Классификация текстов.

Задача. Есть конечное мн-во классов:

1) {спам / не спам}
2) {полож / отриц} – анализ тональности
3) {Толстой, Достоевский, Донцова, ...} – опр. авторы
4) {спорт, политика, общество, ...} – для новостей

по тексту надо определить класс

Дан обучающий корпус – набор текстов, для которых класс уже известен. (это обучение с учителем).

На основе каких данных определять класс?

Features – признаки. – информация, на основе которой происходит классификация.

1. Тип признаков: мешок слов (bag of words)
в общем случае

f_i – i-ый признак

$f_i(text)$ – число, вектор, эл-т конечного мн-ва.

Есть словарь всех возможных слов $V = \{w_1, w_2 \dots w_n\}$

1) $f_i(text) = \begin{cases} 1, & \text{если слово } w_i \text{ встречается} \\ 0, & \text{если нет} \end{cases}$

2) $f_i(text)$ = сколько раз слово w_i встретилось в тексте

text = "Я хочу спать". $V = \{$Я, кот, кровать, спать, хочу$\}$
w_1 w_2 w_3 w_4 w_5

В версии 1)

$f_1(text) = 1$ $f_2(text) = 0$ $f_3(text) = 0$ $f_4(text) = 1$ $f_5(text) = 1$
Я кот кров спать хочу

или $f(text) = (1\ 0\ 0\ 1\ 1)$
– вектор признаков

В версии 2

f("я хочу спать, спать, спать") = (1 0 0 3 1)

3) tf-idf. $f_i(text) = tf \cdot idf$, где

tf = term frequency = сколько раз встретилось слово w_i

idf = inverse document frequency

$$= \frac{1}{\log(1 + \text{кол-во документов со словом } w_i)}$$

tf-idf слова "он" = $\frac{tf(\text{он})}{\log(1 + \text{всего док})}$ } много } мало

Это bag-of words признаки, т.к. игнорируется порядок слов

n-граммы признаки:

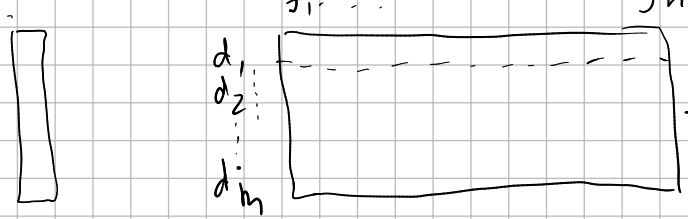
f(text) - сколько раз в тексте встречается определ. n-грамма.

Итого, $\exists$ f - вектор признаков, ф-ия считает сразу все признаки. $f(text) = (f_1(text), f_2(text), \dots)$

Вектор признаков обычно очень длинный. Может быть 10 000 000 признаков.

- сделать вектора короче: задача уменьшения размерности.

Что есть:



m обучающих документов

n признаков

$x_j^{(i)}$ — это j-ый признак i-го документа.

$y^{(i)}$ — это правильные классы для документа d_i $(x^{(i)})$

Для группы методов обучения — **линейные методы**

$\hat{y}^{(i)}$ — предсказание.

Алгоритм определения класса $\hat{y}^{(i)}$ для $x^{(i)}$.

для класса y_j: $\theta_1^{(j)} x_1^{(i)} + \theta_2 x_2^{(i)} + \theta_3 x_3^{(i)} + \dots + \theta_h^{(j)} x_h^{(i)} = x^{(i)} \cdot \theta^{(j)}$

Дан $x^{(i)}$

Вычисляем для каждого класса (j) это значение, выбираем (j), где значение максимально.

Пример. Спам / Не спам

$\theta^{(0)}$ $\theta^{(1)}$

V = {кушите, привет, задача, спам}

$\theta^{(0)} = \begin{pmatrix} 2 \\ -0.1 \\ -2 \\ 1 \end{pmatrix}$ $\theta^{(1)} = \begin{pmatrix} -2 \\ 0.1 \\ 2 \\ -0.5 \end{pmatrix}$ ← ≈ обратные (Н-ки, не не обязательно.

∃ text = "привет, это спам". f_i — 1/0 признак

$f(text) = (0, 1, 0, 1)$

для класса 0: $f(text) \cdot \theta^{(0)} = 0 \cdot 2 - 1 \cdot 0.1 - 0 \cdot 2 + 1 \cdot 1 = 0.9$

