 37.5 и 35.5.

Группа	Погибло	Выжило	**Всего**
Бактерии + сыворотка	13	44	**57**
Только бактерии	25	29	**54**
Всего	**38**	**73**	**111**

Составим новую таблицу сопряженности

Группа	Погибшие	Выжившие	**Всего**
Бактерии + сыворотка	19.5	37.5	**57**
Только бактерии	18.5	35.5	**54**
Всего	**38**	**73**	**111**

$$\chi^2 = \sum \frac{(f_o - f_e)^2}{f_e}, \quad \chi^2 = (13-19.5)^2/19.5 + (44-37.5)^2/37.5 + (25-18.5)^2/18.5 + (29-35.5)^2/35.5 =$$

$$2.16 + 1.12 + 2.31 + 1.20 = 6.79$$

Критерий χи^2 говорит о том, насколько эмпирическая частота V и ожидаемая частота C отличаются друг от друга. Большое значение $\chi 2$ говорит о том, что неверна гипотеза о независимости случайных величин x_i и c_j (и, следовательно, признак x_i может быть полезным для классификации)

АЛГОРИТМ RELIEF

Пусть у нас дан объекта из обучающей выборки x_k, $k = \overline{1, m}$, и пусть его класс c_j, $j = \overline{1, M}$. Введем функции *near-hit* (ближайшее попадание) и *near-miss* (ближайший промах) как расстояние до ближайшего объекта обучающей выборки с классом c_j и как расстояние до ближайшего объекта обучающей выборки с классом, отличным от c_j, соответственно.

Идея этого метода в том, что , предположительно, признак тем более релевантен объекту x_k, чем лучше он разделяет x_k и его near-miss, и чем хуже он разделяет x_k и его near-hit.

Алгоритм работает следующим образом:

- случайным образом выбирается P объектов обучающей выборки;
- для каждого объекта вычисляется его near-hit и near-miss (для этого используется евклидово расстояние);
- вычисляется и нормализуется вектор весов $W = (w_1, ..., w_n)$ для признаков по формуле

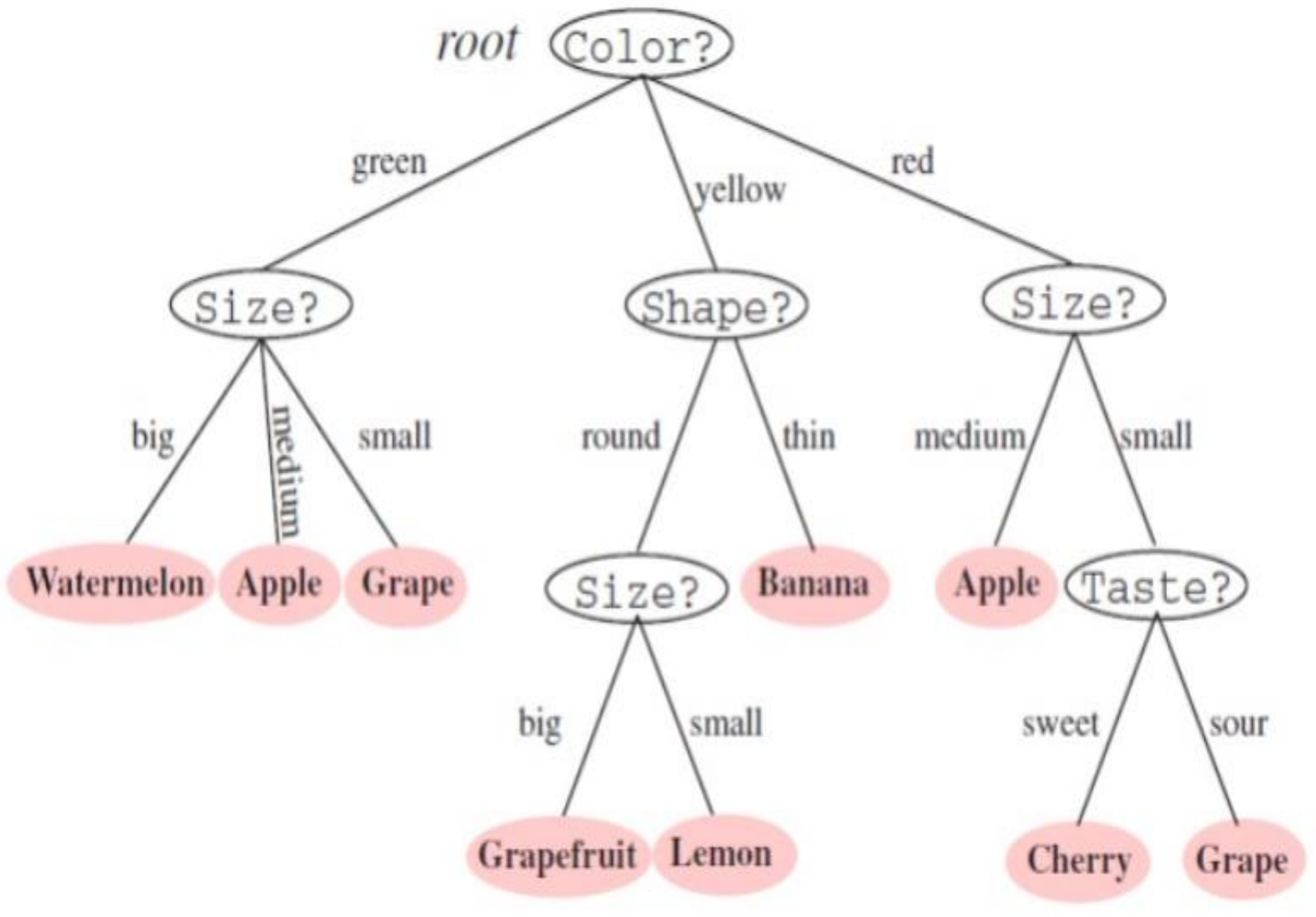
$$w_i = \sum_{k=1}^{p} \left(\delta(x_k^i, \text{near\_miss}(x_k)^i)^2 - \delta(x_k^i, \text{near\_hit}(x_k)^i)^2\right)$$

  Здесь $\delta(a, b)$ — символ Кронекера;
- отбираются те признаки x^i, вес w_i которых превышает заранее заданное значение порога τ.

Применение Random Forest для выделения фич

Random forest — алгоритм машинного обучения, заключающийся в использовании комитета (ансамбля) решающих деревьев для целей классификации.

Структура дерева представляет собой «листья» и «ветки». На ребрах («ветках») дерева решения записаны атрибуты, от которых зависит целевая функция, в «листьях» записаны значения целевой функции, а в остальных узлах — атрибуты, по которым различаются случаи. Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение.



Применение Random Forest для выделения фич

Как это работает?

1. Датасет разбивается на несколько кусочков.
2. Берем много деревьев (для каждого маленького датасета), на каждом уровне деревьев выбираем случайно разное количество фич.
3. Перебираем разные уровни для разных деревьев. В итоге получаем набор деревьев, которые как то (с разной степенью качества) предсказывают значения независимой переменной. Поиск производится исходя из требования снижения среднего индекса неоднородности в выборках.
4. Смотрим какие фичи чаще всего были использованы при правильном предсказании.
5. Рассчитываем величину ‘importance’

Gini Importance or Mean Decrease in Impurity (MDI) calculates each feature importance as the sum over the number of splits (across all tress) that include the feature, proportionally to the number of samples it splits.

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

https://linis.hse.ru/

Phone: +7 (911) 981 9165

Email: skoltsov@hse.ru