In [1]:
```
import IPython.display
IPython.display.display_latex(IPython.display.Latex(filename="../macros.tex"))
```

Работа с признаками

параметризация

minmax scaling

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

standart scaling

$$x_{scaled} = \frac{x - \mu}{\sigma}$$

пропуски в данных

- Удалить объекты с пропусками
- Удалить характеристики с пропусками
- Разделить выборку на 2 с пропусками и без
- Обучить алгоритм МО предсказывать попуски
- Усреднить: среднее, медиана, мода
- "специальное" значение (например, -1 если характеристика $[0, \infty]$)
- Заменить случайным значением
- Заменить наиболее вероятным значением (мат ожиданием)
- Взять следующее или предыдущее значение (временные ряды)
- Использовать алгоритм который может работать с пропусками

Категориальные признаки

Нет возможности для:
”<”,” + ”,” * ”

Binary coding (dummy, one-hot)

$f_i : \hat{X} \to D$
имеем $f_i : X \to D$

для каждого $u \in D$ заводим $f_{i,u}(x) = [f_i(x) = u]$

- новые $u \in D$ игнорируются
- быстро растет размерность пространства признаков
- ограничения на пространство признаков

Нумерация

$f_i : \hat{X} \to D$

просто дать номер каждой категориальной фиче.

$|D|!$ - количество нумераций (какую использовать?)

Счетчики

$Y = \{0, 1\}$

$f^j : \hat{X} \to D$

для каждого $u_k \in D$ вычислить

$$c(u_k) = p(y = 1 \mid x^j = u_k) = \frac{\sum_{i=1}^{N} [x_i^j = u_k][y = 1]}{\sum_{i=1}^{N} [x_i^j = u_k]}$$

$counters(x_i^j) = c(x_i^j)$

Overfitting $\to$ CV

существует много редких, например, основанных на частоте совместного появления:

$$x_i^{j_1 j_2} = \frac{1}{N} \sum_{k=1}^{N} [x_i^{j_1} = x_k^{j_1}][x_i^{j_2} = x_k^{j_2}]$$

использовать алгоритм который работает с категориальными признаками.

Отбор признаков

проверить все комбинации

- 2^M
- лучшую комбинацию

порог разброса

Если разброс фичи меньше порога не используем такую фичу.

RFA(ranking feature adding) RFE(ranking feature elimination)

Мы добавляем (удаляем) каждую фичу и считаем метрику.

Основанные на add-del:

WHILE (Q decreases) DO:
    RFA
    RFE

PCA (principal component analysis) **метод главных компонент**

Имеем X size $[M, N]$

$G = X * U$ - новая фича-матрица

$$\left|\left|G * U^T - X\right|\right| \to min$$

Признаки в U это собственные вектора $X^T * X$ (главные компоненты).

Собственные значения это мера вложения в "информативность". Мы можем расположить вектора в порядке информативности.

В U фичи ортоганальны (нет кореляции).

Количество главных компонент $m \leq rank(X)$

If $m = M$:

$$\left|\left|G * U^T - X\right|\right| = 0$$

If $m \leq M$:

$$X \approx G * U^T$$

Главные компоненты содержат базовую информацию о матрице X. Количество главных компонент называется *эффективная размерность задачи*.

Пусть выполняется для собственных значений:

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_M$$

тогда:

$$E(m) = \frac{\lambda_{m+1} + \lambda_{m+2} + \ldots + \lambda_M}{\lambda_1 + \lambda_2 + \ldots + \lambda_M}$$

$E(m)$ - сколько информации теряется

Вариант выбора m:

$$E(m-1) \ll E(m)$$

PCA

- новые фичи нескоррелированны
- уменьшает количество признаков
- можно использовать для визуализации
- ранжирования
- новые признаки не интерпретируемы