OSLOMET

ACIT4830 – Special Robotics and Control Subject

Topic2 – Classification using Naive Bayes

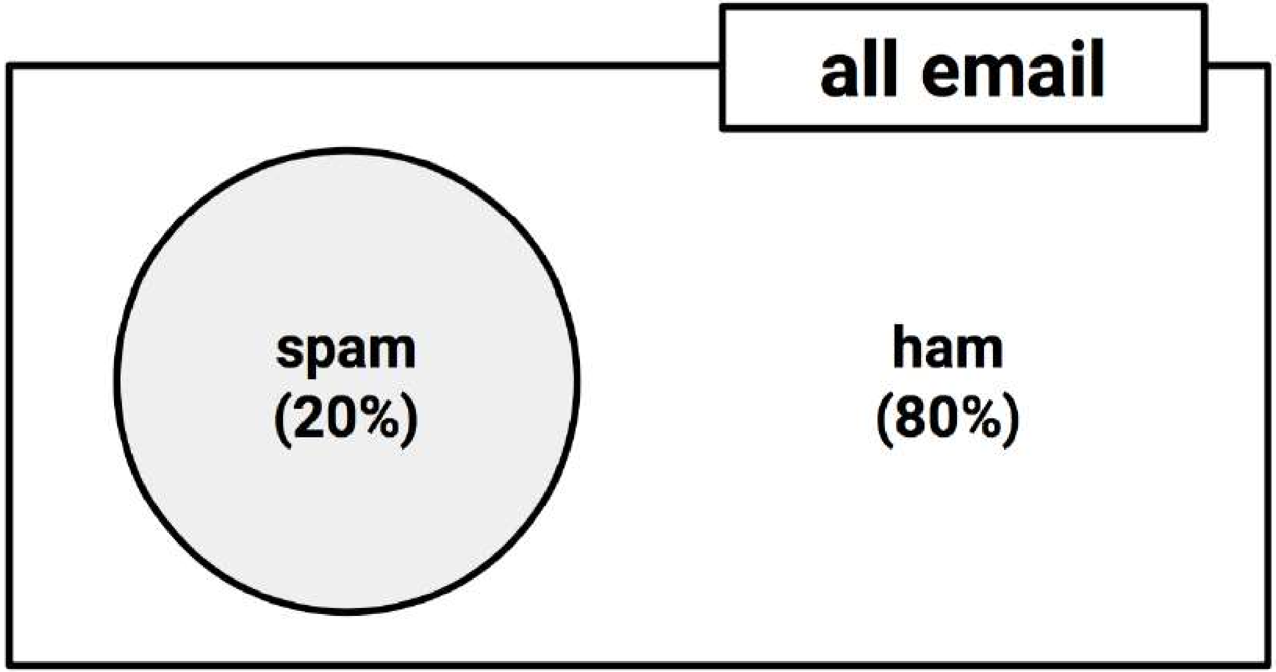
Evi Zouganeli
OsloMet – Oslo Metropolitan University
(evizou@oslomet.no)

