Lecture 6 Feb 25, 2021

Why decision tree?

1. Fast to predict
2. You can throw away data after training unlike Knn
3. Interpretable (based on splitting)
4. Easy to select features (based on information gained)

Why not decision tree

1. Not great on correlated data
2. Very prone to overfitting. (Random tree forests - last lecture)

Class Conditionals (Generative Model)

As you learn the model, you learn the data distribution

We want to learn: $p(x) = \underbrace{p(x, c_1)}_{\text{way to classify points}} + p(x, c_2)$

(Assuming binary classes)

$$p(x, c_i) = \underbrace{p(c_i)}_{\text{prior}} \underbrace{p(x|c_i)}_{\text{likelihood}}$$

Once we know prior / likelihood, we can compute the class posteriori:

$$p(c_i|x) = \frac{p(c_i)\, p(x|c_i)}{p(x)}$$

So for the binary class case,

$$p(C_1|x) = \frac{p(C_1)\,p(x|C_1)}{p(x)} \qquad p(C_2|x) = 1 - p(C_1|x)$$

Thus the decision function is

$$a(x) = \ln\left(\frac{p(C_1|x)}{p(C_2|x)}\right)$$

$$= \ln\left(p(C_1)\,p(x|C_1)\right) - \ln\left(p(C_2)\,p(x|C_2)\right)$$

To summarize we want to find:

1) The priors : $p(C_i)$
2) The likelihoods : $p(x|C_i)$

Useful to form <u>decision boundary</u>, we can also generate new data.

To generate new data:

1) Sample class $\hat{C}$ from prior $p(C)$
2) Sample data $\hat{x}$ from likelihood $p(x|\hat{C})$

How do we learn priors and likelihoods?

Given sample $\{(x_i, y_i)\}_{i=1}^{N}$

1) priors : $p(C_1) = \frac{\#\text{ class 1}}{N}$

$$p(C_2) = \frac{\#\text{ class 2}}{N}$$

2) likelihoods : - partition data based on class

- use all x_i s.t. $y_i = c_j$ to learn $p(x|c_j)$, $j \in \{1, 2\}$.
- use mle to learn.

$$p(c_i|x) = \frac{p(x|c_i)p(c_i)}{p(x)}$$

$$= \frac{p(x|c_i)p(c_i)}{p(x|c_1)p(c_1) + p(x|c_2)p(c_2)}$$

Gaussian Class Conditionals.

We assume likelihoods to be Gaussian

$$G(\vec{x}; \vec{\mu}_i, \Sigma_i) = \frac{1}{\sqrt{2\pi|\Sigma|^d}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu}_i)^T \Sigma^{-1} (\vec{x}-\vec{\mu}_i)\right)$$

To learn params, we use MLE.

The decision boundary, assuming $p(c_1) = p(c_2) = \frac{1}{2}$

$$a(x) = \ln\left(\frac{p(x|c_1)\cancel{p(c_1)}}{p(x|c_2)\cancel{p(c_2)}}\right)$$

$$= \ln p(x|c_1) - \ln p(x|c_2)$$

$$= \ln\left(\frac{1}{\sqrt{2\pi|\Sigma_1|^d}}\right) - \ln\left(\frac{1}{\sqrt{2\pi|\Sigma_2|^d}}\right)$$

$$-\frac{1}{2}(\vec{x}-\vec{\mu}_1)^T \Sigma^{-1} (\vec{x}-\vec{\mu}_1) + \frac{1}{2}(\vec{x}-\vec{\mu}_2)^T \Sigma^{-1} (\vec{x}-\vec{\mu}_2)$$

We see this is a quadratic function of $\vec{x}$.

Why use this?

1) You know data is generated normally

2) You need more expressive model

Problems of this:

1) Only works well with Gaussian
2) Parameters scale quadratically and calculations can get expensive.