COMP47750 Tutorial

Naïve Bayes Classifiers

Pádraig Cunningham

School of Computer Science

Reminder: Naïve Bayes Classifier

- We apply Bayes Theorem with the naïve assumption that all features are *conditionally independent*:

$$P(f_1, f_2 \dots f_n | v_j) = \prod_i P(f_i | v_j)$$

i.e. the value of a particular feature is unrelated to the presence or absence of any other feature, given the class label v_j

- Based on this assumption, the objective of the Naïve Bayes classifier becomes:

Find the most probable class label v for x according to:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(f_i | v_j)$$

i.e. (Class Probability) x (Product of Class-Feature Probabilities)

Reminder: Naïve Bayes Classifier

- The Naïve Bayes classifier is an eager learner, which builds a model in advance. The algorithm operates as follows:
  - **Training Phase:**
    - Estimate the probabilities $P(v_j)$ and $P(f_i|v_j)$ based on their frequencies in the training set.
    - Involves building a contingency table (probability table) of conditional and prior (class) probabilities.
  - **Classification Phase:**
    - Classify new examples using the probability estimates and the Naïve Bayes formula to select the most likely class:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(f_i|v_j)$$

i.e. (Class Probability) x (Product of Class-Feature Probabilities)

Tutorial Q1

Given a contingency table of conditional and prior probabilities for a training set with 10 examples and 5 categorical features:

Swimming	Yes	No
Rain Recently=light	0/4	3/6
Rain Recently=moderate	2/4	3/6
Rain Recently=heavy	2/4	0/6
Rain Today=light	1/4	3/6
Rain Today=moderate	2/4	3/6
Rain Today=heavy	1/4	0/6
Temp=Cold	1/4	5/6
Temp=Warm	3/4	1/6
Wind=Light	2/4	2/6
Wind=Moderate	2/4	2/6
Wind=Gale	0/4	2/6
Sunshine=Some	2/4	4/6
Sunshine=None	2/4	2/6
Class Probabilities	4/10	6/10

Tutorial Q1

Based on the contingency table, classify the two new examples below using Naïve Bayes.

Example	Rain Recently (RR)	Rain Today (RT)	Temp (T)	Wind (W)	Sunshine (S)	Swimming
X1	Heavy	Moderate	Warm	Light	Some	???

Example	Rain Recently (RR)	Rain Today (RT)	Temp (T)	Wind (W)	Sunshine (S)	Swimming
X2	Light	Moderate	Warm	Light	Some	???

Naïve Bayes classification steps:

1. Calculate probability of input having class *Yes*
2. Calculate probability of input having class *No*
3. Normalise probabilities (optional)

Tutorial Q1: Input X1

Test input example for hypothesis 1: *Swimming=Yes*

Example	Rain Recently (RR)	Rain Today (RT)	Temp (T)	Wind (W)	Sunshine (S)	Swimming
X1	Heavy	Moderate	Warm	Light	Some	???

Identify the relevant rows in the contingency table for *Swimming=Yes*:

Swimming	Yes	No
Rain Recently=heavy	2/4	0/6
Rain Today=moderate	2/4	3/6
Temp=Warm	3/4	1/6
Wind=Light	2/4	2/6
Sunshine=Some	2/4	4/6
Class Probabilities	4/10	6/10

Apply NB for *Swimming=Yes* by calculating product of probabilities for input's feature values and class probability:

$$P = (2/4 \times 2/4 \times 3/4 \times 2/4 \times 2/4) \times 4/10$$

$$P = 0.01875$$

Tutorial Q1: Input X1

Test input example for hypothesis 2: *Swimming=No*

Example	Rain Recently (RR)	Rain Today (RT)	Temp (T)	Wind (W)	Sunshine (S)	Swimming
X1	Heavy	Moderate	Warm	Light	Some	???

