Instituto Superior Técnico

Departamento de Engenharia Electrotécnica e de Computadores

Machine Learning

4^{th} Lab Assignment

Shift 5ª 17h Group number 2

Number 81138 Name João Ramiro

Number 81567 Name José Pedro Boavida Miragaia

Naive Bayes classifiers

1 Naive Bayes Classifier

Naive Bayes classifiers normally are rather simple, and are very effective in many practical situations. Describe in your own words how the Bayes classifier works. Be precise. Use equations when appropriate.

O Bayes classifier é o classificador ideal que escolhe uma classe observando qual das classes, tem é mais provável ocorrer para certos parâmetros, ou seja tem conhecimento à posteriori pois sabe a distribuição probabilística das classes em antemão (MAP). Este classificador baseia-se no teorema de Bayes para definir qual a classe e este é dado pela seguinte equação:

$$P(\omega_i|X) = \frac{P(X|\omega_i)P(\omega_i)}{P(X)}$$

Onde $P(\omega_i|X)$ corresponde à distribuição à posteriori das classes. É importante referir que no Bayes Classifier o P(X) é apenas um fator de escala pois é comum para todos os $P(\omega_i|X)$ e por isso não necessita de ser calculado (a menos que se queiram comparar vários vetores de features). O Naive Bayes Classifier é em tudo igual ao Bayes Classifier mas com uma diferença importante, neste classificador assume-se que as features são independentes umas das outras, ou seja a covariância=0. Isto permite o uso da seguinte equação:

$$P(X|\omega_i) = P(X_1|\omega_i) * P(X_2|\omega_i) * P(X_3|\omega_i) \ldots$$

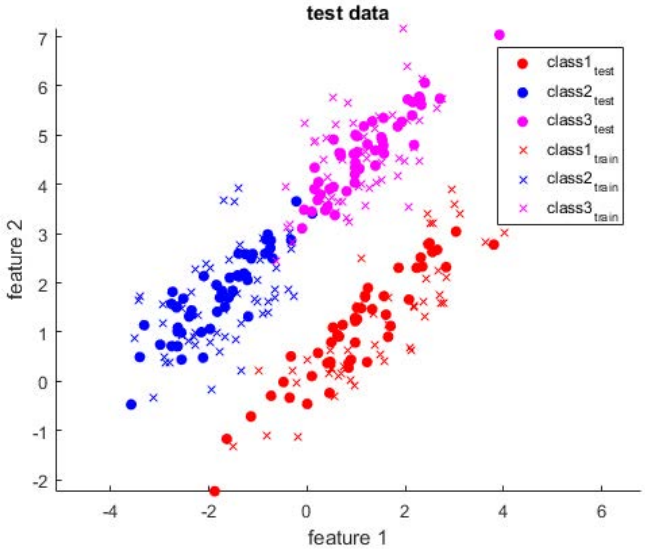
Apartir desta hipótese, seria de esperar que o Naive Bayes Classifier não obtivesse bons resultados no entanto não é isso que ocorre como se pode observar ao longo deste laboratório.

2 A simple example

In this part of the assignment, you'll make a naive Bayes classifier for a very simple set of data. The input data are two-dimensional, and belong to one of three classes. To load the data, load the file `data1.mat`. The data have already been split into training data (variables `xtrain` and `ytrain`) and test data (variables `xtest` and `ytest`).

1. Visualize a scatter plot of the training and test data, using different colors, or symbols, for the different classes. Don't forget to use equal scales for both axes, so that the

scatter plot is not distorted. Draw a sketch of the scatter plot of the training data.



2. Make a Matlab script that creates a naive Bayes classifier based on the training data, and that finds the classifications that that classifier gives to the test data. The script should plot the classifications of the test data as a function of the test pattern index, and should print the percentage of errors that the classifier makes on the test data. Write your own code, do not use any Matlab ready made function for Naive Bayes classfication.

