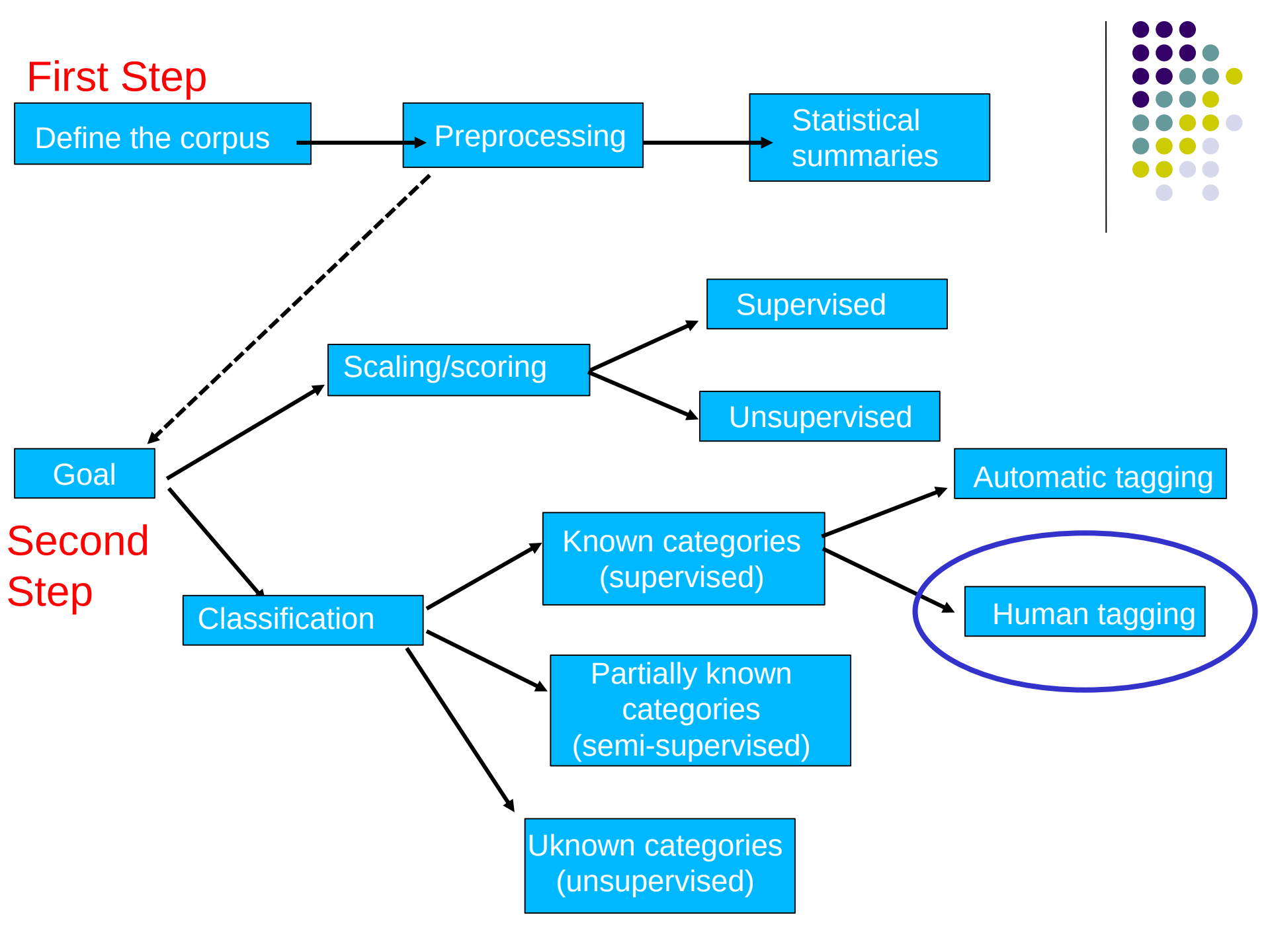
Big Data Analytics

Lecture 3/C

Supervised classification methods (part 1)