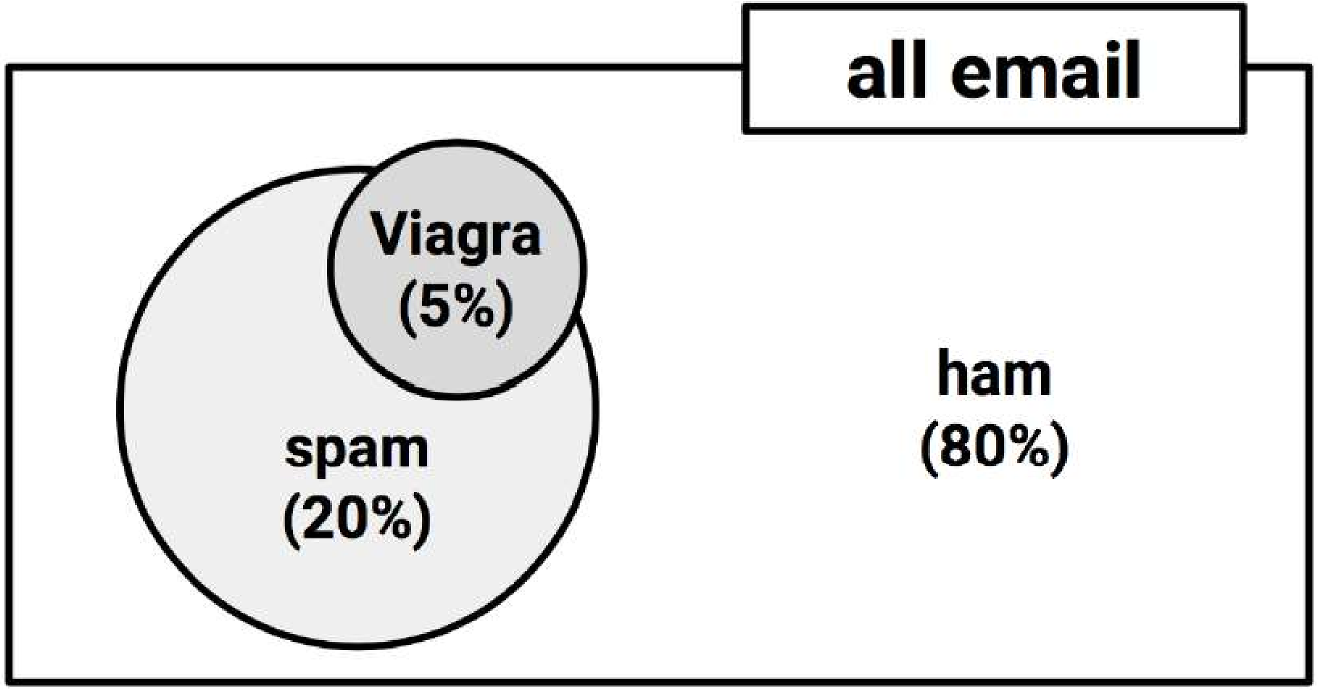
Content

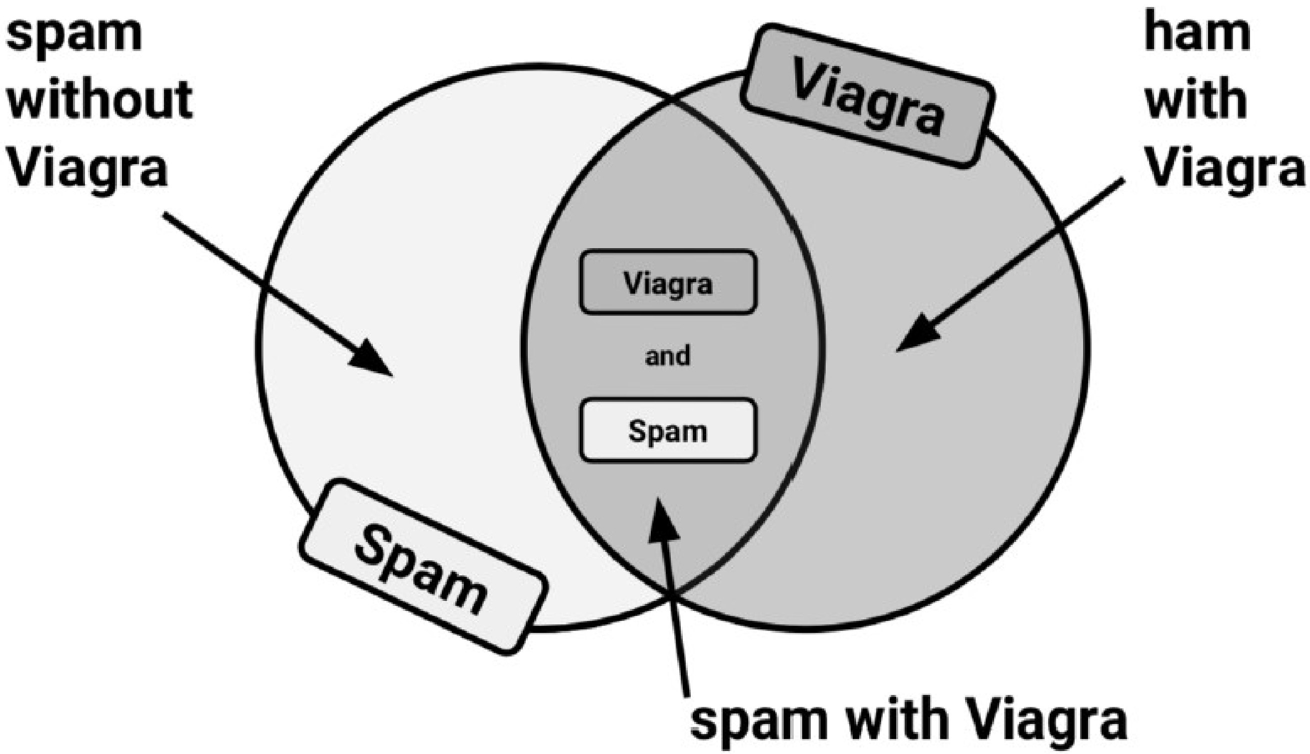
- Probability, joint probability
- Conditional Probability
- Bayes Theorem
- Classification using Naïve Bayes algorithm
- Hands-on exercise

