Learning from Data
Lecture 1: General Principles, Evaluation, Naive Bayes

Malvina Nissim
m.nissim@rug.nl
room 1311.421

14 November 2016

- **Lecturers**:
  - Malvina Nissim: `m.nissim@rug.nl`
  - Hessel Haagsma: `hessel.haagsma@rug.nl`
- **Monday 11-13**: lectures
- **Thursday 13-15**: labs

Readings

- We will suggest various readings along the course, and they will all be accessible on Nestor whenever possible (via links or uploaded PDFs).
- We are not going to use a single textbook for this class, but these are some references you can use to look up topics:

Readings

- with a focus on NLP:
  - Christopher D. Manning and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press. Cambridge, MA. 1999. `http://nlp.stanford.edu/fsnlp/`
  - Christopher D. Manning, Prabhakar Raghavan and Hinrich Schtze, *Introduction to Information Retrieval*, Cambridge University Press. 2008. `http://nlp.stanford.edu/IR-book/`
  - James Pustejovsky and Amber Stubbs *Natural Language Annotation for Machine Learning*, O'Reilly. 2012.
  - Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*, O'Reilly. 2009. `http://www.nltk.org/`
  - Hal Daumé III. *A course in Machine Learning*. `http://ciml.info/` (incomplete manuscript available online – some parts available for free.)

Readings

- more generally on machine learning:
  - Tom Mitchell, *Machine Learning*, McGraw Hill. 1997.
  - Ian H. Witten, Eibe Frank, Mark A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, The Morgan Kaufmann Series in Data Management Systems. 2011.
  - Yaser S. Abu-Mostafa, Malik Magdon-Ismail, Hsuan-Tien Lin, *Learning from Data*, AMLBook. 2012.
  - Peter Flach, *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*, Cambridge University Press. 2012.

Readings

- more specific to Scikit learn (and ML with Python):
  - Luis Pedro Coehlo and Willi Richert, *Building Machine Learning Systems with Python*, PACKT Publishing. 2013.
  - Raúl Garreta and Guillermo Moncecchi, *Learning scikit-learn: Machine Learning in Python*, PACKT Publishing. 2013.

Tools and Libraries

- NLTK
- Scikit-learn
- Keras
- Gensim

Assignments and grading

NB: Studiehandleiding on Nestor

- **marking (no exam)**
  - weekly assignments: each one 1–10
  - final project (coding, report, presentation): 1–10
  - combination of scores: the average of the exercises will make up 50% of the final score. The remaining 50% will be assigned through the project's evaluation.
- **deadlines**
  you should hand in your homework by the deadline, which is normally the following Monday, at 11pm. Please, do not ask for extensions!

Please, submit all the assignments via Nestor

Learning from Data

Learning from Data

learning what?

Learning from Data

what data?

learning to **predict**

Prediction

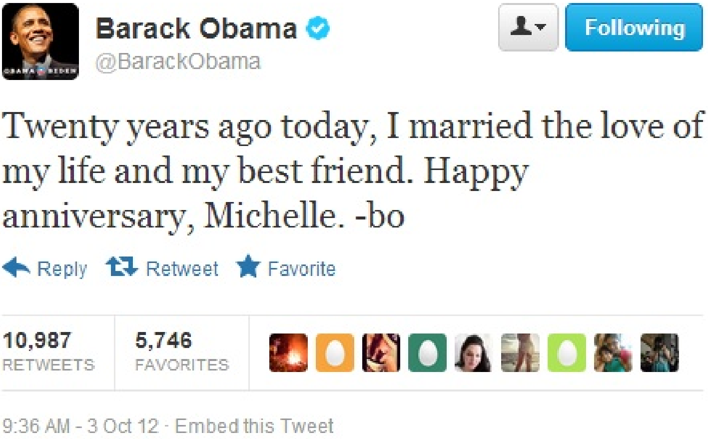
you are given some object — you have to **make a prediction**:

- is today a good day for playing football?
- is this tweet positive or negative?
- is the fourth word in this sentence a verb?
- is this article about the New York marathon?
- does this image contain a train?

Prediction

you are given some object — you have to **make a prediction**:

- is today a good day for playing football?
- is this tweet positive or negative?



Prediction

you are given some object — you have to **make a prediction**:

- is today a good day for playing football?
- is this tweet positive or negative?
- is the fourth word in this sentence a verb?
- is this article about the New York marathon?
- does this image contain a train?

Prediction

you are given some object — you have to **make a prediction**:

- is today a good day for playing football?
- is this tweet positive or negative?
- is the fourth word in this sentence a verb?
- is this article about the New York marathon?

Registration for the 2017 Boston Marathon will open on Monday, Sept. 12, the Boston Athletic Association announced Thursday.

Registration will be the same as in recent years and the fastest qualifiers will again be allowed to register first. The first two days of registration will be for runners who have hit their age group qualifying standard by 20 minutes or better, and then the requirements for registration are reduced in the following days.

Last year, runners needed to be 2 minutes, 28 seconds faster than their qualifying standard to get into the 2016 Boston Marathon, and more than 4,000 qualified runners were not accepted into the field of approximately 30,000 runners. The qualifying standards have not changed for 2017.

The 2017 Boston Marathon will be run on April 17.

