The naïve Bayes’ classifier

Probability Basics

- Prior, conditional and joint probability for random variables
  - 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

$$P(c | \mathbf{x}) = \frac{P(\mathbf{x} | c)P(c)}{P(\mathbf{x})}$$

$$Posterior = \frac{Likelihood \times Prior}{Evidence}$$

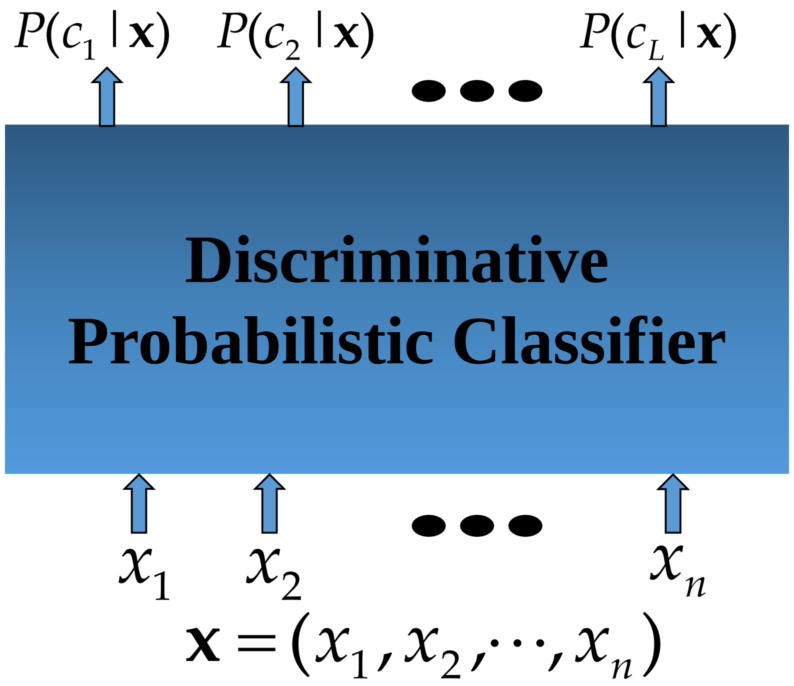
Discriminative

Generative

Probabilistic Classification Principle

- Establishing a probabilistic model for classification
  - **Discriminative model**

$$P(c \mid \mathbf{x}) \quad c = c_1, \cdots, c_L, \mathbf{x} = (x_1, \cdots, x_n)$$

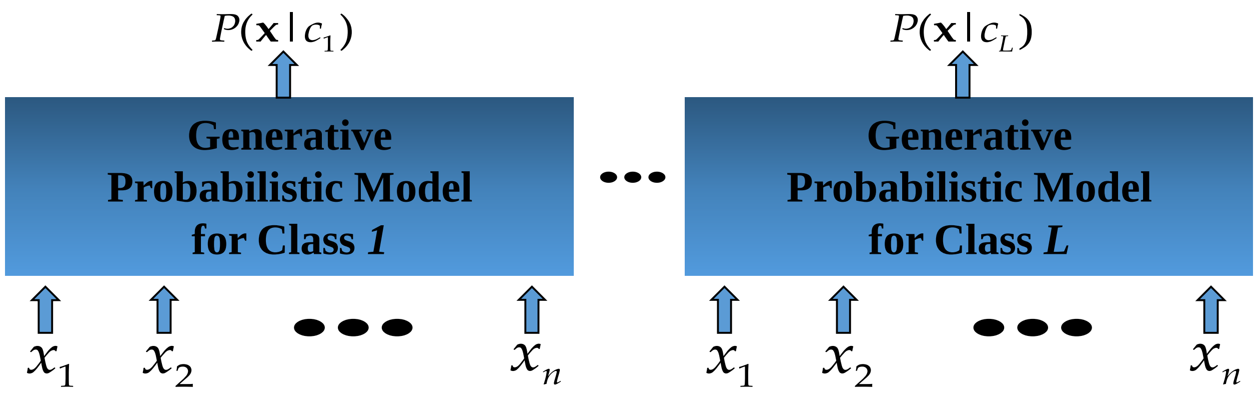


- To train a discriminative classifier (regardless its probabilistic or non-probabilistic nature), all training examples of different classes must be jointly used to build up a single discriminative classifier.
- Output L probabilities for L class labels in a probabilistic classifier while a single label is achieved by a non-probabilistic discriminative classifier .