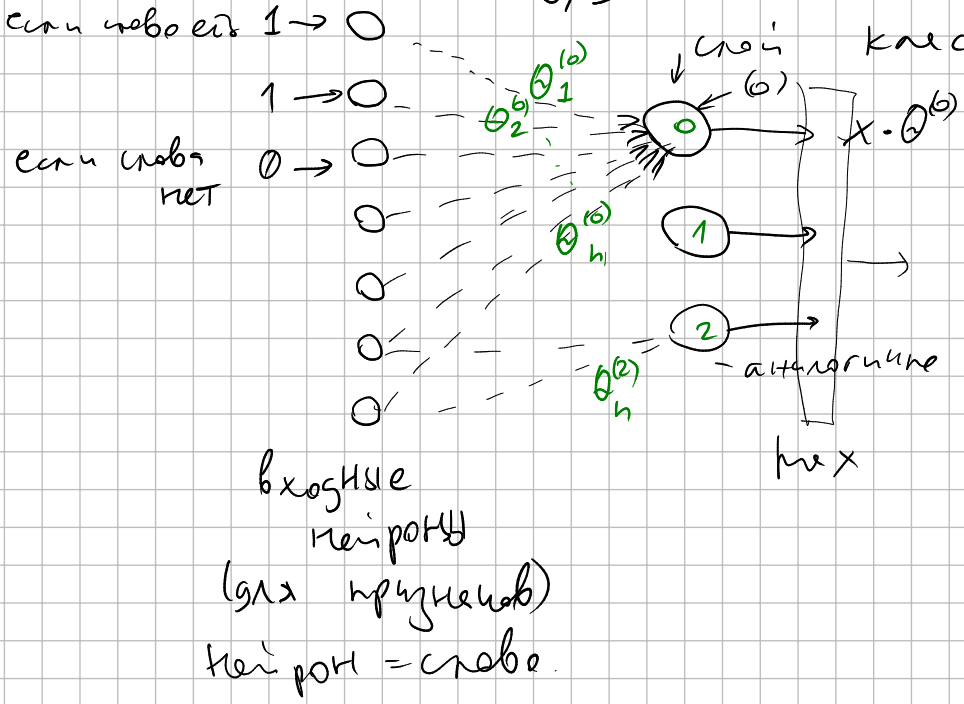
для класса 1: $f(text) \cdot \theta^{(1)} = 0 \cdot (-2) + 1 \cdot 0.1 + 0 \cdot 2 - 1 \cdot 0.5 = -0.4$

$0.9 > 0.4 \Rightarrow$ класс 0 $\Rightarrow$ спам.

Получается, что для определения класса нужно знать θ. Обучение — это поиск θ.

Разные способы обучения — разные способы искать θ.

Замечание. Увидим нейросеть в том, что мы обсуждаем. Текст подаётся на входной слой



Нейрон каждый входной сигнал умножает на вес → складывает все сигналы с весом

входные нейроны (для признаков)

нейрон = слово

Наивный Байесовский классификатор

Naïve

Считаем, что текст появляется случайным образом, т.е. это результат случайного процесса.

$p(y|text) = ?$ — это задача классификации.

Процесс:

Шаг 1 выбираем случайно один из классов в соответствии с априорным распределением.

Например, считаем, что спам : 5%

не спам : 95%

(мы будем искать эти в-ти, важно, что мы считаем, что они есть)

Шаг 2. Для каждого класса есть распределение слов. Например,

в классе "спам" слова распределены так:

привет	купите	он	спам
30%	20%	10%	40%

= 100%

В классе "не спам"

привет	купите	он	спам
60%	5%	30%	5%

= 100%

для каждого слова случайно выбирается, есть оно или нет.

Итак "текст" сгенерирован. Смотрим на текст получаем сколько класс:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$p\{y^{(0)} \mid text\}$

$p\{y^{(1)} \mid text\}$

что больше — ответ.

$$p\{y^{(0)} \mid text\} = \frac{p\{text \mid y^{(0)}\} \cdot p\{y^{(0)}\}}{p\{text\}}$$

← знаю в схеме знаю

$$p\{y^{(1)} \mid text\} = \frac{p\{text \mid y^{(1)}\}\, p\{y^{(1)}\}}{p\{text\}}$$

→ max

для выбора max можно не делить на $p\{text\}$.

Напомним Марковский класс.

Текст → $x = \overbrace{1\ 0\ 0\ 0\ 0\ 0}^{a_1\ a_2\ a_3}\ 1\ 0\ 1\ 0\ 0\ 0\ \overbrace{0}^{a_N}$

класс 1 (спам) → $x \cdot \Theta^{(1)}$

класс (не спам) → $x \cdot \Theta^{(2)}$

класс 3 → $x \cdot \Theta^{(3)}$

max

если text = $w_1, w_2 \dots w_N$

$$p(y^{(i)} \mid text) \sim p(text \mid y^{(i)}) \cdot p(y^{(i)}) \overset{\text{наивный}}{=\!=} p(w_1 \mid y^{(i)}) \cdot p(w_2 \mid y^{(i)}) \cdot \dots \cdot p(w_N \mid y^{(i)}) \cdot p(y^{(i)})$$