Registration schedule

Sept. 12: Runners 20 minutes or faster than age group qualifying standard

Prediction

you are given some object — you have to **make a prediction**:

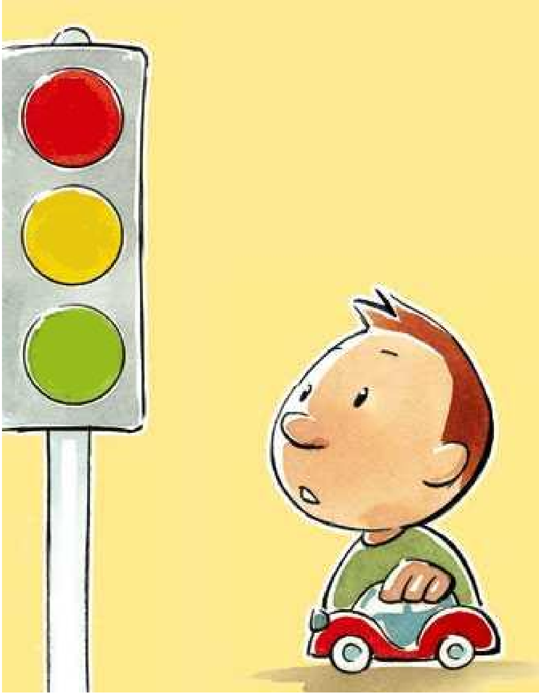
- is today a good day for playing football?
- is this tweet positive or negative?
- is the fourth word in this sentence a verb?
- is this article about the New York marathon?
- does this image contain a train?

Prediction

you are given some object — you have to **make a prediction**:

- is today a good day for playing football?
- is this tweet positive or negative?
- is the fourth word in this sentence a verb?
- is this article about the New York marathon?
- does this image contain a train?

learning = making such predictions by **observing data**



What to do in front of a traffic light?

$\boxed{\text{STOP}}$ or $\boxed{\text{GO}}$?

What to do in front of a traffic light?

$\boxed{\text{STOP}}$ or $\boxed{\text{GO}}$?

Options to teach the appropriate behaviour:

- create a set of *ad hoc rules*, as exhaustive as possible
- collect a set of *real examples* of people's behaviour at a traffic light

What to do in front of a traffic light?

$\boxed{\text{STOP}}$ or $\boxed{\text{GO}}$?

Options to teach the appropriate behaviour:

- create a set of *ad hoc rules*, as exhaustive as possible
- collect a set of *real examples* of people's behaviour at a traffic light

- rules:
  - *if* the light is red, *then* stop
  - *if* the light is green, *then* go
  - *if* the light is yellow, *then* if . . .

What to do in front of a traffic light?

$\boxed{\text{STOP}}$ or $\boxed{\text{GO}}$?

Options to teach the appropriate behaviour:

- create a set of *ad hoc rules*, as exhaustive as possible
- collect a set of *real examples* of people's behaviour at a traffic light

- rules:
  - *if* the light is red, *then* stop
  - *if* the light is green, *then* go
  - *if* the light is yellow, *then* if ...
- examples:
  - collection of examples of behaviour at a traffic light
  - cases are characterised by a set of features (light colour, speed, distance from traffic light, ...) and a result (stop, go)
  - induction and generalisation from observed examples

why do we want to **build** a predicting function from the examples rather than just implementing it?

why do we want to **build** a predicting function from the examples rather than just implementing it?

- often we don't know how to write down the function
- often a hand-written function isn't complete
- what is more expensive here: (acquiring accurate) knowledge or data?

- we have a set of examples and we want to obtain an inference scheme to model our data: we want to **generalise**
- our model is **general enough** if it can describe yet unseen examples (with an acceptable error rate)

$\boxed{\text{learning from data} = \text{inferring what we don't know from what we know}}$

A classic: Text classification

Text classification:

- topic classification
- spam detection
- authorship identification
- author profiling (age, gender, etc)
- sentiment analysis
- ...

A classic: Text classification

input:

- a document d
- a fixed set of classes $C = \{c_1, c_2, ..., c_n\}$

output:

- a predicted class $c \in C$

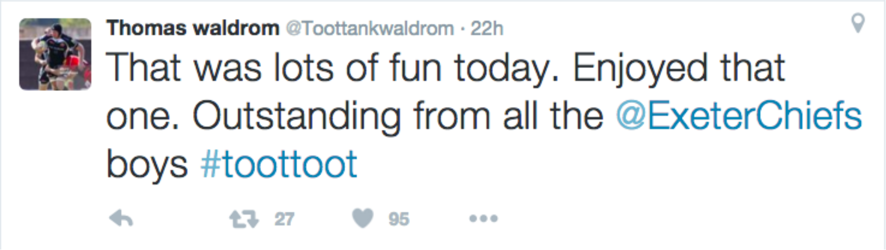
Learning from examples

positive

negative

Learning from examples

predict:



[positive] or [negative]?

Using examples

Can we just use examples as they are?

Using examples

Can we just use examples as they are?

- we need to transform examples into something a machine can understand
- we need to tell the machine what to look for,
  what the relevant aspects of the phenomenon are.

Using examples

in other words:

- we need to turn each example into some sort of machine-readable summary of itself (choosing relevant features)
- $\rightarrow$ our examples must become vectors of feature values

what *are* relevant features?

Clues as Features

just a quick terminology check:

- instances: all examples to learn from
- features: relevant traits that define the instances
- classes: what needs to be predicted

Clues as Features

- we know what we want to learn (target class):
  - for example: $\boxed{\text{positive}}$ or $\boxed{\text{negative}}$

Clues as Features

- we know what we want to learn (target class):
  - for example: $\boxed{\text{positive}}$ or $\boxed{\text{negative}}$
- we have a set of examples to learn from (instances)

Clues as Features

- we know what we want to learn (target class):
  - for example: $\boxed{\text{positive}}$ or $\boxed{\text{negative}}$
- we have a set of examples to learn from (instances)
- what clues might be useful to guess the class from the examples?

Clues as Features

- we know what we want to learn (target class):
  - for example: $\boxed{\text{positive}}$ or $\boxed{\text{negative}}$
- we have a set of examples to learn from (instances)
- what clues might be useful to guess the class from the examples?
  - words in the text
  - types of words in the text (nouns, adjectives, adverbs, ...)
  - (time of) day
  - id of the twitter user
  - ...

  clues $\rightarrow$ features (possible predictors)

  observed occurrences $\rightarrow$ feature values

Bag of words

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.

