9/12/2023: Naive Bayes

Classification Algorithm

Discriminative
- Logistic Regression
- Softmax Regression

Directly model $p(y|x)$

e.g. in logistic Regression

$$p(y=1 \mid x; w) = \sigma(w^T x)$$

Don't try to model $p(x)$

Generative
- Naive Bayes

Model joint distribution $p(x, y)$

$$p(x,y) = \boxed{p(y)}\ \boxed{p(x \mid y)}$$

prior distribution over labels

for a given label, what does a plausible x look like?

What to do at test time?

Given x, want to predict y

Bayes Rule:

$$p(y|x) = \frac{p(y)\, p(x|y)}{\boxed{p(x)}}$$

$$= \sum_{k=1}^{C} p(y=k)\, p(x \mid y=k)$$

Naive Bayes for text classification

input x is a document

Training Data

y	x
+1	great acting and score
-1	terrible directing
+1	great execution
-1	terrible
+1	amazing

Test time:

New review eg. "great directing"

Naive Bayes: Make a key simplifying assumption so that estimating $p(x|y)$ is not too difficult

$|V| = 8$

Parameters: $-\ p(y)$: $+1 = 3/5 = .6$
$-1 = 2/5 = .4$

- $p(x_j|+1)$: Distribution over 8 words
- $p(x_j|-1)$: Different distribution over 8 words

Naive Bayes Assumption: $p(x|y) = \prod_{j=1}^{d} p(x_j | y)$

(where x_j is j-th word of x)

① Parameters of model

- $P(y)$: If we have C classes, need parameter vector $\pi \in \mathbb{R}^C$ where $P(y=k) = \pi_k$
- $P(x|y)$: Because of NB assumption, we just have to model $p(x_j | y)$

Imagine C different dice
Each dice has $|V|$ sides

vocabulary ie set of possible words

use $\tau_{wk} = p(x_j = w | y=k)$ for some $w \in V$

② Learning parameters: Apply MLE to joint distribution

log-likelihood: $\sum_{i=1}^{n} \log P(x^{(i)}, y^{(i)}; \pi, \tau)$

$$= \underbrace{\left(\sum_{i=1}^{n} \log P(y^{(i)}; \pi, \tau) \right)}_{\text{Only depends on } \pi} + \underbrace{\sum_{i=1}^{n} \log P(x^{(i)} | y^{(i)}; \pi, \tau)}_{\text{Only depends on } \tau}$$

$$= \sum_{k=1}^{C} \text{count}(y=k) \log \pi_k$$

maximized when

$$\boxed{\pi_k = \frac{\text{Count}(y=k)}{n}}$$

For each label k, we have $|V|$-sided dice, estimate

$$P(x_j = w | y=k) = \frac{\text{Count}(w, y=k)}{\sum_{w' \in V} \text{count}(w', y=k)}$$

ie τ_{wk}

DO NOT USE THIS FORMULA

Training Data

y	x
+1	great acting and score
-1	terrible directing
+1	great execution
-1	terrible
+1	amazing

No Smoothing

$p(x_j = \text{"great"} \mid +1) = 2/7$

acting $: 1/7$

score $: 1/7$

$\vdots$

terrible $: \boxed{0/7}$

because it generates 0 probabilities

with Smoothing of $\lambda = 1$: $P(x_j = \text{"great"} \mid +1) = \frac{2+1}{7+8} = \frac{3}{15} = \frac{1}{5}$

"terrible" $= \frac{1}{15}$

Suppose: Test example $x =$ "great directing"

- compute $P(x, y=1) = P(y=1)P(x \mid y=1) = 3/5 \times 2/7 \times 0/7 = 0$
- compute $P(x, y=-1) : 2/5 \times 0/3 \times 1/3 = 0$
- By Bayes Rule, get $P(y \mid x) = \frac{0}{0}$

Fix: Laplace Smoothing

Imagine that every (word, label) pair was seen an additional $\boxed{\lambda}$ times

"pseudocounts"

(hyperparameter, > 0

Actual Formula: $$\tau_{wk} = \frac{\text{Count}(w, y=k) + \lambda}{\left(\sum_{w' \in V} \text{Count}(w', y=k)\right) + \underbrace{|V| \cdot \lambda}_{\text{total \# of imaginary counts that were added}}}$$

Avoiding Underflow

$x =$ "great directing but the music was ..."

$p(y) P(x \mid y) = p(y)\, p(\text{"great"} \mid y) \cdot p(\text{"directing"} \mid y) \cdots$

multiplying many small #'s on a computer leads to "underflow" ie answer is 0

Trick: Work in log space

ie: Don't compute $p(x|y)$

Instead compute $\log p(x|y) = \sum_{j=1}^{d} \log p(x_j | y)$

How to predict?

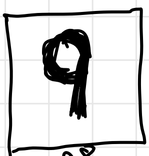
- For each label k, compute $\log P(y=k) + \log P(x | y=k)$
- k where this is largest also has largest $P(x,y) \iff$ largest $P(y=k | x)$

Naive Bayes for general feature vectors

- So far, x was a document with d words
- Now: x is a list/vector of d features

Black/white Image:

28



28

You have $28^2 = 784$ $\{0,1\}$ features

Song

Feature 1

genre → pop, classical, country...

Feature 2

artist

⋮

Key diff: Each feature means something different

Same: Naive Bayes Assumption $P(x|y) = \prod_{i=1}^{d} P(x_j | y)$

Different: $P(x_1 | y)$ different from $P(x_2 | y)$

e.g. Songs: Feature 1

$P(\text{genre} = \text{pop} | y=1)$ $\quad p(\text{genre}=\text{pop} | y=0)$

$P(\text{genre} = \text{rock} | y=1)$ $\quad p(\text{genre}=\text{rock} | y=0)$

⋮

Each one is parameter specific to feature 1