Identify the relevant rows in the contingency table for *Swimming=No*:

Swimming	Yes	No
Rain Recently=heavy	2/4	0/6
Rain Today=moderate	2/4	3/6
Temp=Warm	3/4	1/6
Wind=Light	2/4	2/6
Sunshine=Some	2/4	4/6
Class Probabilities	4/10	6/10

Apply NB for *Swimming=No* by calculating product of probabilities for input's feature values and class probability:

$$P = (0/6 \times 3/6 \times 1/6 \times 2/6 \times 4/6) \times 6/10$$

$$P = 0$$

Tutorial Q1: Input X1

- We calculated probabilities for two hypotheses (class labels):

Yes `P(Y) = 2/4 x 2/4 x 3/4 x 2/4 x 2/4 x 4/10 = 0.01875`

No `P(N) = 0/6 x 3/6 x 1/6 x 2/6 x 4/6 x 6/10 = 0`

- Normalise probabilities to sum to 1:

Yes `P(Y)' = 0.01875/(0.01875+0) = 1.0`

No `P(N)' = 0`

- Output Prediction: **Swimming = Yes**

Tutorial Q1: Input X2

Test input example for hypothesis 1: *Swimming=Yes*

Example	Rain Recently (RR)	Rain Today (RT)	Temp (T)	Wind (W)	Sunshine (S)	Swimming
X2	Light	Moderate	Warm	Light	Some	???

Identify the relevant rows in the contingency table for *Swimming=Yes*:

Swimming	Yes	No
Rain Recently=light	0/4	3/6
Rain Today=moderate	2/4	3/6
Temp=Warm	3/4	1/6
Wind=Light	2/4	2/6
Sunshine=Some	2/4	4/6
Class Probabilities	4/10	6/10

Apply NB for *Swimming=Yes* by calculating product of probabilities for input's feature values and class probability:

```
P = (0/4 x 2/4 x 3/4 x 2/4 x 2/4) x 4/10
P = 0
```

Tutorial Q1: Input X2

Test input example for hypothesis 1: *Swimming=No*

Example	Rain Recently (RR)	Rain Today (RT)	Temp (T)	Wind (W)	Sunshine (S)	Swimming
X2	Light	Moderate	Warm	Light	Some	???

Identify the relevant rows in the contingency table for *Swimming=No*:

Swimming	Yes	No
Rain Recently=light	0/4	3/6
Rain Today=moderate	2/4	3/6
Temp=Warm	3/4	1/6
Wind=Light	2/4	2/6
Sunshine=Some	2/4	4/6
Class Probabilities	4/10	6/10

Apply NB for *Swimming=No* by calculating product of probabilities for input's feature values and class probability:

```
P = (3/6 x 3/6 x 1/6 x 2/6 x 4/6) x 6/10
P = 0.0056
```

Tutorial Q1: Input X2

- Calculated probabilities for two hypotheses (class labels):

Yes `P(Y) = (0/4 x 2/4 x 3/4 x 2/4 x 2/4) x 4/10 = 0`

No `P(N) = (3/6 x 3/6 x 1/6 x 2/6 x 4/6) x 6/10 = 0.0056`

- Normalise probabilities to sum to 1:

Yes `P(Y)' = 0`

No `P(N)' = 0.0056/(0.0056+0) = 1.0`

- Output Prediction: **Swimming = No**

Tutorial Q2(a)

a) Provide the contingency table of conditional and prior probabilities that would be used by Naïve Bayes to build a classifier for this dataset.

	Name	Hair	Height	Build	Lotion	Result
1	Sarah	blonde	average	light	no	sunburned
2	Dana	blonde	tall	average	yes	none
3	Alex	brown	short	average	yes	none
4	Annie	blonde	short	average	no	sunburned
5	Emily	red	average	heavy	no	sunburned
6	Pete	brown	tall	heavy	no	none
7	John	brown	average	heavy	no	none
8	Katie	brown	short	light	yes	none

Tutorial Q2(a)

Construct full contingency table for all features on both classes:

