Naive Bayes Classifiers

Dr. Rahul Kottath

Background

- There are three methods to establish a classifier
  a) Model a classification rule directly
     - Examples: k-NN, decision trees, perceptron, SVM
  b) Model the probability of class memberships given input data
     - Example: multi-layered perceptron with the cross-entropy cost
  c) Make a probabilistic model of data within each class
     - Examples: naive Bayes, model based classifiers
- *a*) and *b*) are examples of discriminative classification
- *c*) is an example of generative classification
- *b*) and *c*) are both examples of probabilistic classification

Things We'd Like to Do

- Spam Classification
  - Given an email, predict whether it is spam or not

- Medical Diagnosis
  - Given a list of symptoms, predict whether a patient has disease X or not

- Weather
  - Based on temperature, humidity, etc… predict if it will rain tomorrow

Bayesian Classification

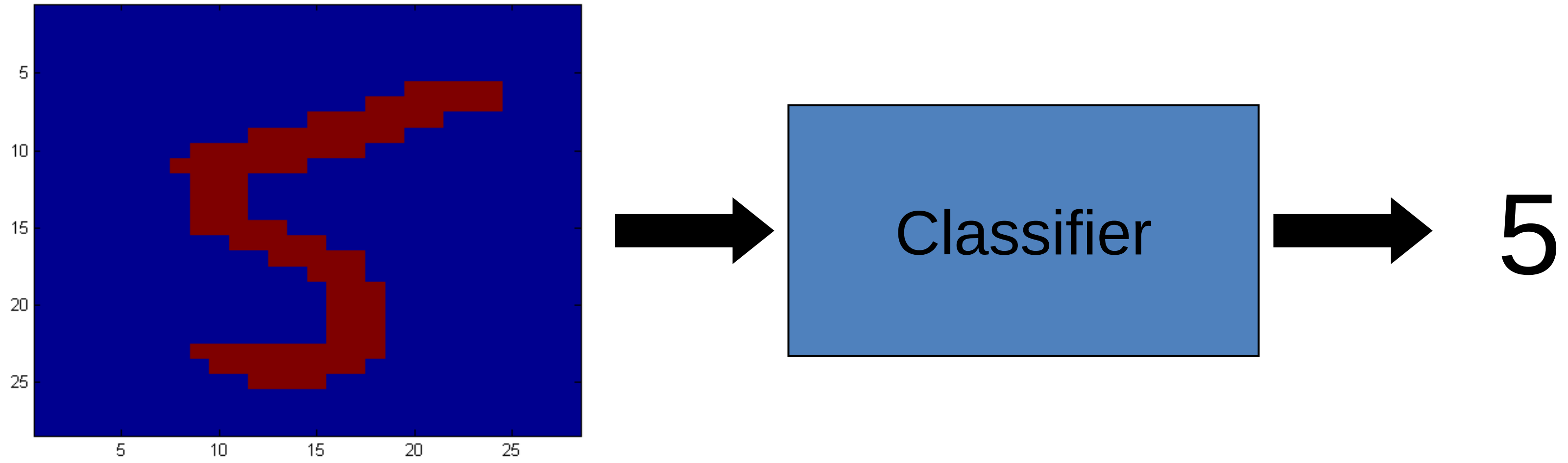
$x = (x_1, x_2, x_3, x_4 \ldots x_n) - y$

- Problem statement:
  - Given features $X_1,X_2,\ldots,X_n$
  - Predict a label Y

Another Application

40×40

Digit Recognition



$X_1,\ldots,X_n \in \{0,1\}$ (Black vs. White pixels)

$Y \in \{5,6\}$ (predict whether a digit is a 5 or a 6)

The Bayes Classifier

- A good strategy is to predict:

$$\arg\max_{Y} P(Y|\underbrace{X_1, \ldots, X_n}_{\text{pixels}})$$

  - (for example: what is the probability that the image represents a 5 given its pixels?)

- So … How do we compute that?

The Bayes Classifier

$P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)}$

- Use Bayes Rule!

