Lec 8

Tuesday, September 24, 2019 11:10

Recap: Naive Bayes

Naive assumption: individual features are independent of one another given label value

or: just the conditional densities factor:

$$\text{"}\mathbb{P}(X=x \mid Y=j)\text{"} = f_j(x) = \prod_{k=1}^{p} f_{jk}(x_k)$$

f_j: conditional density of $X \mid Y=j$

f_{jk}: conditional density of $X_k \mid Y=j$

Then:

$$\mathbb{P}(Y=j \mid X) \overset{\text{bayes law}}{=} \frac{\pi_j f_j(x)}{\sum_k \pi_k f_k(x)} \overset{\text{naive assumption}}{=} \frac{\pi_j \cdot \prod_{k=1}^{p} f_{jk}(x_k)}{\sum_k \pi_k \cdot \prod_{l=1}^{p} f_{kl}(x_l)}$$

π_j – fraction of label j in pop

Naive Bayes classifier:

- estimate each π_j, f_{jk}
- plug in & max over j

Bag of words features

BoW is an approach to featurizing text data.

& make it into quantitative features

Given texts $S_1 \dots S_n$

Given texts $s_1, \ldots, s_n$

e.g. s_1 = "My dog likes your dog"

We treat each as a bag of its words ignore order but (potentially) remember counts

$BoW(s_1) = \{$"your", "likes", "My", "dog"$\times 2\}$

Then we make a dictionary of all words

$D = \bigcup_{i=1}^{n} BoW(s_i)$ ← set of all words that appear in any text in dataset

and feature each text by

X_{ij} = # times that word j appears in text i $\quad \forall j \in D$

(alt ver: 0/1 whether appears)

Intent: X_i capture the overall nature of the text

Important considerations: (you will explore in HW2)

- decapitalization "Dog" → "dog"
- lemmatization "likes" → "like"
- prune contentless words, e.g. "a", "the", etc
- normalize or no normalization
- Bag-of-ngrams $n \geq 2$

Naïve Bayes w/ BoW

BoW generates very high-dim features

→ perfect for Naive assumption, which will help deal w/ high dims

Let's focus on $X_{ij} = 0$ or 1 whether S_i contains word j

Then $\hat{\mathbb{P}}(X_j = 1 \mid Y = k) = \hat{p}_{jk}$

$$= \frac{\sum_{i=1}^{n} \mathbb{I}[Y_i = k, X_{ij} = 1] + \alpha}{\sum_{i=1}^{n} \mathbb{I}[Y_i = k] + \alpha}$$

= α-smoothed frac of k-texts that have word j

Consider binary case (e.g. $Y \in \{$spam, not spam$\}$)

$$\text{logit}\left(\hat{\mathbb{P}}(Y=1 \mid X=x)\right) = \text{logit}(\hat{\pi}_1) + \sum_{j=1}^{p} \left(x_j \log\left(\frac{\hat{p}_{j1}}{\hat{p}_{j0}}\right) + (1-x_j) \log\left(\frac{1-\hat{p}_{j1}}{1-\hat{p}_{j0}}\right)\right)$$

Density estimation

Consider data $Y_1, \dots, Y_n \in \mathbb{R}$ drawn from a distribution w/ CDF (cumulative dist fn) F $(F(y) = \mathbb{P}(Y \le y))$ & PDF (prob density fn) f $(f(y) = F'(y))$

We want to understand this unknown dist from the data.

Estimating the CDF F is easy

empirical CDF: $\hat{F}_n(y) = \frac{1}{n}\sum_i \mathbb{I}[Y_i \le y]$

What about f?

In particular if $y \notin \{Y_1, \dots, Y_n\}$ (haven't seen y) then it's particularly difficult to say how likely y is

Histogram density estimation

Choose cutoffs $y_1 < y_2 < \dots < y_{m+1}$

Get bins $[y_1, y_2), [y_2, y_3), \dots, [y_m, y_{m+1})$

(we need to make sure that $y_1 \le Y_i$ $\forall i$ and $y_{m+1} > Y_i$ $\forall i$)

Count the data in each bin

$$n_j = \sum_{i=1}^n \mathbb{I}[Y_i \in [y_{j-1}, y_j)] = \#\text{ data pts in } j\text{th bin}$$

A histogram is just a bar chart w/ these counts

And a histogram density estimate

$$\hat{f}_n^{\text{hist}}(y) = \frac{n_j}{n} \frac{1}{y_j - y_{j-1}}$$ where j is s.t. $y \in [y_{j-1}, y_j)$

But histograms are not smooth — doesn't really look like f

Kernel Density Estimate (KDE)

$$\hat{f}_n^{KDE}(y) = \frac{1}{n} \sum_{i=1}^{n} K_\lambda(y - y_i)$$

Where $K_\lambda(u) = \frac{1}{\lambda} \varphi(u/\lambda)$

φ: std normal pdf

$\varphi(u) = \frac{1}{(2\pi)^{d/2}} e^{-\frac{1}{2}\|u\|^2}$