BTS | Barcelona Technology School

Classical Data Analysis

Master in Big Data Solutions 2020-2021

Filipa Peleja

Víctor Pajuelo

filipa.peleja@bts.tech

victor.pajuelo@bts.tech

Today's class

Contents

Let's git things done!

Let's see it again

Pull Session 10 notebooks

```
$ git clone https://github.com/vfp1/bts-cda-2020.git

# If you have done that already
$ git pull origin master
```

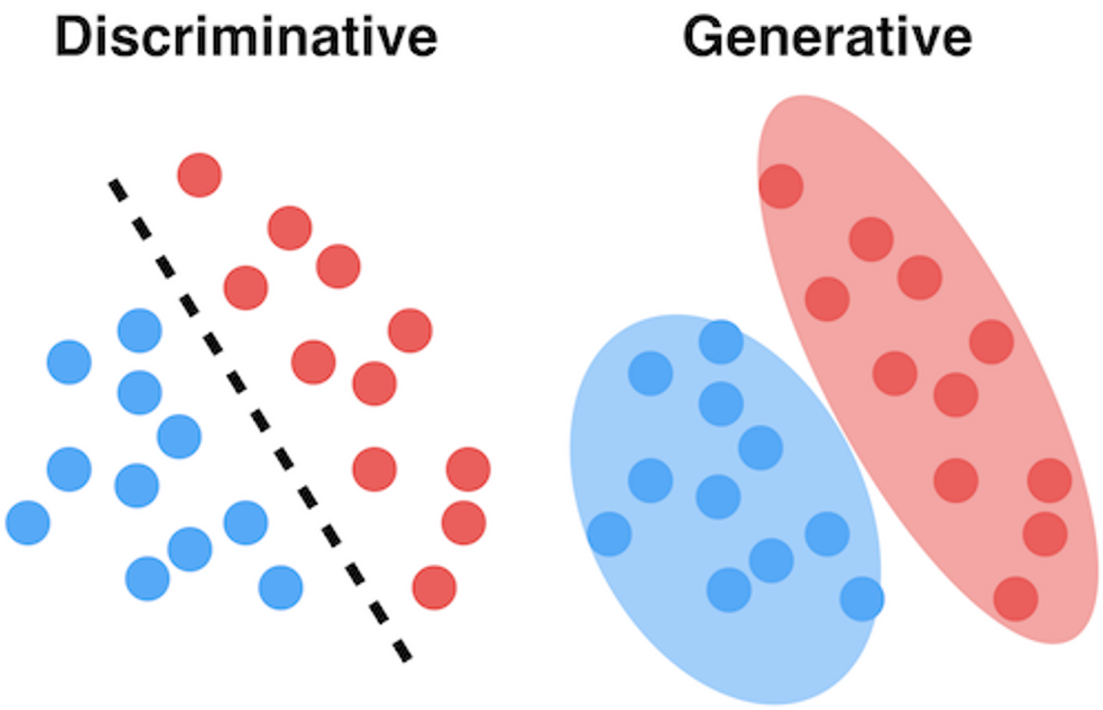
BTS | Barcelona Technology School

MBDS 2020-2021

Generative Models

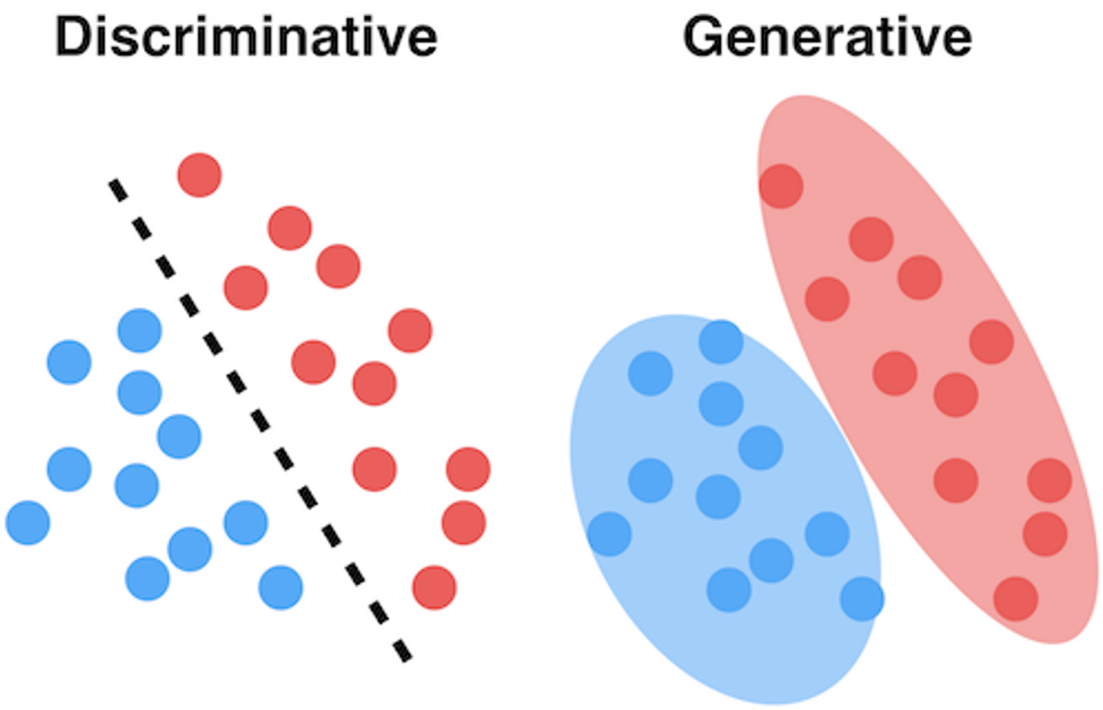
Generative models

Generative models focus on the distributions that follow the classes and try to learn how to "generate" new examples.



Generative models

- Discriminative models focus on the probability of the class, given the features: **P(y | x)**
- Generative models focus on the probability of the features given the class: **P(x | y)**



Conditional probability

- Conditional probability is a mathematical expression of the change in uncertainty due to new information.
  - What is the probability that it rained? **P(rain)**
  - What is the probability that it rained given that the streets are all wet? **P(rain | street wet)**
