IFT6390
Fondements de l'apprentissage machine

Bayes Classifier
Naive Bayes Classifier

Professor: Ioannis Mitliagkas
Slides: Pascal Vincent

Probabilistic approach to learning

- We assumed that the data is generated by an unknown process.
- X,Y is seen as a pair of random variables, distributed according to an unknown probability law P (X,Y).
- X (a vector variable) is itself seen as a set of scalar random variables.

$$P(X,Y) = P(X_{[1]}, \ldots, X_{[d]}, Y)$$

Binary classifier (reminder)

- For binary classification (2 classes), we often use a classifier computing a real valued output $g(x)$.
g is called a **discriminant function**.

- The **decision/classification function** $f(x)$ is then obtained by comparing the output $g(x)$ with a *threshold* t. I.e. we predict the first class if $g(x)$ is smaller than t and the second otherwise.

- One common choice for the *threshold* t is 0, which boils down to looking at the sign of $g(x)$.

- Another common choise is a *threshold* of 0.5 when the *output* $g(x)$ is in [0, 1] and can be interpreted as the probability of one of the class given the input x. (observe that the probability of the other class is $1-g(x)$; the 0.5 threshold corresponds to predicting the most likely class...)

Multiclass classification

- When there are m classes, the output $g(x)$ is an m-dimensional vector containing "**scores**" for each class.

discriminant functions:

$$g(\mathbf{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_m(\mathbf{x}))$$

- The decision/classification $f(x)$ is then the class with the highest score:

prediction = arg max(*output*) $\qquad$ $f(x)$ = arg max($g(x)$)

- When the outputs (scores) are positive and **sum to 1**, we can inperpret the j-th output as the probability of the j-th class given the input.

- This works also for binary classification (2 classes: 2 outputs)!

- Note: with two classes, predicting the class with the highest score is equivalent to predicting the *sign* of the difference between the two scores.

Probability

(what you should be familiar with)

- Discrete and continuous random variables
- Joint probability distribution
- Marginal distribution, marginalization
- Conditionnal probability
- Bayes Rule
- Independence

Bayes Rule

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Thomas Bayes
1702 - 1761

Bayes Classifier

or how to build a classifier from density estimators

- Split the training data into m subsets corresponding to the m each one containing the points from only one class.
- Estimate the density/distribution of points for each class $c \in \{1, .., m\}$

$$\hat{p}_c(x) \simeq P(X = x|Y = c)$$

Computing this part is difficult as the dimension grows

- Estimate *prior probabilities* for each class: (for ex. by counting the proportion of each class in the training data)

$$\hat{P}_c = \frac{n_c}{n} \simeq P(Y = c)$$

- Apply Bayes rule to compute the *posterior probability* of each class given x.

$$\underbrace{P(Y = c|X = x)}_{\textit{posterior}} = \frac{\underbrace{P(X = x|Y = c)}_{\textit{class-conditional density (likelihood)}}\underbrace{P(Y = c)}_{\textit{prior}}}{P(X = x)}$$

$$= \frac{P(X = x|Y = c)P(Y = c)}{\sum_{c'=1}^{m} P(X = x|Y = c')P(Y = c')}$$

$$\simeq \frac{\hat{p}_c(x)\hat{P}_c}{\sum_{c'=1}^{m} \hat{p}_{c'}(x)\hat{P}_{c'}}$$

- Predict the most likely class.

Bayes classifier is diiferent from Naive Bayes. In naive bayes, we assume independence

Naive Bayes

- For naive Bayes, we suppose that, for each class $c \in \{1, .., m\}$, the components of X are independent given c:

This assumption removes the dependency of the dimension, hence computationally easy however, the power is reduced

$$
\begin{aligned}
P(X|Y=c) &= P(X_{[1]}|Y=c)P(X_{[2]}|Y=c)...P(X_{[d]}|Y=c) \\
\hat{p}_c(x) &= \hat{p}_{c,1}(x_{[1]})\hat{p}_{c,2}(x_{[2]})...\hat{p}_{c,d}(x_{[d]})
\end{aligned}
$$

Works well with limited size

- So we just have to estimate univariate densities $\hat{p}_{c,j}(x_{[j]})$ which is an easy task (univariate <=> dimension 1: no curse of dimensionality; histogram methods or Parzen often perform well).

- We then build a Bayes classifier using the estimate $\hat{p}_c(x)$ for each class.

Warning

- "Bayes Classifier" $\neq$ "**Naive** Bayes"
- "Naive Bayes" $\subset$ "Bayes Classifier"
- Naive Bayes is a very naive model, with very low capacity
  (it cannot capture/model interactions between the input components!)
- The Bayes classifier can have a very high capacity
  (depends on the density estimators used to model the class conditional densities)

Other way to build a classifier: Use a good estimate of the joint distribution

- Estimate the joint distribution P(X,Y)
- We can then compute the conditional probabilities for each class *c*:

$$
\begin{aligned}
P(Y = c | X = x) &= \frac{P(X = x, Y = c)}{P(X = x)} \\
&= \frac{P(X = x, Y = c)}{\sum_{c'=1}^{m} P(X = x, Y = c')}
\end{aligned}
$$

- Note that these class probabilities (given x) are proportional to the joint probabilities. The denominator is a simple normalizer (so that they sum to one).