Naive Bayes algorithm

Strengths	Weaknesses
• Simple, fast, and very effective • Does well with noisy and missing data • Requires relatively few examples for training, but also works well with very large numbers of examples • Easy to obtain the estimated probability for a prediction	• Relies on an often-faulty assumption of equally important and independent features • Not ideal for datasets with large numbers of numeric features • Estimated probabilities are less reliable than the predicted classes


all email
spam
(20%)
ham
(80%)


all email
Viagra
(5%)
spam
(20%)
ham
(80%)


spam
without
Viagra
Viagra
ham
with
Viagra
Viagra
and
Spam
Spam
spam with Viagra

Joint Probability A and B

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

Conditional Probability of A given B

likelihood

prior probability

$$P(\text{spam} \mid \text{Viagra}) = \frac{P(\text{Viagra} \mid \text{spam})\,P(\text{spam})}{P(\text{Viagra})}$$

posterior probability

marginal likelihood

Frequency and Likelihood

	Viagra		
Frequency	**Yes**	**No**	**Total**
spam	4	16	20
ham	1	79	80
Total	5	95	100

	Viagra		
Likelihood	**Yes**	**No**	**Total**
spam	4 / 20	16 / 20	20
ham	1 / 80	79 / 80	80
Total	5 / 100	95 / 100	100

	Viagra (W_1)		Money (W_2)		Groceries (W_3)		Unsubscribe (W_4)		
Likelihood	**Yes**	**No**	**Yes**	**No**	**Yes**	**No**	**Yes**	**No**	**Total**
spam	4 / 20	16 / 20	10 / 20	10 / 20	0 / 20	20 / 20	12 / 20	8 / 20	20
ham	1 / 80	79 / 80	14 / 80	66 / 80	8 / 80	71 / 80	23 / 80	57 / 80	80
Total	5 / 100	95 / 100	24 / 100	76 / 100	8 / 100	91 / 100	35 / 100	65 / 100	100

S: Spam
H: Ham (not spam)
B: 'Buy'
C: 'Cheap'

Naive Bayes

$$P(S \mid B \cap C) = \frac{P(B \mid S)\,P(C \mid S)\,P(S)}{P(B \mid S)\,P(C \mid S)\,P(S) + P(B \mid H)\,P(C \mid H)\,P(H)}$$

Laplace estimator/ smoothing: correction to ensure that probability terms are non-zero

Word Cloud – R function wordcloud

Data cleaning for text processing - example

- Make all text lower case
- Remove “stopwords” (e.g. and, but), and numbers
- Remove punctuation; replace w/ white spaces if needed
- Keep stem of words
- Remove white spaces

\+

- **Document-Term Matrix** (**DTM**) format: rows are documents, columns are words; and transposed: **TDM**

Data cleaning illustration

SMS messages before cleaning	SMS messages after cleaning
`> inspect(sms_corpus[1:3])`	`> inspect(corpus_clean[1:3])`
`[[1]]`	`[[1]]`
`Hope you are having a good week. Just checking in`	`hope good week just checking`
`[[2]]`	`[[2]]`
`K..give back my thanks.`	`kgive back thanks`
`[[3]]`	`[[3]]`
`Am also doing in cbe only. But have to pay.`	`also cbe pay`

Naive Bayes algorithm in R

Naive Bayes classification syntax

using the `naiveBayes()` function in the `e1071` package

Building the classifier:

```
m <- naiveBayes(train, class, laplace = 0)
```

- `train` is a data frame or matrix containing training data
- `class` is a factor vector with the class for each row in the training data
- `laplace` is a number to control the Laplace estimator (by default, 0)

The function will return a naive Bayes model object that can be used to make predictions.

Making predictions:

```
p <- predict(m, test, type = "class")
```

- `m` is a model trained by the `naiveBayes()` function
- `test` is a data frame or matrix containing test data with the same features as the training data used to build the classifier
- `type` is either `"class"` or `"raw"` and specifies whether the predictions should be the most likely class value or the raw predicted probabilities

The function will return a vector of predicted class values or raw predicted probabilities depending upon the value of the `type` parameter.

Example:

```
sms_classifier <- naiveBayes(sms_train, sms_type)
sms_predictions <- predict(sms_classifier, sms_test)
```

```
Total Observations in Table:  1390

             | actual
   predicted |       ham |      spam | Row Total |
-------------|-----------|-----------|-----------|
         ham |      1202 |        28 |      1230 |
             |     0.996 |     0.153 |           |
-------------|-----------|-----------|-----------|
        spam |         5 |       155 |       160 |
             |     0.004 |     0.847 |           |
-------------|-----------|-----------|-----------|
Column Total |      1207 |       183 |      1390 |
             |     0.868 |     0.132 |           |
-------------|-----------|-----------|-----------|
```

False positive

False negative

Confusion Matrix