03/02/15

BAYES CLASSIFIERS + NAIVE BAYES

(1) Bayes Error

→ The best ANY ML algorithm can do.

Recall Expected Loss/Error

$\vec{X}, Y \sim P^*(X, Y)$ [Unknown] (Using P^* just to emphasize that this is reality. Not from model)

$$E[\text{Loss}] = E_{P^*(\vec{x}, Y)}[L(\underbrace{y}_{\text{reality}}, \underbrace{\hat{y}}_{\text{prediction}})]$$

$$= \int_{\vec{x}} \int_{y} L(y, \hat{y}) \, p^*(\vec{x}, y) \, d\vec{x} \, dy$$

Bayes Error = min possible value of this

Space of $\vec{X}$ Say $\mathbb{R}^d$ — Space of Y

→ Assume Classification so $\mathcal{Y} = \{1, 2, \ldots, k\}$

[Replace $\int_{\mathcal{Y}}$ with $\sum_{y=1}^{k}$]

→ Assume 0-1 Loss ie $L(y, \hat{y}) = \begin{cases} 1 & y \neq \hat{y} \\ 0 & \text{else} \end{cases}$

$$\Rightarrow E[\text{Loss}] = \int_{\vec{x}} \sum_{y=1}^{k} L(y, \hat{y}) \, p^*(\vec{x}, y) \, d\vec{x}$$

[Mix of PMF & PDF]

$$E[\text{Loss}] = \int_{\vec{x}} \underbrace{\left[\sum_{y=1}^{k} L(y,\hat{y})\, p^*(y|\vec{x})\right] \underbrace{p(\vec{x})\, d\vec{x}}}_{\text{Looks like } E_{p^*(x)}[f(x)]}$$

$$= E_{p^*(\vec{x})}\left[\sum_{y=1}^{k} L(y,\hat{y})\, p^*(y|\vec{x})\right]$$

And what happens when I predict $\hat{y}$

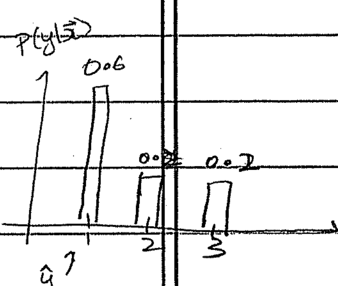
For this $\vec{x}$ how likely is the output to be y (in reality)

$$= E_{p^*(\vec{x})}\left[\underbrace{L(1,\hat{y})}_{\text{either 0 or 1}} p^*(1|\vec{x}) + L(2,\hat{y})\, p^*(2|\vec{x}) + \dots + L(k,\hat{y})\, p^*(k|\vec{x})\right]$$

$$= E_{p^*(\vec{x})}\left[1 - p^*(Y=\hat{y}|\vec{x})\right]$$

So $E[\text{Loss}] = E_{p^*(\vec{x})}\left[1 - p^*(Y=\hat{y}|\vec{x})\right]$

So $\underbrace{\min_{\hat{y}} E[\text{Loss}]}_{\text{Bayes Error}} = \underbrace{\max_{\hat{y}} p^*(Y=y|\vec{x})}_{\text{MAP [Maximum-a-posteriori]}}$



Bayes Error
[Error of the best possible predictor]

MAP [Maximum-a-posteriori]
aka "pick the most confident class"

$\Rightarrow$ So $\hat{y}_{MAP}$ is Bayes Optimal!

→ Bayes Classifier : the classifier that achieves Bayes Error.

[Similarly Bayes Regressor]

→ Note: $\hat{y}_{MAP}$ is Bayes Optimal FOR 0/1-Loss.

Different Loss ⇒ Different Bayes Classifiers

→ Great theory! How do I implement this?

