F20/21DL. Data Mining and Machine Learning

Lab 5. Bayesian Learning and Bayes Nets

Covering Practical work to be done by students in Week 5

The purpose of this lab is:

1. to practice what we have learned so far:
   - Bayesian Probabilities
   - Bayes Nets and algorithms that underpin them
   - Practical considerations in running Bayes nets on the given data
2. to help you to make progress with Python tutorial and your DM & ML portfolio.

1 Lectures and Weekly Tests

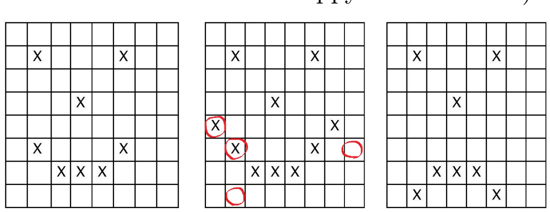
1.1 Bayes Rule, Bayesian Probabilities

1. Read or watch the lecture slides, make sure you understand them, ask questions.
2. **Preparing for Part 1 of the Test 2.1: Bayesian probabilities – basic intuition.**

   Take the following toy data set for **Face Emotions Recognition**:

Picture	Cell 33	Cell 42	Cell 48	Cell 58	Face expression
P1	White	Black	White	White	Happy
P2	Black	Black	White	White	Happy
P3	White	White	White	Black	Sad
P4	White	White	Black	White	Happy
P5	White	White	White	White	Happy
P6	White	White	Black	Black	Sad
P7	Black	White	White	Black	Sad
P8	Black	White	Black	Black	Sad
P9	Black	Black	Black	White	Happy
P10	White	Black	White	Black	Happy

   $P1 - P10$ stand for 10 pictures, cells stand for the highlighted cells in the below picture (that also shows the intuition behind the happy and sad faces):



   Using the formulas from the lecture slides, answer the following questions:

   (a) What is the Prior Probability $P(A)$ of the target feature A, where A is the following event
   - "The grid face is Happy"

(b) Compute the Conditional Probability $P(B|A) = \frac{P(A \wedge B)}{P(A)}$ of event B given event A, where A is as above, and B is the training feature:

- "Cell 42 is white"

(c) Compute the Bayesian Probability $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$ of event A given event B.

(d) Was the knowledge about A revised after observation of B?

1.2 Algorithms behind the Bayes nets

1. In the **DATA MINING** textbook (2011, 2017, Witten et al), read §4.2 p. 90-94 (in 2017 edition, p. 96-100). It shows how to convert a data set into Bayes factors, which constitute a Naive Bayes net.

2. **Prepare for Test 2.1, Part 2: Algorithms Behind Naive Bayes Nets:**

(a) Take the small emotion recognition set as above. For it, produce the same **Table with Counts and Probabilities** as on p.91: it defines a Naive Bayes net for the data set. In your table, the order of columns should be: Cell33, Cell42, Cell48, Cell58, Emotion.

Please convert all fractions in your table to the decimal format rounding to two ***non-zero*** *decimal points. For example, if you have* $\frac{1}{6}$*, it should be converted to* 0.17*. If you have numbers such as* 0.000537...*, you will round to* 0.00054. All test questions will be referring to this format.

(b) Take an example (observation of a new picture)

$$(Cell33 = White), (Cell42 = Black), (Cell48 = Black), (Cell58 = White)$$

i. Use your **Table with Counts and Probabilities** and the formulae given on pp.92-93 to compute the **likelihood of a face being "Happy"** given the observation.

ii. *(Harder, for higher marks)* Use your **Table with Counts and Probabilities** and the formulae given on pp.92-93 to compute the **likelihood of a face being "Sad"** However, in this case, you will need to use the *Laplace Estimator* in order for calculations to work. Please use the Laplace estimator = 1 when you do this exercise, to arrive to the intended answers in the test.

iii. *(Harder, for higher marks)* Use the computed likelihoods[1] to compute **probabilities**:

$$P(Happy|Cell33 = White, Cell42 = Black, Cell48 = Black, Cell58 = White)$$

$$P(Sad|Cell33 = White, Cell42 = Black, Cell48 = Black, Cell58 = White)$$

iv. Do the resulting probabilities correspond to your intuition of Happy/Sad faces (in the data set)?