вероятности $p(w \mid y^{(i)})$ и $p(y^{(i)})$ можно оценить.

$$p(w \mid y^{(i)}) = \frac{\text{сколько раз встречалось слово } w \text{ в классе } i}{\text{всего слов в текстах класса } i}$$

$$p(y^{(i)}) = \frac{\text{сколько текстов класса } i}{\text{сколько всего текстов.}}$$

1) умножать вероятности плохо - underflow.

$$-\log p(y^{(i)} \mid text) = -\log(w_1 \mid y^{(i)}) - \log(w_2 \mid y^{(i)}) - \dots - \log(w_N \mid y^{(i)}) - \log p(y^{(i)}).$$

слово 1 | слово 2 | ... | слово N | N слов в словаре

т.е. $\Theta^{(i)}$: $[-\log p(w_1 \mid y^{(i)}) \quad -\log p(w_2 \mid y^{(i)}) \quad \dots \quad -\log p(y^{(i)})]$

2. сглаживание. $p(w_i \mid y^{(i)}) = \frac{c(w_i, y^{(i)}) + \alpha}{\text{всего слов в классе } i + \alpha \cdot N}$

$c(w_i, y^{(i)})$ — сколько раз; $\alpha = 0.1$; N — слов в словаре

Подбор α - поиск по сетке $\alpha = 0.01, 0.1, 0.05, 0.2$

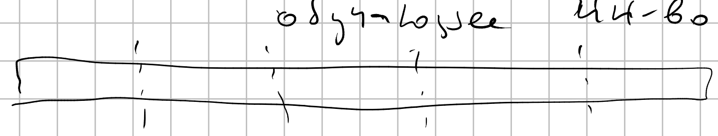
Что оптимизировать: точность $= \frac{\text{кол-во прав ответов}}{\text{общее кол-во вопросов}}$

все примеры

обучающее	valid: Настр.	Тест

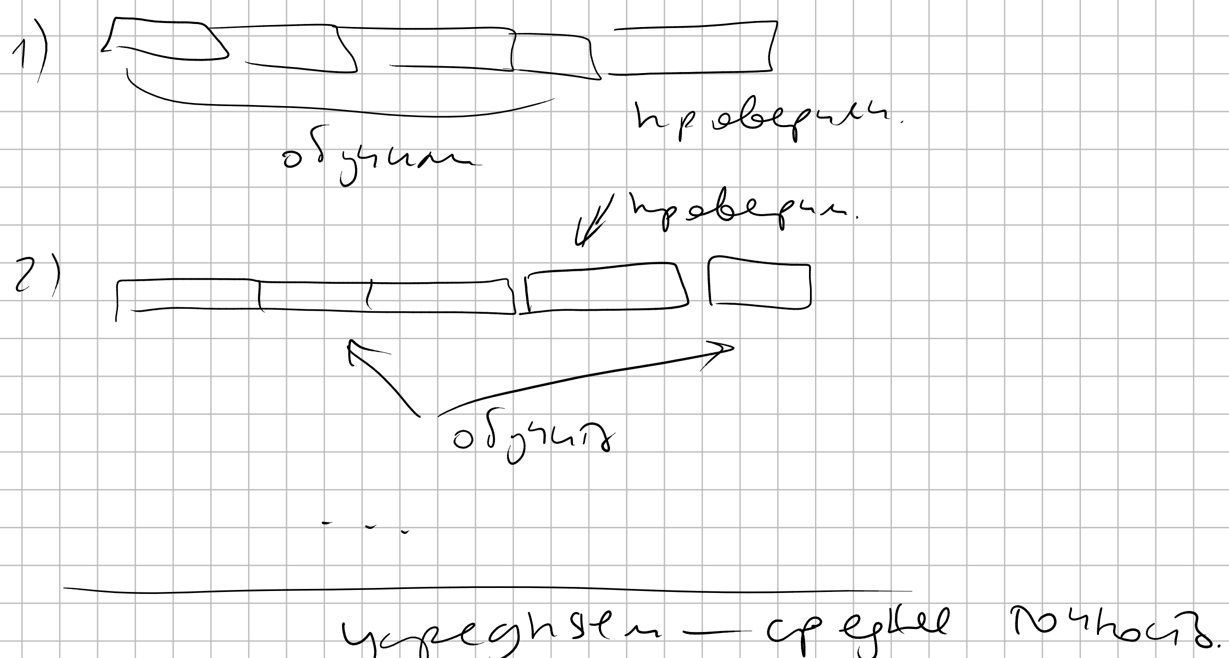
K-fold verification.

маленькое обучающее мн-во. Жалко выделять тестовое мн-во.