Problem: Don't know $P^*(Y \mid \vec{X}=\vec{x})$

Solution: → Generative Approach

estimate
$P(\vec{x} \mid y)$
$P(y)$
from dat ⇒ Bayes Rule ⇒ $P(y \mid \vec{x})$ ⇒ $\hat{y}_{MAP}$

→ Discriminative Approach:

estimate
$P(y \mid \vec{x})$ from data ⇒ predict & done

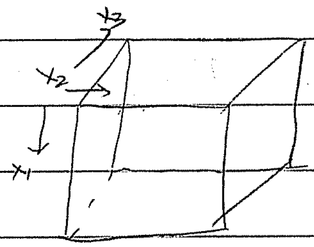
② Bayes Classifiers are hard to learn: A counting argument

Say $\vec{X} = \vec{x} \in \{0,1\}^d$ d-binary-features

$Y = y \in \{1, 2, \ldots, k\}$ k classes

Need to estimate: $P(Y=y)$ & $P(\vec{X}=\vec{x} \mid Y=y)$

How many parameters? $(k-1)$ + $(2^d - 1) \cdot k$



If $d=100$ $2^{100} \approx 10^{30}$ >>> data available

Lots of parameters ⇒ high variance & overfitting

Very sparse table

③ Naive Bayes to the Rescue.

Assume $\underbrace{P(X_1=x_1, \ldots, X_d=x_d \mid Y=y)} = \prod_{j=1}^{d} P(X_j = x_j \mid Y=y) \quad \forall x, y$

"Features are conditionally indep given class"

If I tell you $X_2=x_2$ you don't find out anything about $X_1=x_1$ any more than $Y=y$ already told you

Note: this does not mean $P(X_1 \mid X_2) = P(X_1)$

How many parameters in NB?

$\underbrace{P(X_i = x_i \mid Y=c)}_{d \cdot (2^1 - 1) \cdot k}$ $\underbrace{P(Y=y)}_{(k-1)}$

Decision Rule:

$$\hat{y}_{MAP} = \underset{y}{\arg\max}\ P(Y=y \mid \vec{X}=\vec{x})$$

$$= \underset{y}{\arg\max} \left[\frac{P(\vec{X}=\vec{x} \mid Y=y)\, P(Y=y)}{P(\vec{X}=\vec{x})}\right]$$

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

(sometimes also written as)

$$= \underset{y}{\arg\max} \left[\sum_{j=1}^{d} \log P(X_j=x_j \mid Y=y) + \log P(Y=y)\right]$$

(4) MLE/MAP for estimating NB parameters

→ Given dataset : $D=\{(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)\}$

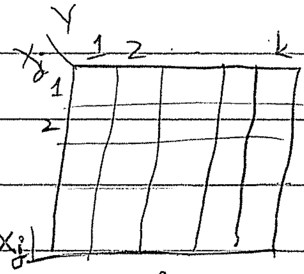
→ What are the parameters?

→ $P(Y)$ ← Categorical

$$\begin{array}{c} 1 \\ \vdots \\ \\ \vdots \\ k \end{array}\begin{bmatrix} \vdots \\ P(Y=c) \\ \vdots \end{bmatrix} \leftarrow \theta_c \text{ (shorthand)}$$

→ $\underbrace{P(X_j=a \mid Y=c)}$

$= \underbrace{\theta_a^{jc}}_{\text{shorthand}}$



$$\sum_{a=1}^{|X_j|} \theta_a^{jc} = 1$$

↑ Each column is categorical

All parameters $\vec{\theta} = \{\pi_1 \ldots, \pi_k, \theta_a^{jc} \ \forall j, c, a\}$

We know how to estimate categorical parameters
HW1 Q2.

$\hat{\theta}_{MLE}$

$$\hat{\pi}_c = P(Y=c) = \frac{\text{Count}(Y=c)}{n}$$

add pseudo-counts for $\hat{\theta}_{MAP}$

$$\hat{\theta}_a^{jc} = P(X_j = a \mid Y=c) = \frac{\text{Count}(X_j = a, Y=c)}{\text{Count}(Y=c)}$$

(5) Bag-of-Words model

NB assumes $X_i \perp X_j \mid Y$
ie $P(X_i \mid X_j, Y) = P(X_i \mid Y)$

BoW additionally assumes $P(X_i \mid Y) = P(X_j \mid Y)$
that all dimensions are essentially the same. More sharing. Less Parameters. Less expressive model class. Less variance. Less overfitting.