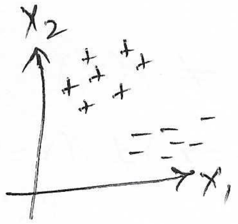
(15)

Naive Bayes

(our first probabilistic classifier)

I/P features: $\vec{x} \in \mathbb{R}^d$

O/P: $y \in \{1 \dots k\}$ or $y \in \{0, 1\}$ or $y \in \{-1, +1\}$

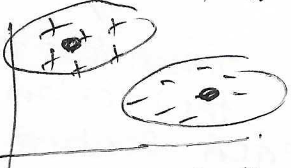


$$\hat{y} = \arg\max_y \, p(Y=y \mid X)$$

Generative approach

$$p(Y=y \mid X=x) = \frac{p(x \mid y)\, p(y)}{p(x)}$$

& estimate: $p(X=x \mid Y=y)$ & $p(Y=y)$



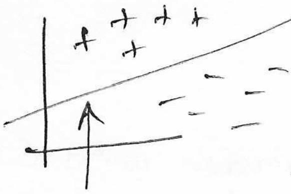
- Try to "explain" your data for two classes
- Can create new data by sampling

Naive Bayes

Discriminative approach

Directly estimate the posterior: $p(Y=y \mid X=x)$

or, estimate (non-probabilistic approach) directly $g: X \to Y$



$g(x)$

estimate g

SVM

No "explanation" of your data

Logistic regression

Consider binary features (for simplicity):

$x_1 \dots x_d \in \{0, 1\}$

$y \in \{1 \dots k\}$

We need to estimate $p(Y=y)$ & $p(X_1=x_1 \dots X_d=x_d \mid Y=y)$

probability distribution of length k that sums to 1

Massive table
# of parameters: $(2^d - 1)k$

If $d=100$, then $2^{100} = 10^{30} \gg$ available data

Huge number of parameters $\longrightarrow$ sparse table $\rightarrow$ hard to solve

The rescue is → assume independence.

⇓

Naive Bayes

Assumption:

$$\underbrace{P(X_1=x_1, \dots X_d = x_d \mid Y=y) = \prod_{j=1}^{d} P(X_j = x_j \mid Y=y)}_{\text{features are conditionally independent given the class.}}$$

$\Rightarrow$ $P(X_1 \mid X_2) \neq P(X_1)$

& $P(X_1=x_1, X_2=x_2) \neq P(X_1=x_1)\,P(X_2=x_2)$

(features are not independent)

$\boxed{\text{NB vs True Bayes}}$

Axis-aligned features.

Number of parameters in NB?

- for binary features, for each feature X_j & each class Y, we estimate 1 parameter (e.g. $P(X_j=1 \mid Y)$), as $P(X_j=0 \mid Y) = 1 - P(X_j=1 \mid Y)$. So for d binary features & k classes, total # of parameters = $\boxed{dk}$

much less than the previous case.

- for categorical features with c categories, for each feature X_j & each class Y, we estimate $(k-1)$ parameters. So for n categorical features with c categories each, & k classes, total # of parameters $= nk(c-1)$

$P(X_j = x_j \mid Y = c)$

↓

$d \cdot (2^1 - 1) \cdot k = dk$

(16)

$$\prod_{j=1}^{d} P(X_j = x_j \mid Y = y)$$

↳ each of these is a table.

& we have to estimate prior probability $P(Y)$

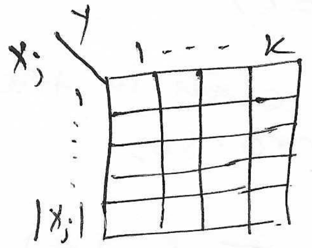
↳ a vector where each entry is just a count:

$$\hat{\pi}_y = \frac{\text{count}(Y=y)}{N}$$

↑ # of training samples with class $Y=y$, divided by total # of samples.

basically an MLE.

We want to estimate this table →



↑ Every column is also a categorical distribution.

we can do the same

↑ If I tell you $Y=y$, then $X_j = x_j$ is a categorical distribution since it sums to 1.

Shorthand notation: $\theta_a^{jc} = P(X_i = a \mid Y = c)$ ← for j-th feature, c-th class, of a particular value 'a'

$$\text{MLE } \theta_a^{jc} = \frac{\text{count}(X_i = a, Y = c)}{\text{count}(Y = c)}$$

Prediction in NB will be then,

$$\hat{y}_{MAP} = \max_y P(Y=y \mid X=x) = \arg\max_y \frac{\prod_{j=1}^{d} P(X_j = x_j \mid Y=y)\, P(Y=y)}{P(X=x)}$$ ↖ ignore this.

$$\Rightarrow \hat{y}_{MAP} = \arg\max_y \left[\log P(Y=y) + \sum_{j=1}^{d} \log P(X_j = x_j \mid Y=y)\right]$$

Laplacian smoothing

$$P(X_j = a \mid Y = c) \Rightarrow \frac{\text{Count}(X_j = a, Y = c)}{\text{Count}(Y = c)}$$

If a feature value never appears with a class in the training data, the numerator gets zero. So,

$$P(X_j = a \mid Y = c) = 0$$

Since NB multiplies probabilities, $P(x \mid y) = \prod_j P(x_j \mid y)$

One zero term kills the entire product, making $P(y \mid x) = 0$

Even if all other evidence strongly supports the class.

This is called zero-frequency problem.

$\downarrow$ solution - Laplacian smoothing.

Pretend that we have seen every possible feature value at least once for every class. So instead of trusting raw counts completely add a small constant to every count.

$$P(X_j = a \mid Y = c) = \frac{\text{Count}(X_j = a, Y = c) + 1}{\text{Count}(Y = c) + |X_j|}$$

total no. of possible values X_j can have

numerator: add 1 to every count

denominator: add $|X_j|$ to keep probabilities summing to 1.