(c) *(Harder, for higher marks)* Consider the following two factors f_1 and f_2, and compute their product $f_1 \times f_2$:

f_1:

$Cell33$	$Cell48$	val
White	White	0.67
White	Black	0.5
Black	White	0.33
Black	Black	0.5

f_2:

$Cell48$	$Emotion$	val
White	Happy	0.67
White	Sad	0.2
Black	Happy	0.33
Black	Sad	0.8

$f_1 \times f_2$:

	$Cell33$	$Cell48$	$Emotion$	val
1	White	White	Happy	?
2	White	White	Sad	?
3	White	Black	Happy	?
4	White	Black	Sad	?
5	Black	White	Happy	?
6	Black	White	Sad	?
7	Black	Black	Happy	?
8	Black	Black	Sad	?

(d) *(Harder, for higher marks)* Consider the following factor $f_3(Cell33, Cell42, Emotion)$:

[1] We are ok with the fact that the likelihood for Happy did not need the Laplace estimator trick, and the likelihood for Sad – did. Just use them as computed in previous steps, to avoid extra manual calculation. When it comes to implementation, the Laplace estimator is used consistently all the time. This is just an exercise.

$Cell33$	$Cell42$	$Emotion$	val
White	White	Happy	0.066
White	White	Sad	0.45
White	Black	Happy	0.13
White	Black	Sad	0.21
Black	White	Happy	0.25
Black	White	Sad	0.13
Black	Black	Happy	0.11
Black	Black	Sad	0.29

and compute $\sum_{Cell42} f_3$.

(e) *(Harder, for higher marks)* Consider the factor $f_3(Cell33, Cell42, Emotion)$ again and compute

$$f_3(Cell33 = White, Cell42, Emotion).$$

2 DM & ML Portfolio

This part is to be completed in groups, and will be assessed during the labs. Marking scheme: this lab will bring you up to 2 points. 1 point for completing the task, 1 additional point for any non-trivial analytical work with the material.

2.1 Python Tutorial and Programming Practice (Prior to the lab)

This part is for your individual programming practice during the week.

- Read the `Scikit-Learn` library documentation that covers Bayes nets: `https://scikit-learn.org/stable/modules/naive_bayes.html#`
- The algorithm that we cover in the lectures is implemented as **"Multinomial Naive Bayes"** in the `Scikit-Learn`. At the very least, you should try to run it this week as part of your Portfolio building.

  Allowing for some adaptation to your own code, this will amount to running a bit of code like this:

```
from sklearn.naive_bayes import MultinomialNB

model = MultinomialNB(alpha=0.0, class_prior=[0.4, 0.6])

model.fit(X, Y)
```

- Repeat the evaluation steps from the last lab, replacing the classifier you used with the Bayes net. Check: does it have better performance in any of the evaluation metrics?
- *(optional for BSc, Mandatory for MSc)* Study other Bayes net algorithms given in `Scikit-Learn`:
  - Gaussian Naive Bayes
  - Complement Naive Bayes
  - Bernoulli Naive Bayes
  - Categorical Naive Bayes

  Again, use the standard evaluation metrics to see whether any of these algorithms bring improvement to either overall accuracy or one of the evaluation metrics.
- *(hard, optional)* There are several implementations of complex Bayes Nets (i.e. Bayes nets with complex hierarchical structure) available on-line.

  Investigate whether any of these libraries can work with your data sets, and whether this brings any improvement. Note: complex Bayes nets can show non-trivial dependencies among features. So, they may have better explanatory power.

2.2 During the lab:

- Share your proposed solutions, discuss which Bayes net implementations and which evaluation metrics worked best. If some worked better, can you hypothesise why?
- Within the Portfolio text, fill in the table:

Naive Bayes Algorithm	Accuracy	TP	FP	TN	TP	Senitivity	Specificity	Precision	Recall	Area Under RoC Curve
Multinomial Naive Bayes										
Gaussian Naive Bayes										
Complement Naive Bayes										
Bernoulli Naive Bayes										
Categorical Naive Bayes										
Complex Bayes net (optional)										

- In the table, highlight with bold the best performing algorithms in each category.
- Make conclusions. You will get more marks for more interesting and “out of the box” questions and answers.
- **The tutors will mark:** quality of your code, completeness of your table and your analysis of the table (algorithms versus metrics performance).

2.3 After the lab:

- *Group rep:* Make sure all group members have tasks for the week
- *Everyone:* Incorporate the discussion during the lab into your Python code
- *Everyone:* Incorporate all code used in the lab into your Portfolio repository.