UNIVERSITÄT LUZERN


First Step
Define the corpus
Preprocessing
Statistical summaries
Supervised
Scaling/scoring
Unsupervised
Goal
Automatic tagging
Second Step
Known categories (supervised)
Classification
Human tagging
Partially known categories (semi-supervised)
Uknown categories (unsupervised)

References

- ✓ Olivella, Santiago, and Shoub Kelsey (2020). Machine Learning in Political Science: Supervised Learning Models. In Luigi Curini and Robert Franzese (eds.), *SAGE Handbook of Research Methods is Political Science & International Relations*, London, Sage, chapter 56

ML alghoritms

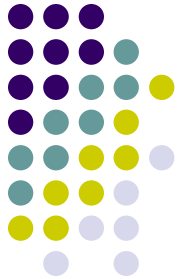
Several different possible machine learning algorithms are available out there

We will offer an intuitive introduction to some of them (pretty standard in text analytics)

Unfortunately we will have no time to discuss about many other of them. However, the general logic they employ remains always the same…

We will begin with *Naïve Bayes classifiers*

Naïve Bayes classifier



Naïve Bayes classifier: the algorithm allows us to predict a class, given a set of features using (Bayes) probability theorem

But what do we mean by Bayes probability theorem?

Bayesian probability incorporates the concept of *conditional probability*, the probability of event *A* given that event *B* has occurred, i.e., p(A|B)

If we have just two classes, we can write:

$$\mathrm{p}(A|B) = \frac{p(A)*p(B|A)}{p(A)*p(B|A)+p(\neg A)*p(B|\neg A)}$$

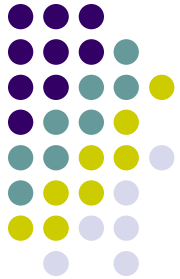
where:

$p(A|B)$=posterior probability

$p(B|A)$= conditional probability or likelihood

$p(A)$=prior probability of the outcome

Naïve Bayes classifier



An example (with no texts involved!):

You are on the train and you want to understand if the person sitting next to you is a centre-right voter

You know a priori (your *priors*!) that 54% of the Italian citizens are centre-right voters (46% centre-left)

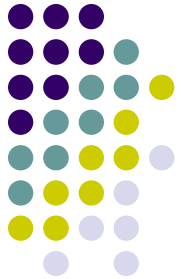
Now the person sitting next to you open a newspaper. You know that the 35% of centre-right voters read that newspaper (while it is read by 65% of centre-left voters) – your *likelihood*!

Which is your update belief that the person sitting next to you is a centre-right voter?

$p(CR|N) = p(CR)\, p(N|CR)/[(p(CR)\, p(N|CR)+p(CL)\, p(N|CL))] =$

$=.54*.35/(.54*.35+.46*.65) = .387$

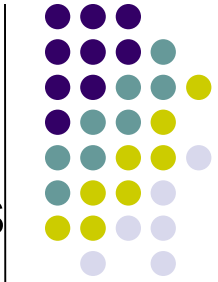
Naïve Bayes classifier



Within a text-analytics framework, the goal (as already discussed!) is to infer the probability that document *i* belongs to category *k* given word profile $\mathbf{W}_i$ (i.e., the probability of a text belonging to category *k* given that its predictors – features – values are $x_1,x_2,\ldots,x_p$

This can be written as $p(C_k|x_1, x_2,\ldots,x_p)$)

Naïve Bayes classifier



More formally, the Bayesian formula for calculating this probability is:

$$\mathrm{p}(C_k|\boldsymbol{W}_i) = \frac{p(C_k)*p(\boldsymbol{W}_i|C_k)}{p(\boldsymbol{W}_i)}$$

In plain English:

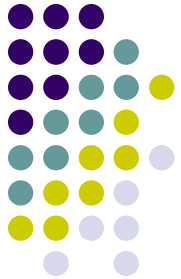
$p(C_k|\boldsymbol{W}_i)$=posterior probability: by combining our observed information, we are updating our *a priori* information on probabilities to compute a posterior probability that an observation has class C_k

$p(\boldsymbol{W}_i|C_k)$= conditional probability or likelihood

$p(C_k)$=prior probability of the outcome (i.e., the average probabily of that outcome in the training-set)

$p(\boldsymbol{W}_i)$=evidence (the word profile we observe)

Naïve Bayes classifier



In other words, the Bayesian formula is simply:

$$\text{Posterior} = \frac{prior * likelihood}{evidence}$$

From above, we can drop $p(\boldsymbol{W}_i)$ - i.e., the evidence - from the denominator since it is a constant across the different categories for each document

Naïve Bayes classifier

An example: let's say we have a training-set on 1000 pieces of fruit. The fruit being a Banana, Orange or some Other fruit. We know **3 variables of each fruit**, whether it's long or not, sweet or not and yellow or not:

Fruit	Long	Sweet	Yellow	Total
Banana	400	350	450	500
Orange	0	150	300	300
Other	100	150	50	200
Total	**500**	**650**	**800**	**1000**

Naïve Bayes classifier

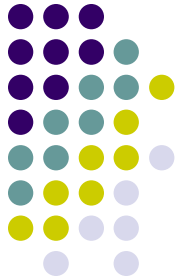
Fruit	Long	Sweet	Yellow	Total
Banana	400	350	450	500
Orange	0	150	300	300
Other	100	150	50	200
Total	**500**	**650**	**800**	**1000**

From the table we know that, in our training-set: 50% of the fruits are bananas; 30% are oranges; 20% are other fruits (our *priors*!)

Based on our table, we can *also* say the following:

Out of 500 bananas 400 (0.8) are Long, 350 (0.7) are Sweet and 450 (0.9) are Yellow; Out of 300 oranges 0 are Long (0.0), 150 (0.5) are Sweet and 300 (1.0) are Yellow; From the remaining 200 fruits, 100 (0.5) are Long, 150 (0.75) are Sweet and 50 (0.25) are Yellow. All these values refer to c*onditional probabilities* or *likelihood*!

Naïve Bayes classifier

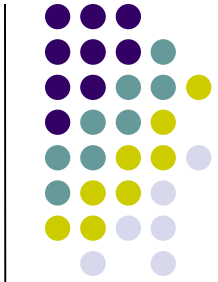


Now let's say we're given the variables of a piece of fruit and we need to predict the class (*out-of-sample prediction*!)

If we're told that the **additional fruit is Long, Sweet and Yellow**, we can classify it using the Bayes formula and subbing in the values for each outcome, whether it's a Banana, an Orange or Other Fruit

The one with the highest probability (score) being the **winner class**!

Naïve Bayes classifier



p(Banana|Long, Sweet, Yellow)=p(Banana=0.5)*p(Long|Banana=0.8)*p(Sweet|Banana=0.7)*p(Yellow|Banana=0.9)=**0.252**

p(Orange|Long, Sweet, Yellow)=p(Orange=0.3)*p(Long|Orange=0)*p(Sweet|Orange=0.5)*p(Yellow|Orange=1)=**0**

p(Other|Long, Sweet, Yellow)=p(Other=0.2)*p(Long|Other=0.5)*p(Sweet|Other=0.75)*p(Yellow|Other=0.25)=**0.01875**

In this case, based on the **highest score**, we can classify this Long, Sweet and Yellow fruit as a Banana

Naïve Bayes classifier

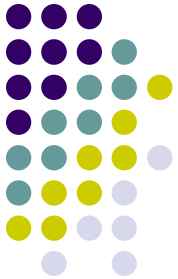
In the case of text-classification, instead of *Banana, Orange and Other* you will have some specific categories (say, *Positive, Negative and Neutral* if we are applying a sentiment analysis) and instead of *Long, Sweet and Yellow* we will have the features (aka: words) included in our DfM

The logic however followed in predicting the category of each text included in the test-set remains exactly the same as the one we just discussed!

