Assignment 1

Bislan Ashinov

1 Naive Bayes Classifier

Обозначения:
d - Документ
c_j - Класс j
v_i - Слово i
k_i - Сколько раз встретилось слово i в документе
α - Параметр для сглаживания
M - Размер словаря
m - Количество классов

a)

$$P(v_i \in d | d \in c_j) = \frac{P(v_i \in d, d \in c_j)}{P(d \in c_j)} = \frac{\frac{\#\{d : v_i \in d, d \in c_j\}}{\#\{d\}}}{\frac{\#\{d : d \in c_j\}}{\#\{d\}}} = \frac{\#\{d : v_i \in d, d \in c_j\}}{\#\{d : d \in c_j\}} \tag{1}$$

b)

$$P(d = (k_1, k_2, ..., k_M) | d \in c_j) = \prod_{i=1}^{M} P(d : \#\{v_i\} = k_i | d \in c_j) = \prod_{i=1}^{M} \frac{\#\{d : \#\{v_i\} = k_i, d \in c_j\}}{\#\{d : d \in c_j\}} \tag{2}$$

С применением аддитивного сглаживания, превратим эту оценку в следующее:

$$\prod_{i=1}^{M} \frac{\alpha + \#\{d : \#\{v_i\} = k_i, d \in c_j\}}{\alpha * M + \#\{d : d \in c_j\}} \tag{3}$$

c)

$$P(d \in c_j | d = (k_1, k_2, ..., k_M)) = \frac{P(d = (k_1, k_2, ..., k_M) | d \in c_j) P(d \in c_j)}{P(d = (k_1, k_2, ..., k_M))} \tag{4}$$

Из пункта b) можем подставить формулу для $P(d = (k_1, k_2, ..., k_M) | d \in c_j)$

Тогда формула (3) примет вид:

$$\frac{P(d \in c_j)}{P(d = (k_1, k_2, ..., k_M))} \prod_{i=1}^{M} \frac{\alpha + \#\{d : \#\{v_i\} = k_i, d \in c_j\}}{\alpha * M + \#\{d : d \in c_j\}} \tag{5}$$

Введем полную систему событий и разложим $P(d = (k_1, k_2, ..., k_M))$ по формуле полной вероятности. $A_j = \{d \in c_j\}$. Очевидно, что события попарно

не пересекаются и в объединении образуют полную систему. Тогда

$$P(d = (k_1, k_2, ..., k_M)) = \sum_{j=1}^{m} P(A_j) P(d = (k_1, k_2, ..., k_M) | A_j) \tag{6}$$

$$\frac{\frac{\#\{d : d \in c_j\}}{N}}{\sum\limits_{l=1}^{m} \frac{\#\{d : d \in c_l\}}{N} P(d = (k_1, k_2, ..., k_M) | d \in c_l)} \prod_{i=1}^{M} \frac{\alpha + \#\{d : \#\{v_i\} = k_i, d \in c_j\}}{\alpha * M + \#\{d : d \in c_j\}} =$$

$$= \frac{\#\{d : d \in c_j\} \prod\limits_{i=1}^{M} \frac{\alpha + \#\{d : \#\{v_i\} = k_i, d \in c_j\}}{\alpha * M + \#\{d : d \in c_j\}}}{\sum\limits_{l=1}^{m} (\#\{d : d \in c_l\} \prod\limits_{i=1}^{M} \frac{\alpha + \#\{d : \#\{v_i\} = k_i, d \in c_l\}}{\alpha * M + \#\{d : d \in c_l\}})}$$

d) Классификатор выберет класс, вероятность которого при условии документа является наибольшей, то есть

$$\underset{j}{\text{argmax}}(P(d \in c_j | d = (k_1, k_2, ..., k_M))) \tag{7}$$

Возьмем полученную оценку из пункта c). Заметим, что в знаменателе всегда стоит сумма по всем классам, значит при выборе максимума из таких дробей можем выбирать максимум из числителей. (Еще раз показали, что вероятность документа не влияет на результат)

$$\underset{j}{\text{argmax}}(\#\{d : d \in c_j\} \prod_{i=1}^{M} \frac{\alpha + \#\{d : \#\{v_i\} = k_i, d \in c_j\}}{\alpha * M + \#\{d : d \in c_j\}}) \tag{8}$$