Bag of words

I **love** this movie! It's **sweet**, but with **satirical** humor. The dialogue is **great** and the adventure scenes are **fun**... It manages to be **whimsical** and **romantic** while **laughing** at the conventions of the fairy tale genre. I would **recommend** it to just about anyone. I've seen it **several** times, and I'm always **happy** to see it **again** whenever I have a friend who hasn't seen it yet.

Bag of words

x love xxxxxxxxxxxxxxxx sweet
xxxxxxx satirical xxxxxxxxxx
xxxxxxxxxxx great xxxxxxx
xxxxxxxxxxxxxxxxxxx fun xxxx
xxxxxxxxxxxxx whimsical xxxx
romantic xxxx laughing
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxx recommend xxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx several xxxxxxxxxxxxxxxxx
xxxxx happy xxxxxxxxx again
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxx

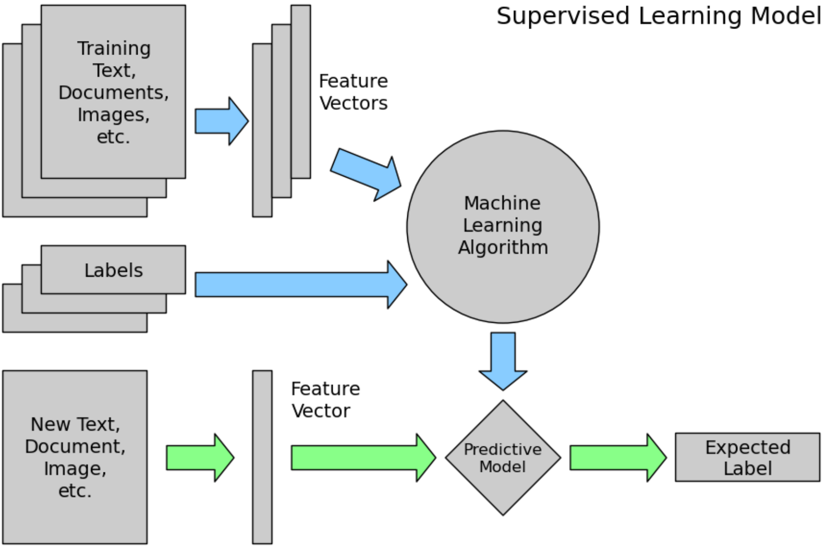
Bag of words

great	2
love	2
recommend	1
laugh	1
happy	1
...	...

What happens in learning, then?

- the learning algorithm observes **given examples**
- it tries to find common patterns that explain the data: it tries to **generalise** so that predictions can be made for **new examples**
- exactly how this is done depends on what **algorithm we are using**

What happens in learning, then?



source: `http://www.astroml.org/`

What happens in learning, then?

- the learning algorithm observes **given examples**
- it tries to find common patterns that explain the data: it tries to **generalise** so that predictions can be made for **new examples**
- exactly how this is done depends on what **algorithm we are using**

keywords here:

- given/new examples
- generalising
- algorithm we are using

What happens in learning, then?

- the learning algorithm observes **given examples**
- it tries to find common patterns that explain the data: it tries to **generalise** so that predictions can be made for **new examples**
- exactly how this is done depends on what **algorithm we are using**

keywords here:

- given/new examples
  - the settings of a learning experiment are important
- generalising
- algorithm we are using

What happens in learning, then?

- the learning algorithm observes **given examples**
- it tries to find common patterns that explain the data: it tries to **generalise** so that predictions can be made for **new examples**
- exactly how this is done depends on what **algorithm we are using**

keywords here:

- given/new examples
- generalising
  - what does it mean to generalise well?
- algorithm we are using

What happens in learning, then?

- the learning algorithm observes **given examples**
- it tries to find common patterns that explain the data: it tries to **generalise** so that predictions can be made for **new examples**
- exactly how this is done depends on what **algorithm we are using**

keywords here:

- given/new examples
- generalising
- algorithm we are using
  - we'll see one today (we'll see several more in this course)

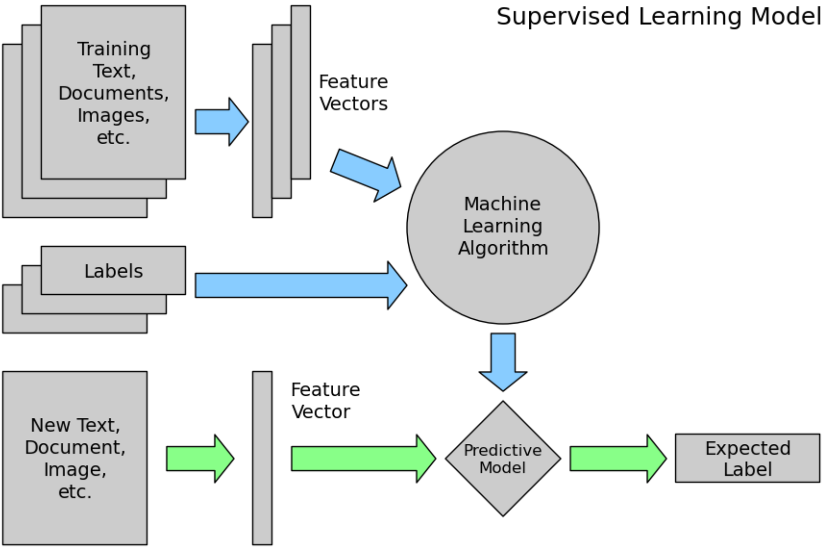
What happens in learning, then?