  - And it is not symmetric!
  - What is the probability that the streets are wet given it rained? **P(street wet | rain)**

Conditional probability

- The **joint probability** expresses the probability that two events happen at the same time:
  - What is the probability that the next car I see is a blue seat?
  - **P(seat, blue) = P(seat)P(blue | seat)**
- Remark: If variables A and B are independent, then

  P(A, B) = P(A) P(B) iff P(A | B) = P(A)

How does it work?

- Let's look at the weather dataset and corresponding target variable 'Play'. Now, we need to classify whether players will play or not based on weather condition

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

How does it work?

Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction

Players will play if weather is sunny. Is this statement is correct?

Use the method of posterior probability,

$$P(Yes|Sunny) = P(Sunny|Yes) * \frac{P(Yes)}{P(Sunny)}$$

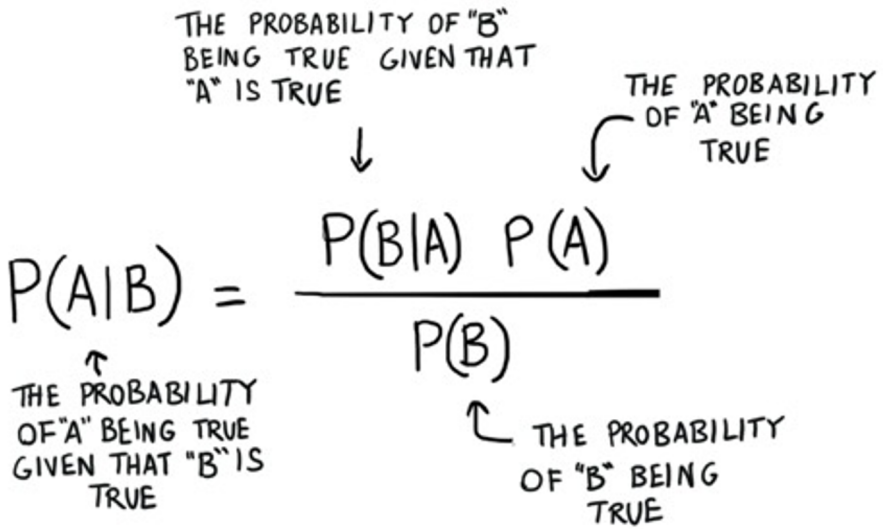
where $P(Sunny|Yes) = \frac{3}{9} = 0.33$, $P(Sunny) = \frac{5}{14} = 0.36$, $P(Yes) = \frac{9}{14} = 0.64$