Probabilistic Classification Principle

- Establishing a probabilistic model for classification (cont.)
  - **Generative model (must be probabilistic)**

$$P(\mathbf{x}|c) \quad c = c_1, \cdots, c_L, \mathbf{x} = (x_1, \cdots, x_n)$$



$$\mathbf{x} = (x_1, x_2, \cdots, x_n)$$

- *L* probabilistic models have to be trained independently
- Each is trained on only the examples of the same label
- Output *L* probabilities for a given input with *L* models
- "Generative" means that such a model can produce data subject to the distribution via sampling.

Probabilistic Classification Principle

- **M**aximum **A** **P**osterior (**MAP**) classification rule
  - For an input $\mathbf{x}$, find the largest one from L probabilities output by a discriminative probabilistic classifier $P(c_1 | \mathbf{x}), \ldots, P(c_L | \mathbf{x})$.
  - Assign x to label c^* if $P(c^* | \mathbf{x})$ is the largest.
- Generative classification with the MAP rule
  - Apply Bayesian rule to convert them into posterior probabilities

$$P(c_i | \mathbf{x}) = \frac{P(\mathbf{x} | c_i)P(c_i)}{P(\mathbf{x})} \propto P(\mathbf{x} | c_i)P(c_i)$$

$$\text{for } i = 1, 2, \cdots, L$$

Common factor for all L probabilities

  - Then apply the MAP rule to assign a label

Naïve Bayes

- Bayes classification

$$P(c\,|\,\mathbf{x}) \propto P(\mathbf{x}\,|\,c)P(c) = P(x_1,\cdots,x_n\,|\,c)P(c) \;\text{ for } c = c_1,\ldots,c_L.$$

Difficulty: learning the joint probability $P(x_1,\cdots,x_n\,|\,c)$ is often infeasible!

- Naïve Bayes classification
  - Assume all input features are class conditionally independent!

$$\begin{aligned} P(x_1,x_2,\cdots,x_n\,|\,c) &= \underline{P(x_1\,|\,x_2,\cdots,x_n,c)}P(x_2,\cdots,x_n\,|\,c) \\ &= \underline{P(x_1\,|\,c)}P(x_2,\cdots,x_n\,|\,c) \\ &= P(x_1\,|\,c)P(x_2\,|\,c)\cdots P(x_n\,|\,c) \end{aligned}$$

Applying the independence assumption

  - Apply the MAP classification rule: assign $\mathbf{x}' = (a_1,a_2,\cdots,a_n)$ to c^* if

$$\underbrace{[P(a_1\,|\,c^*)\cdots P(a_n\,|\,c^*)]}_{\text{estimate of } P(a_1,\cdots,a_n\,|\,c^*)}P(c^*) > \underbrace{[P(a_1\,|\,c)\cdots P(a_n\,|\,c)]}_{\text{esitmate of } P(a_1,\cdots,a_n\,|\,c)}P(c),\quad c \neq c^*, c = c_1,\cdots,c_L$$

Naïve Bayes

- Algorithm: Discrete-Valued Features
  - Learning Phase: Given a training set **S** of F features and L classes,

    For each target value of c_i $(c_i = c_1, \cdots, c_L)$

    $\hat{P}(c_i) \leftarrow$ estimate $P(c_i)$ with examples in S;

    For every feature value x_{jk} of each feature x_j $(j = 1, \cdots, F; k = 1, \cdots, N_j)$

    $\hat{P}(x_j = x_{jk} | c_i) \leftarrow$ estimate $P(x_{jk} | c_i)$ with examples in S;

    Output: $F * L$ conditional probabilistic (generative) models
  - Test Phase: Given an unknown instance $\mathbf{x}' = (a'_1, \cdots, a'_n)$

    "Look up tables" to assign the label c^* to $\mathbf{X}'$ if

$$[\hat{P}(a'_1 | c^*) \cdots \hat{P}(a'_n | c^*)]\hat{P}(c^*) > [\hat{P}(a'_1 | c_i) \cdots \hat{P}(a'_n | c_i)]\hat{P}(c_i), \quad c_i \neq c^*, c_i = c_1, \cdots, c_L$$

The Naïve Bayes Classifier example

Deadline?	Is there a party?	Lazy?	Activity
Urgent	Yes	Yes	Party
Urgent	No	Yes	Study
Near	Yes	Yes	Party
None	Yes	No	Party
None	No	Yes	Pub
None	Yes	No	Party
Near	No	No	Study
Near	No	Yes	TV
Near	Yes	Yes	Party
Urgent	No	No	Study

Suppose that you have deadlines looming, but none of them are particularly urgent, that there is no party on, and that you are currently lazy. Then the classifier needs to evaluate your activity for today

The Naïve Bayes Classifier example contd..