Likelihood   Prior

$$P(Y|\underbrace{X_1, \ldots, X_n}_{B}) = \frac{P(X_1, \ldots, X_n|Y)P(Y)}{P(X_1, \ldots, X_n)}$$

Normalization Constant

- Why did this help? Well, we think that we might be able to specify how features are "generated" by the class label

The Bayes Classifier

$$\frac{p}{} + \frac{1-p}{} = 1$$

- Let's expand this for our digit recognition task:

$$\frac{a}{a+b} + \frac{b}{a+b}$$

$$= \frac{a+b}{a+b} = 1$$

$$P(\underline{Y=5}|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y=5)P(Y=5)}{P(X_1,\ldots,X_n|Y=5)P(Y=5)+P(X_1,\ldots,X_n|Y=6)P(Y=6)}$$

$$P(\underline{Y=6}|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y=6)P(Y=6)}{P(X_1,\ldots,X_n|Y=5)P(Y=5)+P(X_1,\ldots,X_n|Y=6)P(Y=6)}$$

- To classify, we'll simply compute these two probabilities and predict based on which one is greater

Probability Basics

- Prior, conditional and joint probability
  - Prior probability: $P(X)$
  - Conditional probability: $P(X_1 | X_2), P(X_2 | X_1)$
  - Joint probability: $\mathbf{X} = (X_1, X_2), P(\mathbf{X}) = P(X_1, X_2)$
  - Relationship: $P(X_1, X_2) = P(X_2 | X_1)P(X_1) = P(X_1 | X_2)P(X_2)$
  - Independence: $P(X_2 | X_1) = P(X_2), P(X_1 | X_2) = P(X_1), P(X_1, X_2) = P(X_1)P(X_2)$
- Bayesian Rule

$$P(C | \mathbf{X}) = \frac{P(\mathbf{X} | C)P(C)}{P(\mathbf{X})} \qquad Posterior = \frac{Likelihood \times Prior}{Evidence}$$

Naïve Bayes

$P(x_1/x_2,x_3,\dots,x_n,C) \cdot P(x_2,x_3\dots x_n/C) \cdots P(x_n/C) \cdot P(C)$

$P(x_1/C) \cdot P(x_2/C) \cdot P(x_3/C) \cdots$

- Bayes classification

$$P(C\,|\,\mathbf{X}) \propto P(\mathbf{X}\,|\,C)P(C) = P(X_1,\cdots,X_n\,|\,C)P(C)$$

Difficulty: learning the joint probability

$P(\text{Yes}/X) = P(x_1/C=\text{Yes}) \cdot P(x_2/C=\text{Yes}) \cdots P(x_n/C=\text{Yes}) \cdot P(C=\text{Yes})$

- Naïve Bayes classification
  - Making the assumption that all input attributes are independent

$$\begin{aligned} P(X_1,X_2,\cdots,X_n\,|\,C) &= P(X_1\,|\,X_2,\cdots,X_n;C)P(X_2,\cdots,X_n\,|\,C) \\ &= P(X_1\,|\,C)P(X_2,\cdots,X_n\,|\,C) \\ &= P(X_1\,|\,C)P(X_2\,|\,C)\cdots P(X_n\,|\,C) \end{aligned}$$

$P(C/X) = \prod_{i=1}^{n} P(x_i/C) \cdot P(C)$

Example

$P(Yes) = 9/14$

$P(No) = 5/14$

x_1 x_2 x_3 x_4

Example: Play Tennis

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

4 YES
0 NO

9 YES
5 NO

D15

Example

2 YES   3 (NO)

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Example

Humidity	Play=*Yes*	Play=*No*
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

$P(\text{Play}=Yes) = 9/14$

$P(\text{Play}=No) = 5/14$

Example

$P(Yes/x') = P(outlook = sunny/yes) \cdot P(Temp = Cool/yes) \cdot P(Hum = High/yes) \cdot P(wind = Strong/yes) \cdot P(yes)$

$P(No/x') =$

- Test Phase
  - Given a new instance,

    x′=(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)

play = ?

  - Look up tables

$$\underbrace{\begin{array}{l} P(\text{Outlook}=\textit{Sunny}\,|\,\text{Play}=\textit{Yes}) = 2/9 \\ P(\text{Temperature}=\textit{Cool}\,|\,\text{Play}=\textit{Yes}) = 3/9 \\ P(\text{Huminity}=\textit{High}\,|\,\text{Play}=\textit{Yes}) = 3/9 \\ P(\text{Wind}=\textit{Strong}\,|\,\text{Play}=\textit{Yes}) = 3/9 \\ P(\text{Play}=\textit{Yes}) = 9/14 \end{array}}_{Yes}
\qquad
\underbrace{\begin{array}{l} P(\text{Outlook}=\textit{Sunny}\,|\,\text{Play}=\textit{No}) = 3/5 \\ P(\text{Temperature}=\textit{Cool}\,|\,\text{Play}==\textit{No}) = 1/5 \\ P(\text{Huminity}=\textit{High}\,|\,\text{Play}=\textit{No}) = 4/5 \\ P(\text{Wind}=\textit{Strong}\,|\,\text{Play}=\textit{No}) = 3/5 \\ P(\text{Play}=\textit{No}) = 5/14 \end{array}}_{No}$$