- the learning algorithm observes **given examples**
- it tries to find common patterns that explain the data: it tries to **generalise** so that predictions can be made for **new examples**
- exactly how this is done depends on what **algorithm we are using**

keywords here:

- given/new examples
    - **the settings of a learning experiment are important**
- generalising
- algorithm we are using

Settings



source: http://www.astroml.org/

Data sets

- training set: instances for training the system
- development set: instances for tuning the system and estimate error
- test or evaluation set: previously unseen instances on which model can be tested to asses its performance

Data sets

- training set: instances for training the system
- development set: instances for tuning the system and estimate error
- test or evaluation set: previously unseen instances on which model can be tested to asses its performance

building and tuning the model (repeatedly)

evaluating the model (just once!)

Cross-validation

what if we don't have a lot of labelled data?

a separate test-set (e.g. 20%) might be not representative and could contain particularly easy/difficult instances

Cross-validation

what if we don't have a lot of labelled data?

a separate test-set (e.g. 20%) might be not representative and could contain particularly easy/difficult instances

possible solution: cross-validation

- the whole dataset is split k times (e.g. $k = 5$)
- training/testing is repeated k times
- the whole dataset gets tested

Cross-validation

possible solution: cross-validation

- the whole dataset is split k times (e.g. $k = 5$)
- training/testing is repeated k times
- the whole dataset gets tested

Cross-validation

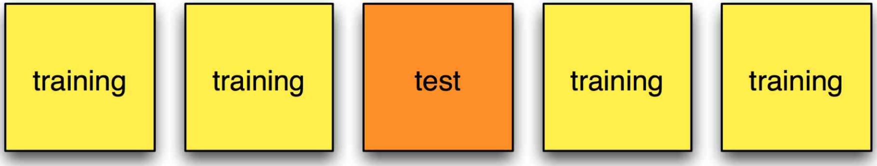
possible solution: cross-validation

- the whole dataset is split k times (e.g. $k = 5$)
- training/testing is repeated k times
- the whole dataset gets tested

Cross-validation

possible solution: cross-validation

- the whole dataset is split k times (e.g. $k = 5$)
- training/testing is repeated k times
- the whole dataset gets tested



Cross-validation

possible solution: cross-validation

- the whole dataset is split k times (e.g. $k = 5$)
- training/testing is repeated k times
- the whole dataset gets tested

Cross-validation

possible solution: cross-validation

- the whole dataset is split k times (e.g. $k = 5$)
- training/testing is repeated k times
- the whole dataset gets tested

What happens in learning, then?

- the learning algorithm observes **given examples**
- it tries to find common patterns that explain the data: it tries to **generalise** so that predictions can be made for **new examples**
- exactly how this is done depends on **what algorithm** we are using

keywords here:

- given/new examples
  - the settings of a learning experiment are important
- generalising
  - **what does it mean to generalise well?**
- algorithm we are using
  - we'll see one today (we'll see several more in this course)

Evaluation

Evaluation of results

$\rightarrow$ is the system really able to generalise?

Evaluation of results

→ is the system really able to generalise?

- the test set is equipped with class labels, manually assigned (gold standard)
- for each instance in the test set, we compare the class predicted by the classifier with the class specified in the gold standard
- how do we *measure* performance?
- when is a model good enough?

Evaluation example

Word Sense Disambiguation for the English verb ‘‘to poach"

- Some swindlers are trying to **poach** upon the rich preserves
- Firms began to **poach** partners and to recruit dozens of [. . .]
- [. . .] that will allow them to **poach** workers or markets
- [. . .] fry a teaspoonful of the pate or **poach** it in [. . .]
- [. . .] gently, and **poach** spoonfuls of meringue in this
- Let them **poach** for 3 to 4 minutes

for the resulting feature vectors, you have to predict: steal or boil?

```
trying,to,upon,the,?
began,to,partners,and,?
them,to,workers,or,?
pate,or,it,in,?
gently,and,spoonfuls,of,?
let,them,for,3,?
```

you have your gold standard:

```
trying,to,upon,the,steal
began,to,partners,and,steal
them,to,workers,or,steal
pate,or,it,in,boil
gently,and,spoonfuls,of,boil
let,them,for,3,boil
```

you compare:

gold	prediction
trying,to,upon,the,steal	trying,to,upon,the,steal
began,to,partners,and,steal	began,to,partners,and,boil
them,to,workers,or,steal	them,to,workers,or,boil
pate,or,it,in,boil	pate,or,it,in,boil
gently,and,spoonfuls,of,boil	gently,and,spoonfuls,of,steal
let,them,for,3,boil	let,them,for,3,boil

- how do we *measure* performance?
- when is a model good enough?

Evaluation measures

- accuracy: percentage of correct decisions overall

Evaluation measures

- accuracy: percentage of correct decisions overall

gold	prediction
`trying,to,upon,the,steal`	`trying,to,upon,the,steal`
`began,to,partners,and,steal`	`began,to,partners,and,boil`
`them,to,workers,or,steal`	`them,to,workers,or,boil`
`pate,or,it,in,boil`	`pate,or,it,in,boil`
`gently,and,spoonfuls,of,boil`	`gently,and,spoonfuls,of,steal`
`let,them,for,3,boil`	`let,them,for,3,boil`

accuracy = ?