Step 1: Convert the data set into a frequency table

Frequency table

	Party		**study**		**TV**		**Pub**		**Lazy**	
	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No
Urgent	1	0	1	1	0	0	0	0	6	4
Near	2	0	0	1	1	0	0	0		
None	0	2	0	0	0	0	1	0		

Step 2: Create Likelihood table by finding the probabilities

Step 3: Use Naive Bayesian equation to calculate the posterior probability for each class.

The class with the highest posterior probability is the outcome of prediction

The Naïve Bayes Classifier example contd..

- P(Party) × P(Near | Party) × P(No Party | Party) × P(Lazy | Party)
- P(Study) × P(Near | Study) × P(No Party | Study) × P(Lazy | Study)
- P(Pub) × P(Near | Pub) × P(No Party | Pub) × P(Lazy | Pub)
- P(TV) × P(Near | TV) × P(No Party | TV) × P(Lazy | TV)

$$P(\text{Party}|\text{near (not urgent) deadline, no party, lazy}) = \frac{5}{10} \times \frac{2}{5} \times \frac{0}{5} \times \frac{3}{5} = 0$$

$$P(\text{Study}|\text{near (not urgent) deadline, no party, lazy}) = \frac{3}{10} \times \frac{1}{3} \times \frac{3}{3} \times \frac{1}{3} = \frac{1}{30}$$

$$P(\text{Pub}|\text{near (not urgent) deadline, no party, lazy}) = \frac{1}{10} \times \frac{0}{1} \times \frac{1}{1} \times \frac{1}{1} = 0$$

$$P(\text{TV}|\text{near (not urgent) deadline, no party, lazy}) = \frac{1}{10} \times \frac{1}{1} \times \frac{1}{1} \times \frac{1}{1} = \frac{1}{10}$$

So based on this you will be watching TV tonight.

The Naïve Bayes Classifier example

- 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

The Naïve Bayes Classifier example contd..

The weather data, with counts and probabilities														
outlook			temperature			humidity			windy			play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

A new day				
outlook	temperature	humidity	windy	play
sunny	cool	high	true	?

The Naïve Bayes Classifier example contd..

- Test Phase
  - Given a new instance, predict its label

    $\mathbf{x}'$=(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)
  - Look up tables achieved in the learning phrase

    P(Outlook=*Sunny*|Play=*Yes*) = 2/9

    P(Temperature=*Cool*|Play=*Yes*) = 3/9

    P(Huminity=*High*|Play=*Yes*) = 3/9

    P(Wind=*Strong*|Play=*Yes*) = 3/9

    P(Play=*Yes*) = 9/14

    P(Outlook=*Sunny*|Play=*No*) = 3/5

    P(Temperature=*Cool*|Play==*No*) = 1/5

    P(Huminity=*High*|Play=*No*) = 4/5

    P(Wind=*Strong*|Play=*No*) = 3/5

    P(Play=*No*) = 5/14
  - Decision making with the MAP rule

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

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

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

The Naïve Bayes Classifier example

Example No.	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

We want to classify a Red Domestic SUV.

The Naïve Bayes Classifier example contd..

Color			type			Origin			stolen	
	Yes	No		Yes	No		Yes	No	Yes	No
Red	3	2	sports	4	2	Domestic	2	3	5	5
yellow	2	3	SUV	1	3	Imported	3	2		
Probabilities (w.r.t car steeling)										
	Yes	No		Yes	No		Yes	No	Yes	No
Red	3/5	2/5	sports	4/5	2/5	Domestic	2/5	3/5	5/10	5/10
yellow	2/5	3/5	SUV	1/5	3/5	Imported	3/5	2/5		

P(stolen =yes| Red, domestic, SUV)= p(stolen=yes)*p(red|stolen=yes)*p(domestic|stolen=yes)*p(SUV|stolen=yes)
=5/10*3/5*2/5*1/5=0.024

