Classification Algorithm

Discriminative

- logistic regression
- softmax regression

Directly model $P(Y|X=x)$

e.g. logistic regression

$P(y=1|x;w) = \sigma(w^T x)$

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

Generative

- naive bayes

Jointly model $P(x,y)$

$P(x,y) = P(y)\,P(x|y)$

prior distribution over labels

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

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

normalizing constant

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

Naive Bayes (assume $x \in \mathbb{R}^d$)

The Naive Bayes assumption is : $P(x|y) = \prod_{j=1}^{d} P(x_j|y)$

all x_j's are conditionally independent given y

don't assume "independent"

$p(y=0) = p(y=1) = 0.5$

suppose $X \in \mathbb{R}^2$

$x_1, x_2 \in \{0,1\}$

$\underbrace{\begin{aligned} P(x_1=1|y=0) &= 0.9 \\ P(x_2=1|y=0) &= 0.8 \end{aligned}}_{P(x|y=0)}$

$\underbrace{\begin{aligned} P(x_1=1|y=1) &= 0.2 \\ P(x_2=1|y=1) &= 0.05 \end{aligned}}_{P(x|y=1)}$

If $x_1=1$, $y=0$ is more likely $\Rightarrow$ $x_2=1$ is more likely

Common case: $x_j \in \{0,1\} \ \forall j$

For this, we use multivariate Bernoulli Naive Bayes

many of these → dist. over $\{0,1\}$

Example 1: Black white images

28

28  → $28^2 = 784$-dim vector, each $\in \{0,1\}$

Example 2: Text classification

$$\text{input} = \text{document} \longrightarrow \left.\begin{array}{c|l} 1 & a \\ 0 & \text{aardvark} \\ \vdots & \vdots \\ 0 & \text{zebra} \end{array}\right\} \text{Vocabulary } V \text{ of size } |V|$$

does the word occur in the doc?

Parameters of Multivariate Bernoulli NB model

- $P(Y)$: Distribution over C classes ⇒

  $\boxed{\pi} \in \mathbb{R}^C$ where $p(y=k) = \pi_k$

  param 1

- $P(x_j | y=k) \ \forall j \in \{1, \dots, d\}$, $k \in \{1, \dots, C\}$ → each one is a Bernoulli

  so we have $\tau \in \mathbb{R}^{d \times C}$ where $p(x_j = 1 | y=k) = \tau_{jk}$

How to choose π and τ? Apply MLE

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

log likelihood

$$= \underbrace{\sum_{i=1}^{n} \underline{\log P(y^{(i)}; \pi)} + \log P(x^{(i)} | y^{(i)}; \tau)}_{\text{general form for generative classifier}}$$

$$\sum_{i=1}^{n} \log P(y^{(i)}; \pi)$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{C} I\{y^{(i)}=k\} \log \underbrace{P(y=k, \pi)}_{\pi_k}$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{C} I\{y^{(i)}=k\} \log \pi_k$$

let $count(y=k)$ means $\sum_{i=1}^{n} I\{y^{(i)}=k\}$

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

If $C=2$:

$$= count(y=1) \log \pi_1 + (1 - count(y=1)) \log(1-\pi_1)$$

From HW0: maximized when $\pi_1 = \frac{count(y=1)}{n}$

When $C>2$, MLE estimate for π is

$$\pi_k = \frac{count(y=k)}{n}$$

What about τ?

maximize $\sum_{i=1}^{n} \log P(x^{(i)} | y^{(i)}; \tau)$

By similar derivation,

$$\tau_{jk} = \frac{count(x_j=1, y=k)}{count(y=k)}$$

← Don't use this

$P(x_j=1 | y=k; \tau)$

$$\tau_{11} = P(x_1=1 | y=1) = \frac{1}{3}$$

$$\tau_{21} = P(x_2=1 | y=1) = \frac{2}{3}$$

Y	X_1	X_2
1	0	1
2	1	1
3	1	0
2	0	0
2	0	1
1	0	1
1	1	0

What happens when some counts are zero?

Text classification

- "giraffe" never occurs when $y=1$

- "choir" never occurs when $y=2$

if a document has "giraffe" and "choir"

$$p(x \mid y=1) = 0$$

$$p(x \mid y=2) = 0$$

assuming zero possibility for possibile event is BAD

Solution: Laplace smoothing ("pseudocounts")

pretend we've seen every (feature, label) pair $\boxed{\lambda}$ times

hyperparameter $\lambda = 1$ reasonable

Better formula for T_{jk}:

$$T_{jk} = \frac{\text{count}(y=k, x_j=1) + \lambda}{\text{count}(y=k) + 2\lambda}$$

once for $(y=k, x_j=1)$
once for $(y=k, x_j=0)$

If no training data, then

$$T_{jk} = \frac{1}{2}$$

with enough training data, ignore λ's

Another variant: Multinomial NB (for text classification)

Input $x^{(i)}$ is a document: list of words w/ length d.

By Naive Bayes assumption,

$$p(x^{(i)} \mid y^{(i)}) = \prod_{j=1}^{d_i} \underbrace{p(x_j^{(i)} \mid y^{(i)})}_{\text{multinomial distribution over } V \text{ (vocabulary)}}$$

Note: presenves frequency info

Additional assumption:

$p(x_j | y)$ is same for all j

Distribution of 1st word | y = Dist. of 27th word | y

doc = [dog dog dog ⋯]

$p(\text{dog} | y)^3$

Parameters

- $p(y)$ - same as before : π where $p(y=k) = \pi_k$

- $p(x_j | y=k)$ = Dist. over vocabulary V for each k

$$\underbrace{P_{uk}}_{u \in V,\ k \in \{1, \dots, c\}} = p(x_j = u | y = k)$$

To estimate best P_{uk}, we count

$$P_{uk} = \frac{\text{count}(x_j = u, y = k) + \lambda}{\sum_{i=1}^{n} I\{y^{(i)} = k\}\, d_i + |V|\lambda}$$

← add λ for every possible word in dictionary

d_i: # of words in i^{th} doc

whole denominator = # words that go with $y=k$