Evaluation measures

- accuracy: percentage of correct decisions overall

gold	prediction
trying,to,upon,the,steal	trying,to,upon,the,steal
began,to,partners,and,steal	began,to,partners,and,boil
them,to,workers,or,steal	them,to,workers,or,boil
pate,or,it,in,boil	pate,or,it,in,boil
gently,and,spoonfuls,of,boil	gently,and,spoonfuls,of,steal
let,them,for,3,boil	let,them,for,3,boil

accuracy $= 3/6 = 50\%$

Evaluation measures

Consider class "X"

- true positive (**TP**): X classified as X
- true negative (**TN**): ¬X classified as ¬X
- false positive (**FP**): ¬X classified as X
- false negative (**FN**): X classified as ¬X

Evaluation measures

Consider class "X"

- true positive (**TP**): X classified as X
- true negative (**TN**): ¬X classified as ¬X
- false positive (**FP**): ¬X classified as X
- false negative (**FN**): X classified as ¬X

Consider class "steal"

- true positive (**TP**): steal classified as steal
- true negative (**TN**): boil classified as boil
- false positive (**FP**): boil classified as steal
- false negative (**FN**): steal classified as boil

confusion matrix

steal

prediction

gold standard	steal	boil
steal	**TP**	
boil		**TN**

confusion matrix

<u>steal</u>

prediction

gold standard		steal	boil
	steal	**TP**	**FN**
	boil	**FP**	**TN**

Evaluation measures

- **precision**$_X$:
  correct decisions over instances assigned to class "X"
  $TP/(TP + FP)$
- **recall**$_X$:
  correct assignments to class "X" over all instances of class "X" in test set
  $TP/(TP + FN)$
- **f-score**$_X$:
  combined measure of precision and recall
  $F = \frac{2PR}{P+R}$

Evaluation measures

gold	prediction
`trying,to,upon,the,steal`	`trying,to,upon,the,steal`
`began,to,partners,and,steal`	`began,to,partners,and,boil`
`them,to,workers,or,steal`	`them,to,workers,or,boil`
`pate,or,it,in,boil`	`pate,or,it,in,boil`
`gently,and,spoonfuls,of,boil`	`gently,and,spoonfuls,of,steal`
`let,them,for,3,boil`	`let,them,for,3,boil`

$\text{precision}_{steal} = ?$
$\text{recall}_{steal} = ?$

Evaluation measures

gold	prediction
trying,to,upon,the,steal	trying,to,upon,the,steal
began,to,partners,and,steal	began,to,partners,and,boil
them,to,workers,or,steal	them,to,workers,or,boil
pate,or,it,in,boil	pate,or,it,in,boil
gently,and,spoonfuls,of,boil	gently,and,spoonfuls,of,steal
let,them,for,3,boil	let,them,for,3,boil

$\text{precision}_{steal} = 1/2 = 50\%$
$\text{recall}_{steal} =$

Evaluation measures

gold	prediction
`trying,to,upon,the,steal`	`trying,to,upon,the,steal`
`began,to,partners,and,steal`	`began,to,partners,and,boil`
`them,to,workers,or,steal`	`them,to,workers,or,boil`
`pate,or,it,in,boil`	`pate,or,it,in,boil`
`gently,and,spoonfuls,of,boil`	`gently,and,spoonfuls,of,steal`
`let,them,for,3,boil`	`let,them,for,3,boil`

$\text{precision}_{steal} = 1/2 = 50\%$
$\text{recall}_{steal} = 1/3 = 33\%$

Evaluation measures

what is good enough?

- upperbound: inter-annotator agreement
- baseline: performance of basic, simple model
  for example: **assignment of most frequent class in data set**
  - sense_1 9/10 and sense_2 1/10
  - sense_1 6/10 and sense_2 4/10

What happens in learning, then?

- the learning algorithm observes **given examples**
- it tries to find common patterns that explain the data: it tries to **generalise** so that predictions can be made for **new examples**
- exactly how this is done depends on what **algorithm we are using**

keywords here:

- given/new examples
  - the settings of a learning experiment are important
- generalising
  - what does it mean to generalise well?
- algorithm we are using
  - **we are going to see one now**

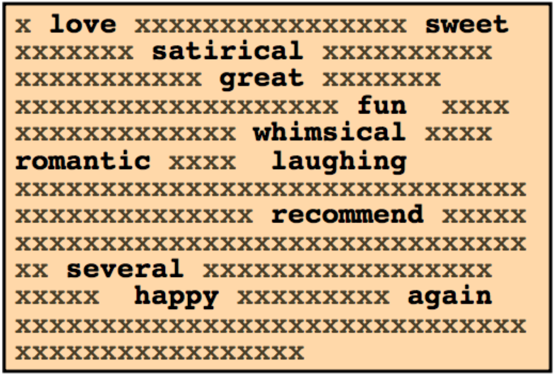
Naive Bayes

Naive Bayes classification

- simple classification method based on Bayes rule
- relies on a simple representation of documents: bag of words

Naive Bayes classification

- simple classification method based on Bayes rule
- relies on a simple representation of documents: bag of words



Naive Bayes classification

- simple classification method based on Bayes rule
- relies on a simple representation of documents: bag of words

great	2
love	2
recommend	1
laugh	1
happy	1
...	...

Conditional probability

- conditional probability of an event: a probability obtained with the **additional information** that **some other event** has already occurred
- new information is used to revise the probability of the initial event
- **prior** vs **posterior probability**
  - prior: probability obtained "as things stand", before any additional information is acquired
  - posterior: probability value which has been revised by using additional information

Example

in a corpus: happy tweets = 45% ; sad tweets = 55%

- I select one instance, how probable is it to be happy?
- The tweet contains the word “cheerful”.
  “cheerful” occurs in 65% of happy tweets,
  and in 20% of sad tweets.
  Does the probability now change?
- which one is prior and which one posterior?

Example

in a corpus: happy tweets = 45% ; sad tweets = 55%

- I select one instance, how probable is it to be happy?
- The tweet contains the word “cheerful”.
  “cheerful” occurs in 65% of happy tweets,
  and in 20% of sad tweets.
  Does the probability now change?
- which one is prior and which one posterior?