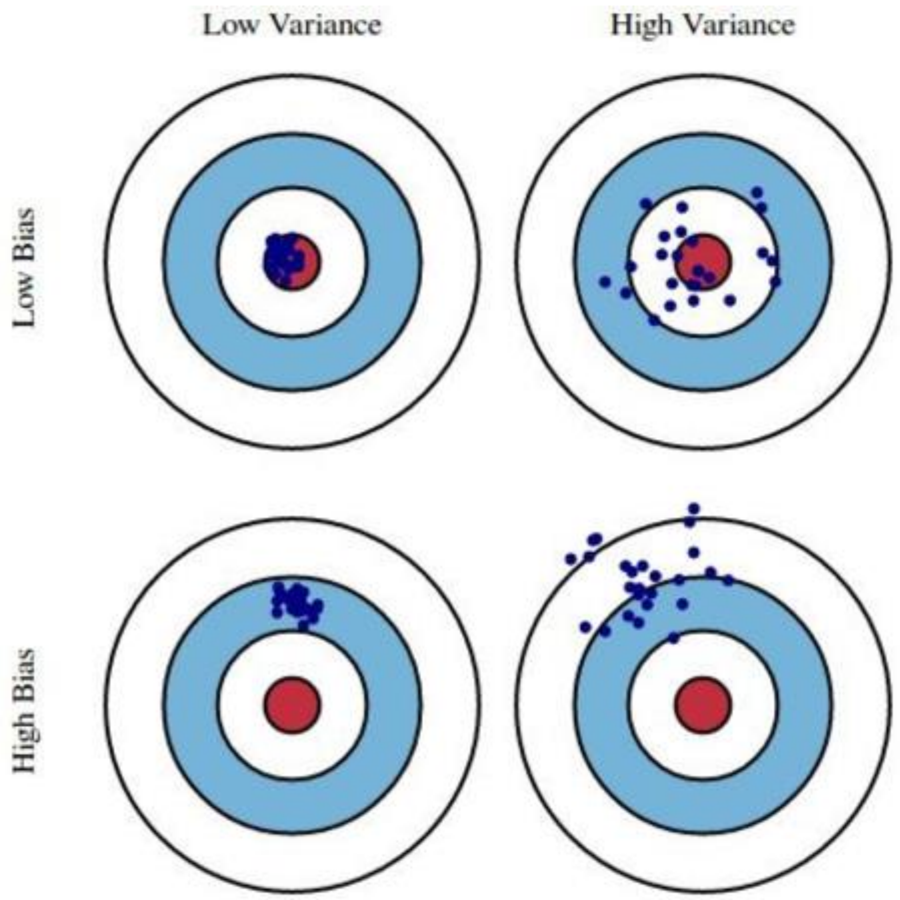
P(stolen =No| Red, domestic, SUV)= p(stolen=no)*p(red|stolen=no)*p(domestic|stolen=no)*p(SUV|stolen=no)
=5/10*2/5*3/5*3/5=0.072

P(stolen =yes| Red, domestic, SUV) < P(stolen =No| Red, domestic, SUV)
The Red Domestic SUV is not stolen

Trade-off between Bias-variance

"Bias is the algorithm's tendency to consistently learn the wrong thing by not taking into account all the information in the data (underfitting)."

"Variance is the algorithm's tendency to learn random things irrespective of the real signal by fitting highly flexible models that follow the error/noise in the data too closely (overfitting)."



Bull's eye diagram

Trade-off between Bias-variance contd..

Low Bias — High Variance:

- A low bias and high variance problem is overfitting. Different data sets are depicting insights given their respective dataset. Hence, the models will predict differently. However, if average the results, we will have a pretty accurate prediction.

High Bias — Low Variance:

- The predictions will be similar to one another but on average, they are inaccurate.

Trade-off between Bias-variance contd..

If you have HIGH VARIANCE PROBLEM:

- You can get more training examples because a larger the dataset is more probable to get a higher predictions.
- Try smaller sets of features (because you are overfitting)
- Try increasing lambda, so you can not over-fit the training set as much. The higher the lambda, the more the regularization applies, for Linear Regression with regularization.

If you have HIGH BIAS PROBLEM:

- Try getting additional features, you are generalizing the datasets.
- Try adding polynomial features, make the model more complicated.
- Try decreasing lambda, so you can try to fit the data better. The lower the lambda, the less the regularization applies, for Linear Regression with regularization.

Thank you