Therefore, $P(Yes|Sunny) = \frac{0.33*0.64}{0.36} = 0.6$ which has higher probability

Naive Bayes uses a similar method to predict the probability of different class based on various attributes

Bayes Rule

The Bayes Rule allows us to "invert" the order of the conditional probabilities:

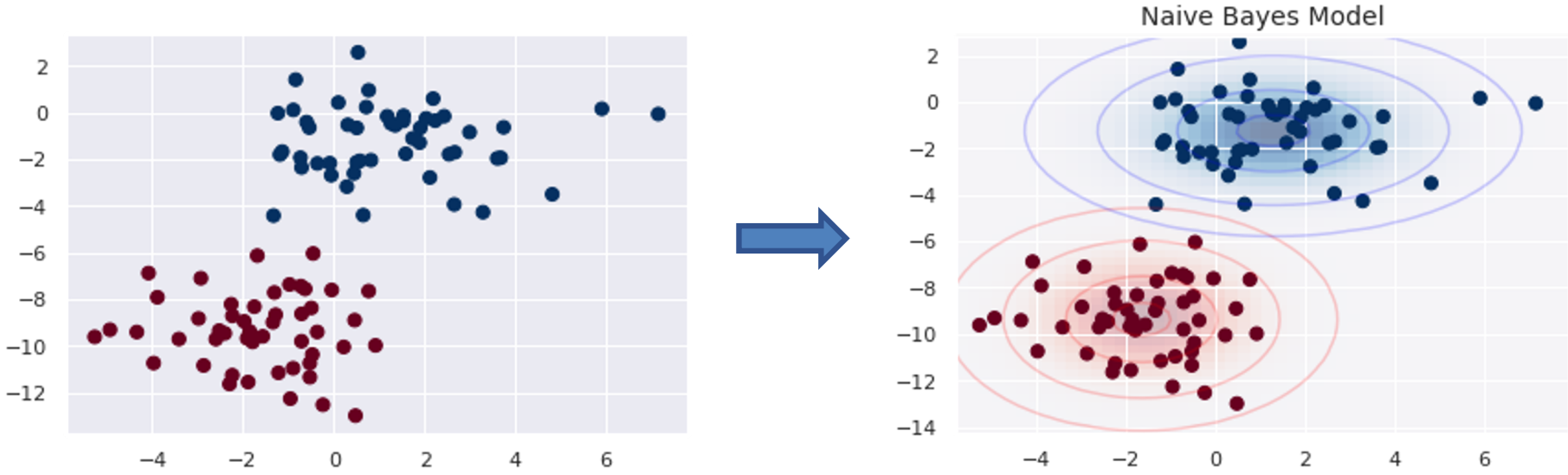


This is very useful, because sometimes it is easier to compute the conditional probability in the "other" direction. It is also very helpful for medical tests.

Gaussian Bayes Classifier

- What if the variables that we want to use are continuous?

- One option might be to discretize them and use Multinomial Bayes Classifier.
- Another option is to use a Bayes model that can treat continuous variables:



Gaussian Bayes Classifier

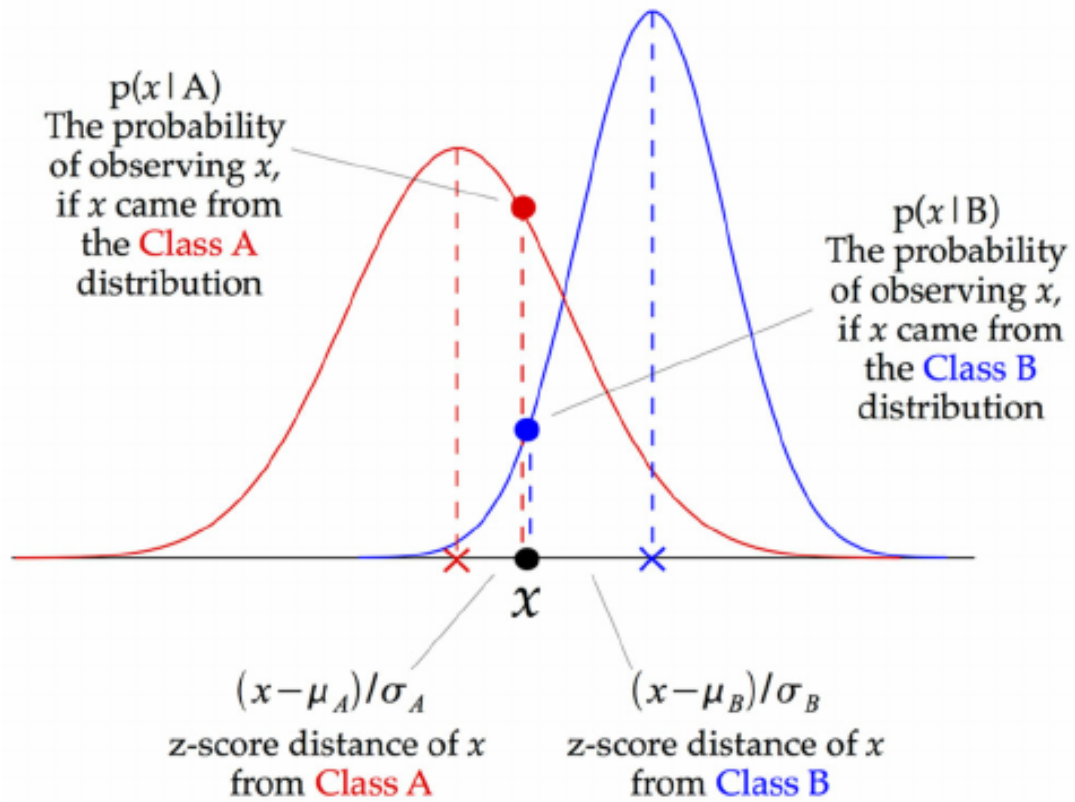
- Another option is to use a Bayes model that can treat continuous variables

$$p(X_i = x|Y = y_k) = \frac{1}{\sqrt{2\pi\sigma_{ik}^2}} \; e^{-\frac{1}{2}(\frac{x-\mu_{ik}}{\sigma_{ik}})^2}$$

- In this case we will have to estimate the mean and the standard deviation for each class and feature. Note we are also assuming independence!

Gaussian Naïve Bayes

- Assume features follow a normal distribution. Instead of discrete counts, we have continuous features (e.g., the popular Iris dataset where the features are sepal width, petal width, sepal length, petal length)



https://heartbeat.fritz.ai/understanding-the-mathematics-behind-naive-bayes-ab6ee85f50d0

More on Naïve Bayes

- **Multinomial Naive Bayes**
  - The multinomial naive Bayes model is typically used for discrete counts. E.g., if we have a text classification problem, we can take the idea of Bernoulli trials one step further and instead of "word occurs in the document" we have "count how often word occurs in the document", you can think of it as "number of times outcome number x_i is observed over the n trials"
- **Multi-variate Bernoulli Naive Bayes**
  - The binomial model is useful if your feature vectors are binary (i.e., 0s and 1s). One application would be text classification with a bag of words model where the 0s 1s are "word occurs in the document" and "word does not occur in the document"

https://www.reddit.com/r/MachineLearning/comments/2uhhbh/difference_between_binomial_multinomial_and/co8xxls/
https://towardsdatascience.com/comparing-a-variety-of-naive-bayes-classification-algorithms-fc5fa298379e

Text Classification

Possible use cases

Spam vs Not spam classification

Automated sorting – topic classification

Classify products: Toys, electronics, furniture, …

Sentiment Classification

Text Classification

Subscribe to our newsletter and be the first to know our latest offer and discounts. Click to subscribe. <- SPAM?

- In order to process this, the first thing we need to do is to define a **feature vector**

- **Vocabulary:** Set of "all possible" words

  (in practice most frequent words: 2000-3000)

- **Bag of words:** Representation of a document by the words that appear in it (but it does not take the order into account!)
  - When using the bag of words representation, we assume that the appearance of each word is independent.

Text Classification

- Imagine that we have a dataset with m documents (emails). Which are classified between spam and not spam. Training the **Bernuille Bayes classifier** consists in computing the probabilities P(x1 | y) P(x2 | y) … P(x_n | y) and prior probability P(y).