Example

in a corpus: happy tweets = 45% ; sad tweets = 55%

- I select one instance, how probable is it to be happy?
- The tweet contains the word "cheerful".
  "cheerful" occurs in 65% of happy tweets,
  and in 20% of sad tweets.
  Does the probability now change?
- which one is prior and which one posterior?

Bayes' Theorem

$$p(c_j|i) = \frac{p(i|c_j)p(c_j)}{[p(c_j)p(i|c_j)] + [p(\neg c_j)p(i|\neg c_j)]}$$

- c_j: a given class (happy)
- i: a given instance ("I always feel cheerful on Friday evening")

Bayes' Theorem

$$p(c_j|i) = \frac{p(i|c_j)p(c_j)}{[p(c_j)p(i|c_j)] + [p(\neg c_j)p(i|\neg c_j)]}$$

- c_j: a given class (happy)
- i: a given instance ("I always feel cheerful on Friday evening")

- $p(c_j|i)$ = pr of instance i being in class c_j
  how likely is it this given tweet from the corpus is happy?

Bayes' Theorem

$$p(c_j|i) = \frac{p(i|c_j)p(c_j)}{[p(c_j)p(i|c_j)] + [p(\neg c_j)p(i|\neg c_j)]}$$

- c_j: a given class (happy)
- i: a given instance ("I always feel cheerful on Friday evening")

- $p(c_j|i)$ = pr of instance i being in class c_j
  how likely is it this given tweet from the corpus is happy?
- $p(i|c_j)$ = (*likelihood function*) = pr of generating instance i given class c_j (happy)
  given class c_j (happy) how likely is it to get i?
  TRUE POSITIVE: 65%

Bayes' Theorem

$$p(c_j|i) = \frac{p(i|c_j)p(c_j)}{[p(c_j)p(i|c_j)] + [p(\neg c_j)p(i|\neg c_j)]}$$

- c_j: a given class (happy)
- i: a given instance ("I always feel cheerful on Friday evening")

- $p(c_j|i)$ = pr of instance i being in class c_j
  how likely is it this given tweet from the corpus is happy?
- $p(i|c_j)$ = (*likelihood function*) = pr of generating instance i given class c_j (happy)
  given class c_j (happy) how likely is it to get i?
  TRUE POSITIVE: 65%
- $p(i|\neg c_j)$ = pr of generating instance i given $\neg c_j$ (sad)
  given $\neg c_j$ (sad) how likely is it to get i?
  FALSE POSITIVE: 20%

Worked Example

$$p(c_j|i) = \frac{p(i|c_j)p(c_j)}{[p(c_j)p(i|c_j)] + [p(\neg c_j)p(i|\neg c_j)]}$$

- c_j: happy
- i: "I always feel cheerful on Friday evening" (represented by feature value "cheerful")
- happy = 45%
- said = 55%
- "cheerful" in 65% of happy tweets
- "cheerful" in 20% of sad tweets

Worked Example

$$p(c_j|i) = \frac{p(i|c_j)p(c_j)}{[p(c_j)p(i|c_j)] + [p(\neg c_j)p(i|\neg c_j)]}$$

- c_j: happy
- i: "I always feel cheerful on Friday evening" (represented by feature value "cheerful")
- happy = 45%
- said = 55%
- "cheerful" in 65% of happy tweets
- "cheerful" in 20% of sad tweets

Worked Example

$$p(c_j|i) = \frac{p(i|c_j)p(c_j)}{[p(c_j)p(i|c_j)] + [p(\neg c_j)p(i|\neg c_j)]}$$

- c_j: happy
- i: "I always feel cheerful on Friday evening"
  (represented by feature value "cheerful")
- happy = 45%
- said = 55%
- "cheerful" in 65% of happy tweets
- "cheerful" in 20% of sad tweets
- $p(c_j) =$

Worked Example

$$p(c_j|i) = \frac{p(i|c_j) \cdot 0.45}{[0.45 \cdot p(i|c_j)] + [p(\neg c_j)p(i|\neg c_j)]}$$

- c_j: happy
- i: "I always feel cheerful on Friday evening" (represented by feature value "cheerful")
- happy = 45%
- said = 55%
- "cheerful" in 65% of happy tweets
- "cheerful" in 20% of sad tweets
- $p(c_j) = 0.45$

Worked Example

$$p(c_j|i) = \frac{p(i|c_j) \cdot 0.45}{[0.45 \cdot p(i|c_j)] + [p(\neg c_j)p(i|\neg c_j)]}$$

- c_j: happy
- i: "I always feel cheerful on Friday evening" (represented by feature value "cheerful")
- happy = 45%
- said = 55%
- "cheerful" in 65% of happy tweets
- "cheerful" in 20% of sad tweets
- $p(c_j) = 0.45$
- $p(\neg c_j) =$

Worked Example

$$p(c_j|i) = \frac{p(i|c_j) \cdot 0.45}{[0.45 \cdot p(i|c_j)] + [0.55 \cdot p(i|\neg c_j)]}$$

- c_j: happy
- i: "I always feel cheerful on Friday evening" (represented by feature value "cheerful")
- happy = 45%
- said = 55%
- "cheerful" in 65% of happy tweets
- "cheerful" in 20% of sad tweets
- $p(c_j) = 0.45$
- $p(\neg c_j) = 0.55$

Worked Example

$$p(c_j|i) = \frac{p(i|c_j) \cdot 0.45}{[0.45 \cdot p(i|c_j)] + [0.55 \cdot p(i|\neg c_j)]}$$

- c_j: happy
- i: "I always feel cheerful on Friday evening"
  (represented by feature value "cheerful")
- happy $= 45\%$
- said $= 55\%$
- "cheerful" in 65% of happy tweets
- "cheerful" in 20% of sad tweets
- $p(i|c_j)$ = given class c_j (happy) how likely is it to get i?