Feature Value	Sunburned	None
Hair=blonde		
Hair=brown		
Hair=red		
Height=average		
Height=tall		
Height=short		
Build=light		
Build=average		
Build=heavy		
Lotion=no		
Lotion=yes		
Class Probabilities		

Tutorial Q2(a)

Construct full contingency table for all features on both classes:

Feature Value	**Sunburned**	**None**
Hair=blonde	2/3	1/5
Hair=brown	0/3	4/5
Hair=red	1/3	0/5
Height=average	2/3	1/5
Height=tall	0/3	2/5
Height=short	1/3	2/5
Build=light	1/3	1/5
Build=average	1/3	2/5
Build=heavy	1/3	2/5
Lotion=no	3/3	2/5
Lotion=yes	0/3	3/5
Class Probabilities	3/8	5/8

Tutorial Q2(b)

Use the contingency table to calculate the Naïve Bayes scores:

	Hair	Height	Build	Lotion	Result
X	blonde	average	heavy	no	???

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(f_i|v_j)$$

Result	Sunburned	None
Hair=blonde	2/3	1/5
Height=average	2/3	1/5
Build=heavy	1/3	2/5
Lotion=no	3/3	2/5
Class Probabilities	3/8	5/8

Calculate raw probabilities for two classes:

```
P(S) = (2/3)x(2/3)x(1/3)x(3/3) x (3/8)
P(S) = 0.056

P(N) = (1/5)x(1/5)x(2/5)x(2/5) x (5/8)
P(N) = 0.004
```

Normalise probabilities:

```
P(S)' = 0.056/(0.056+0.004) = 0.933
P(N)' = 0.004/(0.056+0.004) = 0.067
```

➡ **Output: Sunburned**

Tutorial Q3(a)

a) Calculate the contingency table that would be used by Naïve Bayes to build a classifier using this training data.

Example	Credit History	Debt	Income	Risk
1	bad	low	0to30	high
2	bad	high	30to60	high
3	bad	low	0to30	high
4	unknown	high	30to60	high
5	unknown	high	0to30	high
6	good	high	0to30	high
7	bad	low	over60	medium
8	unknown	low	30to60	medium
9	good	high	30to60	medium
10	unknown	low	over60	low
11	unknown	low	over60	low
12	good	low	over60	low
13	good	high	over60	low
14	good	high	over60	low

Tutorial Q3(a)

a) Calculate the contingency table that would be used by Naïve Bayes to build a classifier using this training data.

Contingency table for each of the descriptive features across 3 classes:

Risk	high	medium	low
CH=bad			
CH=unknown			
CH=good			
Debt=low			
Debt=high			
Income=0to30			
Income=30to60			
Income=over60			
Class Probabilities			

Tutorial Q3(a)

a) Calculate the contingency table that would be used by Naïve Bayes to build a classifier using this training data.

Contingency table for each of the descriptive features across 3 classes:

Risk	high	medium	low
CH=bad	3/6	1/3	0
CH=unknown	2/6	1/3	2/5
CH=good	1/6	1/3	3/5
Debt=low	2/6	2/3	3/5
Debt=high	4/6	1/3	2/5
Income=0to30	4/6	0	0
Income=30to60	2/6	2/3	0
Income=over60	0	1/3	5/5
Class Probabilities	6/14	3/14	5/14

Tutorial Q3(a)

b) Predict the risk level for the new loan application X below.

	Credit History	Debt	Income	Risk
X	bad	low	30to60	???

Risk	high	medium	low
CH=bad	3/6	1/3	0
Debt=low	2/6	2/3	3/5
Income=30to60	2/6	2/3	0
Class Probabilities	6/14	3/14	5/14

Calculate raw probabilities for 3 classes, using contingency table:

```
P(H) = (3/6)x(2/6)x(2/6) x (6/14) = 0.0238
P(M) = (1/3)x(2/3)x(2/3) x (3/14) = 0.0317
P(L) = (0)x(3/5)x(0) x (5/14) = 0
```

Normalise probabilities:

```
P(H)' = 0.0238/(0.0238+0.0317+0) = 0.4288
P(M)' = 0.0317/(0.0238+0.0317+0) = 0.5712
P(L)' = 0
```

➡ **Output: Medium Risk**

Tutorial 4(a)

4.(a) Given the nature of the `AthleteSelection` data which would be the best of the Naive Bayes options in scikit-learn for that classification task?