2 Multinomial Naive Bayes Classifier

Обозначения:
d - Документ
c_j - Класс j
v_i - Слово i
k_i - Сколько раз встретилось слово i в документе
α - Параметр для сглаживания
M - Размер словаря
m -Количество классов
N_d - Количество слов в документе
N - Количество документов в выборке C_{Nd} - Количество перестановок слов в документе

a)

$$P(v_i \in d | d \in c_j) = \frac{\sum\limits_{d \in c_j} cnt(v_i, d)}{\sum\limits_{d \in c_j} len(d)}, \tag{9}$$

Со сглаживанием:

$$P(v_i \in d | d \in c_j) = \frac{\alpha + \sum\limits_{d \in c_j} cnt(v_i, d)}{\alpha * M + \sum\limits_{d \in c_j} len(d)} \tag{10}$$

где $cnt(v_i, d)$ - количество слов v_i в документе d, а $len(d)$ - количество слов в d.

b)

$$P(d = (k_1, k_2, ..., k_m) | d \in c_j) = P(N) * C_{Nd} * \prod_{i=1}^{N_d} P(v_i | c_j), \tag{11}$$

где C_{Nd} - количество перестановок, равное полиномиальному коэффициенту

$$C_{Nd} = \frac{N_d!}{k_1! k_2! ... k_m!}$$

c)

$$P(d \in c_j | d = (k_1, k_2, ..., k_3)) = \frac{P(d = (k_1, k_2, ..., k_m) | d \in c_j) P(d \in c_j)}{P(d = (k_1, k_2, ..., k_3))} =$$

$$= \frac{\frac{\#\{d : d \in c_j\}}{N} P(N_d) * C * \prod\limits_{i=1}^{N_d} P(v_i | c_j)}{P(d)} = \frac{\frac{\#\{d : d \in c_j\}}{N} P(N_d) * C * \prod\limits_{i=1}^{N_d} \frac{\alpha + \sum\limits_{d \in c_j} cnt(v_i, d)}{\alpha * M + \sum\limits_{d \in c_j} len(d)}}{P(d)}$$

Разложим также P(d) по формуле полной вероятности с полной системой событий $A_j = d \in c_j$, вынесем за скобку в сумме $P(N_d)$, C_{Nd} и N и сократим:

$$\frac{\#\{d : d \in c_j\} * \prod\limits_{i=1}^{N_d} \frac{\alpha + \sum\limits_{d \in c_j} cnt(v_i, d)}{\alpha * M + \sum\limits_{d \in c_j} len(d)}}{\sum\limits_{l=1}^{m} \#\{d : d \in c_l\} \prod\limits_{i=1}^{N_d} \frac{\alpha + \sum\limits_{d \in c_l} cnt(v_i, d)}{\alpha * M + \sum\limits_{d \in c_l} len(d)}} \tag{12}$$

d) Классификатор выбирает класс с наибольшей вероятностью. Как и в модели Бернулли, достаточно оценить числитель дроби, так как знаменатель равен сумме по всем классам и для всех классов одинаков. Необходимо найти:

$$\underset{j}{\text{argmax}}(\#\{d : d \in c_j\} \prod_{i=1}^{M} \frac{\alpha + \sum\limits_{d \in c_l} cnt(v_i, d)}{\alpha * M + \sum\limits_{d \in c_l} len(d)}) \tag{13}$$