Worked Example

$$p(c_j|i) = \frac{0.65 \cdot 0.45}{[0.45 \cdot 0.65] + [0.55 \cdot p(i|\neg c_j)]}$$

- c_j: happy
- i: "I always feel cheerful on Friday evening" (represented by feature value "cheerful")
- happy = 45%
- said = 55%
- "cheerful" in 65% of happy tweets
- "cheerful" in 20% of sad tweets
- $p(i|c_j)$ = given class c_j (happy) how likely is it to get i? 65%

Worked Example

$$p(c_j|i) = \frac{0.65 \cdot 0.45}{[0.45 \cdot 0.65] + [0.55 \cdot p(i|\neg c_j)]}$$

- c_j: happy
- i: "I always feel cheerful on Friday evening" (represented by feature value "cheerful")
- happy = 45%
- said = 55%
- "cheerful" in 65% of happy tweets
- "cheerful" in 20% of sad tweets
- $p(i|c_j)$ = given class c_j (happy) how likely is it to get i? 65% TRUE POSITIVE
- $p(i|\neg c_j)$ = given class $\neg c_j$ (sad) how likely is it to get i?

Worked Example

$$p(c_j|i) = \frac{0.65 \cdot 0.45}{[0.45 \cdot 0.65] + [0.55 \cdot 0.20]}$$

- c_j: happy
- i: "I always feel cheerful on Friday evening" (represented by feature value "cheerful")
- happy = 45%
- said = 55%
- "cheerful" in 65% of happy tweets
- "cheerful" in 20% of sad tweets
- $p(i|c_j)$ = given class c_j (happy) how likely is it to get i? 65% TRUE POSITIVE
- $p(i|\neg c_j)$ = given class $\neg c_j$ (sad) how likely is it to get i? 20%

Worked Example

$$p(c_j|i) = \frac{0.65 \cdot 0.45}{[0.45 \cdot 0.65] + [0.55 \cdot 0.20]}$$

- c_j: happy
- i: "I always feel cheerful on Friday evening"
(represented by feature value "cheerful")
- happy = 45%
- said = 55%
- "cheerful" in 65% of happy tweets
- "cheerful" in 20% of sad tweets
- $p(i|c_j)$ = given class c_j (happy) how likely is it to get i? 65%
TRUE POSITIVE
- $p(i|\neg c_j)$ = given class $\neg c_j$ (sad) how likely is it to get i? 20%
FALSE POSITIVE

Worked Example

$$p(c_j|i) = \frac{0.65 \cdot 0.45}{[0.45 \cdot 0.65] + [0.55 \cdot 0.20]} = 0.73$$

- c_j: happy
- i: “I always feel cheerful on Friday evening” (represented by feature value “cheerful”)
- happy = 45%
- said = 55%
- “cheerful” in 65% of happy tweets
- “cheerful” in 20% of sad tweets
- $p(i|c_j)$ = given class c_j (happy) how likely is it to get i? 65% TRUE POSITIVE
- $p(i|\neg c_j)$ = given class $\neg c_j$ (sad) how likely is it to get i? 20% FALSE POSITIVE

Worked Example

$$p(c_j|i) = \frac{0.65 \cdot 0.45}{[0.45 \cdot 0.65] + [0.55 \cdot 0.20]} = 0.73$$

- **prior** probability of i as *happy* $= 0.45$
- **posterior** probability of i as *happy* $= 0.73$

Worked Example

$$p(c_j|i) = \frac{0.65 \cdot 0.45}{[0.45 \cdot 0.65] + [0.55 \cdot 0.20]} = 0.73$$

- **prior** probability of i as *happy* $= 0.45$
- **posterior** probability of i as *happy* $= 0.73$

$$p(c_j|i) = \frac{p(i|c_j)p(c_j)}{p(i)}$$

$p(i)$ = the probability of i ("cheerful") overall

Worked Example

$$p(c_j|i) = \frac{0.65 \cdot 0.45}{[0.45 \cdot 0.65] + [0.55 \cdot 0.20]} = 0.73$$

- **prior** probability of i as *happy* $= 0.45$
- **posterior** probability of i as *happy* $= 0.73$

$$p(c_j|i) = \frac{p(i|c_j)p(c_j)}{p(i)}$$

$p(i)$ = the probability of i ("cheerful") overall

$65\% \cdot 45\% + 20\% \cdot 55\%$

Worked Example

$$p(c_j|i) = \frac{p(i|c_j)p(c_j)}{p(i)}$$

how do we make a classifier out of this?
we need to pick a class c out of a set of possible class values.

Worked Example

$$p(c_j|i) = \frac{p(i|c_j)p(c_j)}{p(i)}$$

how do we make a classifier out of this?
we need to pick a class c out of a set of possible class values.

we add a *decision rule*: maximum a posteriori (*map*)

$$c_{map} = \arg\max_{c \in C} \frac{p(i|c) \cdot p(c)}{p(i)}$$

Worked Example

$$p(c_j|i) = \frac{p(i|c_j)p(c_j)}{p(i)}$$

how do we make a classifier out of this?
we need to pick a class c out of a set of possible class values.

we add a *decision rule*: maximum a posteriori (*map*)

$$c_{map} = \arg\max_{c \in C} \frac{p(i|c) \cdot p(c)}{p(i)}$$

note that because we only need to *compare* values, we can drop the denominator, which basically serves as normalising function

$$c_{map} = \arg\max_{c \in C} p(i|c) \cdot p(c)$$

Issues

- more features
- independence of features
- zeros and smoothing (Laplace or add-one smoothing)
- underflow

Issues

- more features $i_1, ...i_n$

$$c_{map} = \arg\max_{c \in C} p(i_1, i_2, ..., i_n | c) \cdot p(c)$$

- independence of features
- zeros and smoothing (Laplace or add-one smoothing)
- underflow