Gaussian Naive Bayes is an option because the features are real values - not counts or categories. The data is probably not exactly Gaussian but probably close enough.

We could discretize the data and then use Categorical NB.

Tutorial 4(b)

4.(b) A ranking classifier is a classifier that can rank a test set in order of confidence for a given classification outcome. Naive Bayes is a ranking classifier because the 'probability' can be used as a confidence measure for ranking.

1. Train a Naive Bayes classifier from the `AthleteSelection` data. Load the test data from `AthleteTest.csv` and apply the classifier.
2. Use the `predict_proba` method to find the probability of being selected.
3. Rank the test set by probability of being selected.
   - 3.1. Who is most likely to be selected?
   - 3.2. Who is least likely?

Some code for this exercise is available in the notebook **`07 Naive Bayes Tutorial`**. You will also need to download the test data file **'AthleteTest.csv'**.

Tutorial 4(b)

```
gnb = GaussianNB()
ath_NB = gnb.fit(X,y)

y_probs = ath_NB.predict_proba(X_test)
ath_test['Prob']=y_probs[:,1]
ath_test.sort_values(by=['Prob'], ascending=False, inplace = True)
ath_test
```

```
y_probs
Out[12]:
array([[9.58686371e-01, 4.13136290e-02],
       [8.77017219e-01, 1.22982781e-01],
       [8.80671574e-02, 9.11932843e-01],
       [8.49522335e-01, 1.50477665e-01],
       [2.00167162e-01, 7.99832838e-01],
       [2.64304710e-06, 9.99997357e-01],
       [5.48092049e-05, 9.99945191e-01],
       [2.70690822e-02, 9.72930918e-01],
       [1.45717357e-01, 8.54282643e-01],
       [2.70690822e-02, 9.72930918e-01]])
In [ ]:
```

	Speed	Agility	Prob
Athlete			
t6	8.1	7.8	0.999997
t7	7.7	5.2	0.999945
t8	6.1	5.5	0.972931
t10	6.1	5.5	0.972931
t3	5.5	7.2	0.911933
t9	5.5	6.0	0.854283
t5	5.5	5.2	0.799833
t4	3.8	8.8	0.150478
t2	4.5	4.5	0.122983
t1	3.3	8.2	0.041314

Tutorial 4(c)

When a **`GaussianNB`** model is trained the model is stored in two parameters **'theta_'** and **'var_'.** Train a **`GaussianNB`** model and check to see if these parameters agree with your own estimates.

Hint: this code will give you the estimates you need.

```
athlete[athlete['Selected']=='No']['Agility'].mean()

athlete[athlete['Selected']=='No']['Agility'].var(ddof=0)
```

The **'var_'** parameter contains the square of the standard deviation (the variance) rather than the standard deviations. The figures should agree exactly if **ddof** is set to zero is the **var** calculation.

Question 4(c)

Athlete	Speed	Agility	Selected
x1	2.50	6.00	No
x2	3.75	8.00	No
x3	2.25	5.50	No
x4	3.25	8.25	No
x5	2.75	7.50	No

```
ath_NB.var_
array([[0.80685764, 3.99305556],
       [1.37402344, 3.91308594]])
ath_NB.theta_
array([[3.39583333, 5.08333333],
       [6.40625   , 6.96875   ]])
athlete[athlete['Selected']=='Yes']['Speed'].mean()
6.40625
athlete[athlete['Selected']=='Yes']['Speed'].var(ddof=0)
1.3740234375
athlete[athlete['Selected']=='No']['Speed'].mean()
3.3958333333333335
athlete[athlete['Selected']=='No']['Speed'].var(ddof=0)
0.8068576388888888
```