K=5 частей

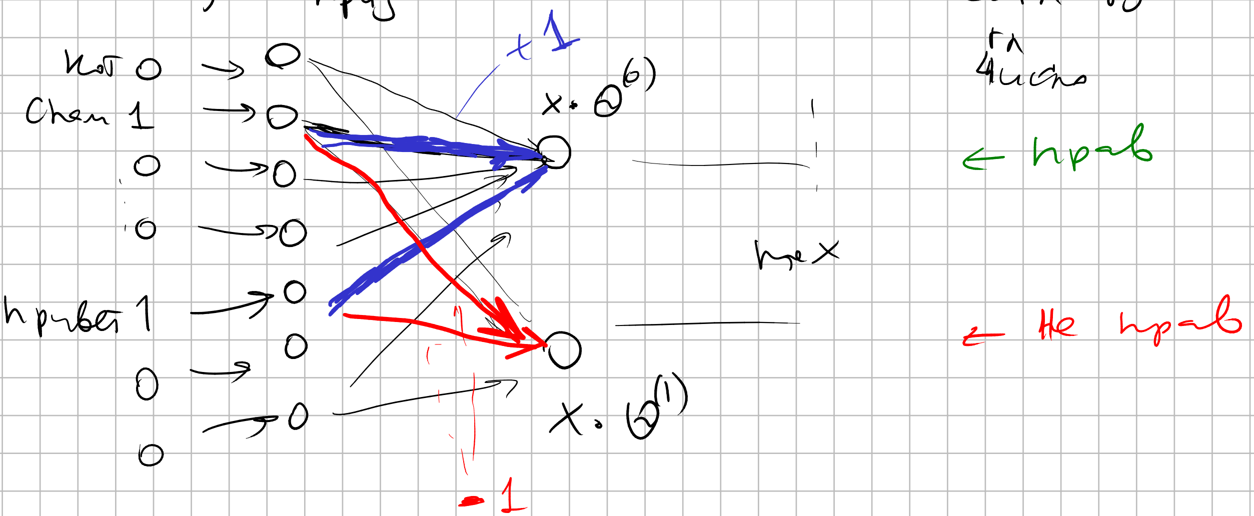
5 раз обучаем на всём кроме 1 кусочка.



Обучение перцептрона.

словам текста + доп признаки

доп. признак (приставка "пере-")
загл буквы
гл
числа



Как обучать.

1. сначала веса θ — маленькие случайные числа.

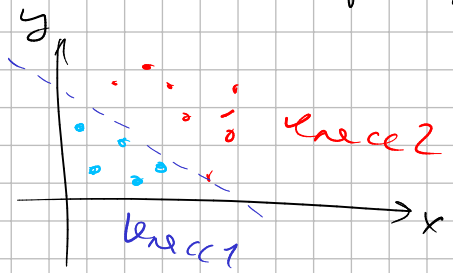
2. Постепенно даём примеры $x^{(i)}$ и знаем $y^{(i)}$ → прав. класс

3. подаём пример $x^{(i)}$ на себя, получаем класс $\hat{y}$.
4. если $y^{(i)} = \hat{y}$, идём дальше.
5. иначе уменьшаем веса не активировавших верных связей, увеличиваем — не активировавшихся верных.
6. повторяем, пока не кончатся примеры $x^{(i)}$

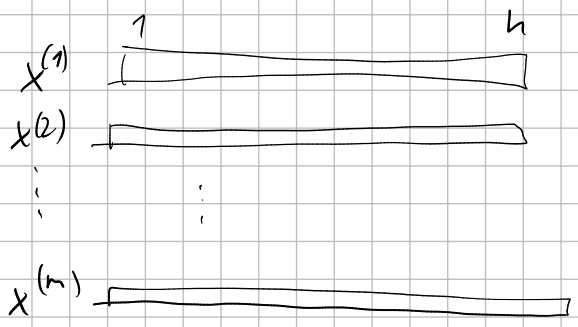
– Особенности алгоритма.

1) online-режим. Классифицируем по мере получения данных. Т.е. можно пользоваться до конца обучения.

2) Алгоритм сходится, если исх. мн-во примеров линейно разделимо



Логистическая регрессия



h предсказов $\rightarrow y^{(i)}$ – прав класса

$\rightarrow y^{(m)}$

все классы $y_1, y_2 \dots y_k$

пример x, класс y

$p(y|x)$ – вер-ть класса y при условии текста x

$$p(y|x) \approx \frac{e^{+x \cdot \theta^{(y)}}}{e^{+x\theta^{(y_1)}} + e^{+x \cdot \theta^{(y_2)}} + \dots + e^{+x\theta^{(y_k)}}} \in [0, 1]$$

$$\sum p(y_i|x) = 1$$

Что хотим оптимизировать Θ:

$$p(\text{все классы правильные}) = p(y^{(1)} \mid x^{(1)}) \cdot p(y^{(2)} \mid x^{(2)}) \dots p(y^{(m)} \mid x^{(m)})$$

$\longrightarrow$ max