3 Практическая часть

d) Минимальные веса

Слова	Байесовские веса	Pos	Neg
’4/10’	-3.9327313682595886	0	49
’3/10’	-3.8048979967497036	0	43
’thunderbirds’	-3.6842700089610894	0	38
’boll’	-3.6180206234198877	1	72
’/>4/10’	-3.5470688874476046	0	33
’2/10’	-3.486444265631169	0	31
’beowulf’	-3.486444265631169	0	31
’ajay’	-3.421905744493598	0	29
’dahmer’	-3.421905744493598	0	29
’uwe’	-3.3880041928179168	1	57
’deathstalker’	-3.3529128730066464	0	27
’welch’	-3.3165452288357713	0	26
’tedium’	-3.278804900852924	0	25
’kareena’	-3.278804900852924	0	25
’seagal’	-3.2261611673675024	2	73
’turgid’	-3.1987621931793875	0	23
’*1/2’	-3.1987621931793875	0	23
’hobgoblins’	-3.1987621931793875	0	23
’sarne’	-3.1987621931793875	0	23
’palermo’	-3.1987621931793875	0	23
’/>3/10’	-3.156202578760592	0	22
’unwatchable’	-3.156202578760592	2	68
’grendel’	-3.156202578760592	0	22
’kibbutz’	-3.156202578760592	0	22
’maddy’	-3.1117508161897582	0	21
’dreck’	-3.1117508161897582	1	43
’kinski’	-3.1117508161897582	0	21
’slater’	-3.1117508161897582	0	21
’lordi’	-3.0652308005548665	0	20
’shaq’	-3.0652308005548665	0	20

Максимальные веса

Слова	Байесовские веса	Pos	Neg
’edie’	4.283356730372729	73	0
’antwone’	4.213398141765818	68	0
’gundam’	4.10642602221365	61	0
’paulie’	4.09016550134187	60	0
’mildred’	4.073636199390659	59	0
’corbett’	3.7404917528621215	42	0
’din’	3.6928637038728667	40	0
’flavia’	3.6681710912824954	39	0
’biko’	3.5902095498127835	36	0
’deathtrap’	3.5346396986579727	34	0
’trier’	3.4757991986350394	32	0
’ossessione’	3.445027539968285	31	0
’7/10’	3.429279183000146	62	1
’yokai’	3.413278841653705	30	0
’brashear’	3.3804890188307137	29	0
’mclaglen’	3.3804890188307137	29	0
’visconti’	3.3804890188307137	29	0
’/>7/10’	3.3804890188307137	29	0
’daisies’	3.3465874671550324	28	0
’creasy’	3.3465874671550324	28	0
’gino’	3.3465874671550324	28	0
’8/10’	3.3291957244431636	56	1
’sox’	3.3291957244431636	56	1
’venoms’	3.311496147343762	27	0
’rea’	3.311496147343762	27	0
’iturbi’	3.311496147343762	27	0
’ultimatum’	3.311496147343762	27	0
’warhols’	3.311496147343762	27	0
’gunga’	3.275128503172887	26	0
’tsui’	3.275128503172887	26	0

e)

Модель	Время обучения	train set	dev set	dev-b set
Bernully	8.17s	93.14	85.76	73.65
Multinomial	10.05s	92.41	85.68	73.55

f)Модели с биграммами дают более точный результат

Модель	train set	dev set	dev-b set
Bernully	99.54	88.12	74.90
Multinomial	99.39	87.98	75.30

При добавлении 3-грамм результат снова улучшил, но уже не так заметно на тестовых выборках, однако все еще заметно на валидационных

Модель	train set	dev set	dev-b set
Bernully	99.99	89.15	75.35
Multinomial	99.99	89.15	75.35

Также добавление 4-грамм дает результат точнее, но проверить мне это удалось лишь раз, так как мой компьютер очень тяжело переносит использование такого количества оперативной памяти. К сожалению, точные результаты у меня не сохранены, но именно эта модель была отослана мною в соревнование

19.10.2020. Также там были удалены униграммы, что также повысило точность.