Issues

- more features
- independence of features
  features are often not independent, but we assume they are, otherwise calculations get too complicated. This is why it is called *naive*.
  Because we do, we can estimate conditional probability of each feature j separately, for a total of n features.

$$c_{map} = \arg\max_{c \in C} \prod_{j=1}^{n} p(i_j|c) \cdot p(c)$$

  (a *Bayesian Network* doesn't include the independence assumption.)
- zeros and smoothing (Laplace or add-one smoothing)
- underflow

Issues

- more features
- independence of features
- zeros and smoothing (Laplace or add-one smoothing)
  it can happen that some values are zero. To prevent this problem in the calculations, the value 1 is added to all observed counts
- underflow

Issues

- more features
- independence of features
- zeros and smoothing (Laplace or add-one smoothing)
- underflow

  posterior probabilities are usually very very small, especially with lots of features (think of a bag-of-words approach in text classification). This is called the *underflow* problem

  For this reason, most implementations of a NB classifier are like this:

$$c_{map} = \arg\max_{c \in C}[\sum_{j=1}^{n} \log p(i_j|c) + \log p(c)]$$

log(xy) = log(x) + log(y)

what we have seen is a **supervised classification** problem what we have seen is a **supervised classification** problem

- supervised vs unsupervised learning
- classification vs regression

Classification vs Regression

create models of prediction from gathered data

- classification
  the dependent variables are categorical
  - input x: feature vector
  - output: **discrete class label**

- regression
  the dependent variables are numerical
  - input x: feature vector
  - output y: **continuous value**

can you think of a problem where you would need a **regression model**?

classification and regression are the most standard ways of doing **supervised learning**

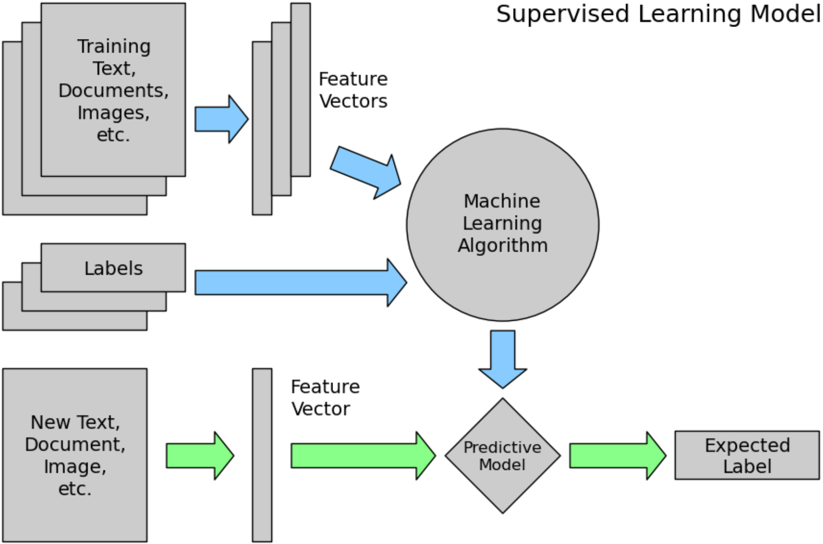
Supervised and Unsupervised Learning

information about the correct distribution/label of the training examples

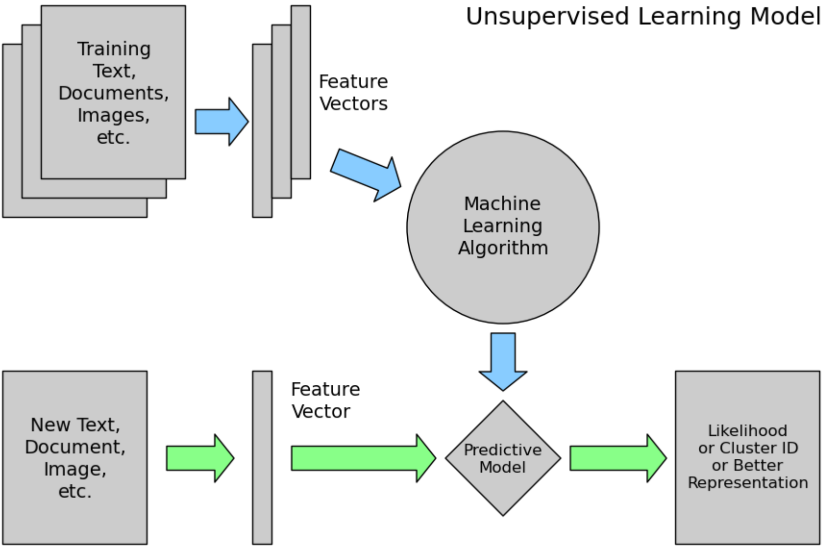
- in supervised learning it is known

  → fitting a model to labelled data which has the correct answer
- in unsupervised learning it is not known

  → finding structure in unlabelled data



source: `http://www.astroml.org/`



source: http://www.astroml.org/

Supervised learning

supervised learning – classification or regression

- in training, instances are associated with their class label
- based on features, the system must search for patterns and build a model
- the model must be able to *predict* the class of previously unseen instances

Unsupervised learning

unsupervised learning – clustering

- partitioning instances into subsets (clusters) that share similar characteristics
- subsets are not predefined
- a system can be told *how many* clusters it should form (K-means algorithm)

Semi-supervised learning

Semi-supervised learning

using a small set of labelled data + a large set of unlabelled data

intuitively:

- it should be better than just using a small set of labelled data
- it should be better than just using a large amount of unlabelled data

in practice

In practice

- Python3 up and running
- install scikit-learn
  http://scikit-learn.org/stable/install.html
- Labs!
- Assignment1