$P(Yes/x')$

Example

MAP rule

$P(Yes|\mathbf{x}')$: $[P(Sunny|\text{Yes})P(Cool|Yes)P(High|\text{Yes})P(Strong|Yes)]P(\text{Play}=Yes) = \underline{0.0053}$

$P(No|\mathbf{x}')$: $[P(Sunny|\text{No})\ P(Cool|\text{No})P(High|No)P(Strong|No)]P(\text{Play}=No) = \underline{0.0206}$

Given the fact $P(Yes|\mathbf{x}') < P(No|\mathbf{x}')$, we label $\mathbf{x}'$ to be "*No*".

$$P(Yes/x') = \frac{0.0053}{0.0053 + 0.0206}$$

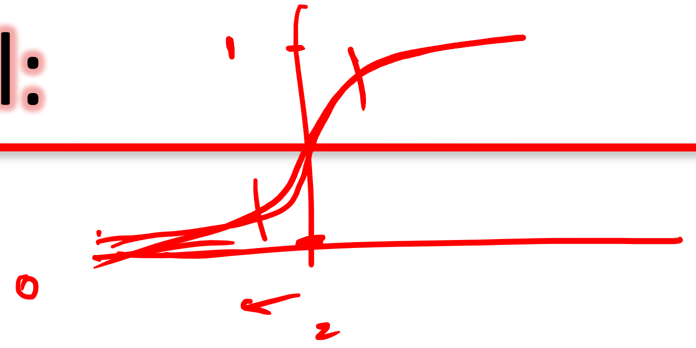
$$P(No/x') = \frac{0.0206}{0.0053 + 0.0206}$$

$$P(Yes/x') + P(No/x') = 1$$

Conclusions

- Naïve Bayes based on the independence assumption
  - Training is very easy and fast; just requiring considering each attribute in each class separately
  - Test is straightforward; just looking up tables or calculating conditional probabilities with normal distributions
- A popular generative model
  - Performance competitive to most of state-of-the-art classifiers even in presence of violating independence assumption
  - Many successful applications, e.g., spam mail filtering
  - A good candidate of a base learner in ensemble learning
  - Apart from classification, naïve Bayes can do more…

Evaluating a Classification model:



1. Log Loss or Cross-Entropy Loss:

- It is used for evaluating the performance of a classifier, whose output is a probability value between the 0 and 1.
- For a good binary Classification model, the value of log loss should be near to 0.
- The value of log loss increases if the predicted value deviates from the actual value.
- The lower log loss represents the higher accuracy of the model.
- For Binary classification, cross-entropy can be calculated as:

actual	predict	loss
0	0	0
0	1	
1	0	
1	1	0

$\log 1 = 0$

$$\text{Loss} = -(y\log(p)+(1-y)\log(1-p))$$

Actual

Evaluating a Classification model:

Predicted

2. Confusion Matrix:

- The confusion matrix provides us a matrix/table as output and describes the performance of the model.
- It is also known as the error matrix.
- The matrix consists of predictions result in a summarized form, which has a total number of correct predictions and incorrect predictions. The matrix looks like as below table:

	Actual Positive	**Actual Negative**
Predicted Positive	True Positive	False Positive
Predicted Negative	False Negative	True Negative

$$\text{Accuracy} = \frac{TP+TN}{\text{Total Population}}$$

Example

example confusion matrix for a binary classifier

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

$\frac{150}{165}$

Example

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Example

- Accuracy: Overall, how often is the classifier correct?
  - (TP+TN)/total = (100+50)/165 = 0.91
- Misclassification Rate: Overall, how often is it wrong?
  - (FP+FN)/total = (10+5)/165 = 0.09
  - equivalent to 1 minus Accuracy
  - also known as "Error Rate"
- True Positive Rate: When it's actually yes, how often does it predict yes?
  - TP/actual yes = 100/105 = 0.95
  - also known as "Sensitivity" or "Recall"

Example

→ Can Accuracy good measure ?

NO

- False Positive Rate: When it's actually no, how often does it predict yes?
  - FP/actual no = 10/60 = 0.17
- True Negative Rate: When it's actually no, how often does it predict no?
  - TN/actual no = 50/60 = 0.83
  - equivalent to 1 minus False Positive Rate
  - also known as "Specificity"
- Precision: When it predicts yes, how often is it correct?
  - TP/predicted yes = 100/110 = 0.91

F1 Score

Python Packages needed

- pandas
  - Data Analytics
- numpy
  - Numerical Computing
- matplotlib.pyplot
  - Plotting graphs
- sklearn
  - Classification and Regression Classes

Implementation Using sklearn

Let's go to Jupyter Notebook!