- P(y) = # spam email / # emails

- P(xi | y=1) = # emails with word xi / # spam emails

- P(xi | y=0) = # emails with word xi / # non-spam emails

- Prediction for a new email is done by selecting the highest probability between:

$$P(y=1 \mid x) , P(y=0 \mid x)$$

- This is done following the bayes rule.

Text Classification

- The bag of words representation has a variants which are a bit more sophisticated:
- **Term frequency:** Number of times a given word appears in a document.
- **Inverse document frequency:** $$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$
- **TF-IDF:** Term frequency normalized by IDF.
- In this case, we would use a **Multinomial naive bayes classifier**
- This model will take into account the probability of each word appearing multiple times. We are still considering that the number of times each word appears is independent! (which we know is not true).

Recap Naïve Bayes

- Even though the naive assumption is rarely true, the algorithm performs surprisingly good in many cases
- Handles high dimensional data well. Easy to parallelize and handles big data well
- Performs better than more complicated models when the data set is small

---

- The estimated probability is often inaccurate because of the naive assumption. Not ideal for regression use or probability estimation
- When data is abundant, other more complicated models tend to outperform Naive Bayes

Naïve Bayes

In class exercises

Naïve Bayes

In class exercises

Go to the notebook

BTS | Barcelona Technology School

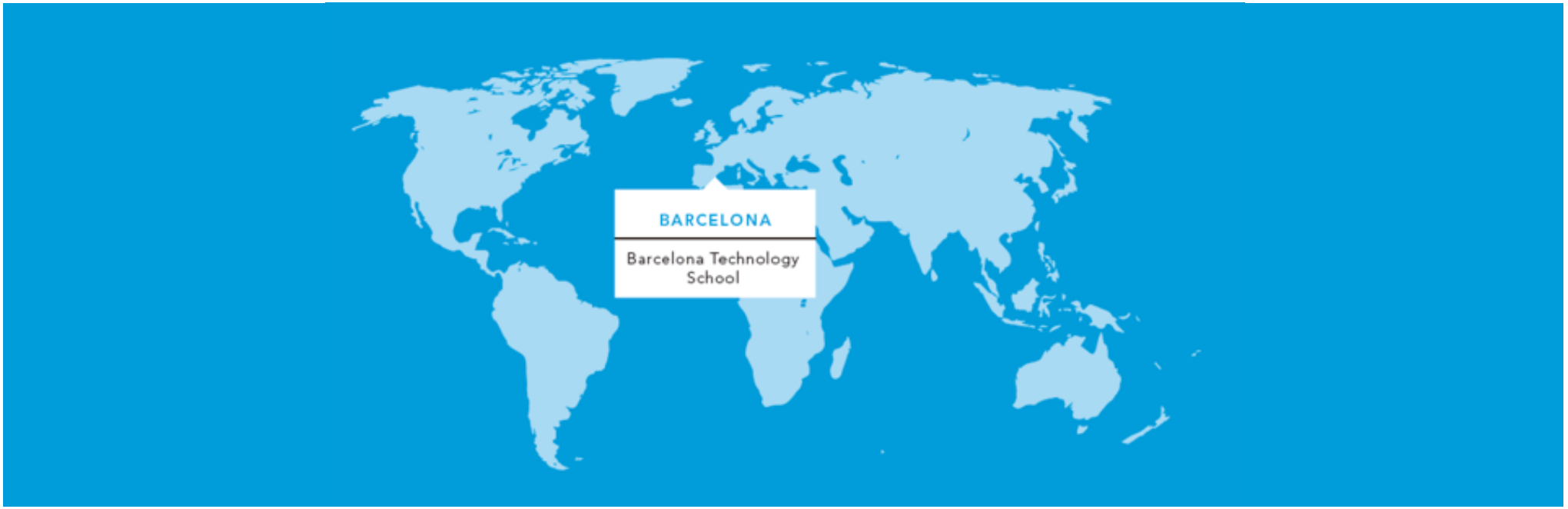
MBDS 2020-2021

RECAP

Resources

Important resources

- Aurelien Geron's Machine Learning book
- Tan, Pang-Ning. *Introduction to data mining*. Pearson Education India, 2006.
- Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. New York: Springer series in statistics, 2001.
- Andrew Ng lectures: https://www.youtube.com/playlist?list=PLLssT5z_DsK-h9vYZkQkYNWcItqhlRJLN


BARCELONA
Barcelona Technology School