Given the features included in that (unread) text, we will estimate its posterior probability to belong to each given category…

…then we will assign that (unread) text to the category with the largest posterior probability among the one we estimated!

Naïve Bayes classifier

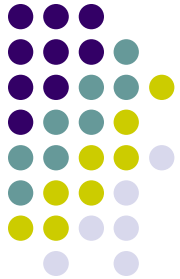


Note a possible problem of the logic just explained: given that naïve Bayes uses the product of variable probabilities conditioned on each class, we run into a serious problem when new data includes a variable value that **never occurs** for one or more levels of a response class

This is what happens with $p(Long|Orange)$ =0 . This “0” will ripple through the entire multiplication of all variable and will always force the posterior probability to be zero for that class

This is clear a HUGE PROBLEM when dealing with a sparse DfM (as it is usually the case)!

Naïve Bayes classifier

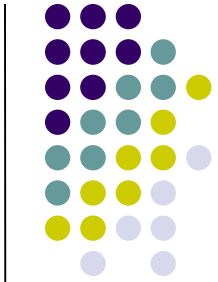


A solution to this problem involves using the ***Laplace smoother***

The Laplace smoother adds a small number to each of the counts in the frequencies for each feature, which ensures that each feature has a nonzero probability of occurring for each class

Typically, a value of 1 for the Laplace smoother is employed, but this is a *tuning parameter* to incorporate and optimize with cross validation (more on this later on!)

Naïve Bayes classifier



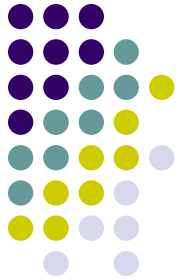
Why Naïve?

Cause it assumes that every feature being classified is **independent** of the value of any other variable given the response variable

In the previous example each of the three variables (Long, Sweet, Yellow) are considered to *contribute independently* to the probability that the fruit is a Banana, *regardless of any correlations* between features

By making this assumption we can simplify our calculation such that the posterior probability is simply the product of the probability distribution for each individual variable conditioned on the response category

Naïve Bayes classifier

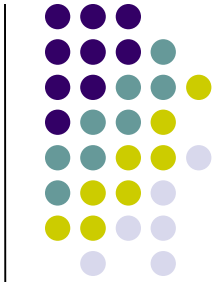


Variables, however, aren't always independent!

Although the model is *clearly wrong* – quite often features are not conditionally independent - it has proven to be a useful classifier for a diverse set of tasks

The same is true for text analysis: assuming that features (i.e., words) are generated indipendently is wrong given that the use of words is highly correlated in any data set. However, Naïve Bayes classifier can still be useful (remember the First Principle of text-analysis!)

Naïve Bayes classifier



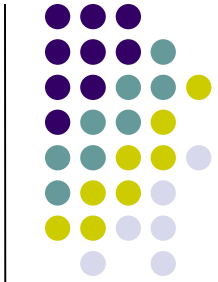
Naïve Bayes classifier algorithm has a simple, but powerful (and fast!), approach to learning the relationship between words and categories

Moreover, it has been shown to perform surprisingly well with very small amounts of training data that most other classifiers, would find significantly insufficient

As a result, if you find yourself with a small amount of training data Naïve Bayes would be a good bet!

Likewise, Naïve Bayes' simplicity prevents it from fitting its training data too closely and therefore does not tend toward **overfitting** especially on smaller datasets like other approaches do

Naïve Bayes classifier



However, Naïve Bayes classifier also behaves differently on the other end of the spectrum, when provided with large amounts of training data

As it is fed increasing quantities of training data, the performance of the Naïve Bayes classifier plateaus above a certain threshold

Its simplicity prevents it from benefiting incrementally from training data past a certain point