Cornell Tech

CS 5785: Lecture 10

Natalie Abrams, Matthew Dalton, Brian Pfaff

9/27/2018

1 Recap: Kernel Density Estimation

Kernel Regression

$$f_n(x) = \frac{\sum_{i=1}^{n} k_\lambda(x_i - x) y_i}{\sum_{i=1}^{n} k_\lambda(x_i - x)} \approx E(Y|X = x)$$

Kernel classification:

$$\hat{P}(Y = j|X = x) = \frac{\hat{\pi}_j \hat{f}_j(x)}{\sum_{k=1}^{n} \hat{\pi}_k \hat{f}_k(x)}$$

A problem we encounter is the Curse of Dimensionality!

As we increase the dimension. the amount of data within an area decreases. And example is taking half of each dimension as we increase it. See the below matrix for how much of the area we cover as we increase the dimension

$$\begin{pmatrix} \text{dimension} & \text{area} \\ 1 & 0.5 \\ 2 & 0.25 \\ 3 & 0.125 \\ 4 & 0.0625 \end{pmatrix}$$

2 Naive Bayes Assumption and Classifier

From above:

$$P(Y = j|X = x) = \frac{\pi_j f_j(x)}{\sum_{k=1}^{n} \pi_k f_k(x)}$$

where:

π_j = prevalence of class j, which is easy to estimate regardless of p

$f_j(x)$ = density of (x | y=j), which is hard to estimate for large p (especially with KDE)
Instead, we can approximate using the Naive Bayes assumption:

$$f_j(x) = f_j(x_1, x_2, ..., x_p) = \prod_{k=1}^{1-p} f_{jk}(x_k)$$

$$Example: \prod_{k=1}^{4} f_j(k) = \pi_1 * \pi_2 * \pi_3 * \pi_4$$

Meaning: we assume that x_1,...,x_p are statistically independent given y
Simpler way of estimating $f_j(x)$:

$$P(Y = j|X = x) = \frac{\pi_j f_j(x)}{\sum_{k=1}^{n} \pi_k f_k(x)}$$

$$\frac{\pi_j \prod_{l=1}^{1-p} f_{j,l}(x_l)}{\sum_{k=1}^{n} \pi_k \prod_{l=1}^{1-p} f_{k,l}(x_l)}$$

$$\frac{P(Y = j|X = x)}{P(Y = k|X = x)} = \frac{\pi_j \prod_{l=1}^{1-p} f_{j,l}(x_l)}{\pi_k \prod_{l=1}^{1-p} f_{k,l}(x_l)}$$

Now take logs:

$$\log(\frac{P(Y = j|X = x)}{P(Y = k|X = x)} = \log(\frac{\pi_j \prod_{l=1}^{1-p} f_{j,l}(x_l)}{\pi_k \prod_{l=1}^{1-p} f_{k,l}(x_l)})$$

$$\log(\frac{P(Y = j|X = x)}{P(Y = k|X = x)} = \log(\frac{\pi_j}{\pi_k}) + \sum_{l=1}^{p} \log(\frac{f_{j,l}(x_l)}{f_{k,l}(x_l)})$$

if $Y \in (0, 1)$ the above is: logit(P(Y=1 | X=x))

2.1 Naive Bayes Classifier:

We estimate each of the π_j, $f_{j,l}(x_l)$ and plug in above. For the class frequencies, use the empirical frequency:

$$\hat{\pi}_j = \frac{\sum_{i=1}^{n} \mathbb{1}[Y_i = j]}{n}$$

For discrete features use empirical frequencies as well:

$$\hat{f}_{jl}(x_l) = \frac{\sum_{i=1}^{n} \mathbb{1}[X_{il} = x_l, Y_i = j]}{\sum_{i=1}^{n} \mathbb{1}[Y_i = j]}$$

What if $\hat{f}_{jl}(x_l) = 0$, $\hat{f}_{kl}(x_l) \neq 0$?
If there is a 0, then the probabilities will move towards a 0. This can be a problem as we do not want the fact that we are estimating things to have such a large influence on our results (Specially, with sparse data). The log odds will go to $\pm\infty$.

One solution is to use Laplace smoothing (pad the data)

$$\hat{f}_{jl}(x_l) = \frac{\sum_{i=1}^{n} \mathbb{1}[X_{il} = x_l, Y_i = j] + \alpha}{\sum_{i=1}^{n} \mathbb{1}[Y_i = j] + \alpha}$$

As n increases, α gets washed out. So as the number of samples increase, alpha will be less influential

ie: $\frac{0.1}{1} = 0.1$ but $\frac{0.1}{1000} = 0.0001$