   Give the listing of your script in a separate file. The script should have enough comments to allow the reader to understand how it works (normally, this will correspond to less than one comment per line). You don't need to make a very general script: you can make as simple a script as you wish, as long as it does what is requested.

   Suggestion: You will need to estimate probability densities of certain sets of data, to solve this item. For the estimation of each density, use a Gaussian distribution. Estimate its mean and variance, respectively, as the mean and variance of the corresponding set of data (for the variance estimates, divide by N, and not by $N-1$, where N is the number of data points). The estimator that you'll obtain in this way is the maximum-likelihood estimator of the Gaussian distribution.

3. Indicate the percentage of errors that you obtained in the test set.

   Test set error rate: 4.6667%

4. Comment on your results.

ESCREVER CENAS

The classification problem that you have just solved is very small, and was specially prepared to illustrate the basic working of naive Bayes classifiers. You should be aware,

however, that the real-life situations in which these classifiers are normally most useful are rather different from this one: they are situations in which the data to be classified have a large number of features and each feature gives some information on which is the correct class. Normally, for each individual feature, there is a significant probability of giving a wrong indication. However, with a large number of features, the probability of many of them being simultaneously wrong is very low, and, because of that, the naive Bayes classifier gives a reliable classification. The second part of this assignment addresses such a situation.

3 Language recognizer

One of the applications in which naive Bayes classifiers give good results and are relatively simple to implement, is language recognition. In the second part of this assignment, you will make some of the code of a naive Bayes language recognizer, and you will then test the recognizer. The training data are provided to you. Most of the code of the recognizer is also provided, but the parts that specifically concern the classifier's computations are missing. You will be asked to provide them. After that, you will be asked to test the recognizer.

3.1 Software and data

The Matlab code for the recognizer is given in the file `languagerecognizer.m`. This code is incomplete, and should be completed by you as indicated ahead. The code consists of two parts, which are clearly identified by comments:

- The *first part* reads the trigram counts of the training data for the various languages, from files that are supplied. The names of these files are of the form `xx_trigram_count_filtered.tsv`, where `xx` is a two-character code identifying the language that the file refers to (`pt` for Portuguese, `es` for Spanish, `fr` for French, and `en` for English).

  The aforementioned files contain the data of one trigram per line: each line contains the trigram, followed by the number of times that that trigram occurred in the corresponding language's training data. Before counting the trigrams in the training data, all upper case characters were converted to lower case. The set of characters that was considered was {abcdefghijklmnopqrstuvwxyzáéíóúàèìòùâêîôûäëïöüãõñ .,:;!¡?¿-'} (note that there is a blank character in the set). Trigrams containing characters outside that set were discarded. You may want to look into the trigram count files to have an idea of what are their contents, or to check the numbers of occurrences of some specific trigrams.

  After executing the *first part* of the code, the following variables are available:

  - `languages`: Cell array that stores the two-character codes for the languages. For

example, `languages{4}` contains the string `'en'`. Note that the argument is between braces, not parentheses.

- `total_counts`: Array that contains the total number of trigrams that occurred in the training data, for each language. For example, `total_counts(4)` contains the total number of trigrams that occurred in the training data for English. Trigrams that occurred repeatedly are counted multiple times.

The *first part* of the code is complete: you shouldn't add any code to it.

- The *second part* of the code consists, basically, of a loop that repeatedly asks for a line of input text and then classifies it. Each iteration of the loop performs the following operations:

  - Ask for a line of input text and read it.
  - Check whether the input text contains only the word `quit`. If so, exit the loop (this will end the program).
  - Convert all the input text to lower case.
  - Perform a loop on the languages. Within this loop, perform a loop on all the trigrams of the input text.
  - Print the scores of the various languages, the recognized language and the classification margin.

  This description of the operations performed by the *second part* of the code may sound somewhat incomplete, because this part of the code actually is incomplete. You should complete it by adding code, as described below.

The places where you may need to add code are clearly marked, with comments, in the file `languagerecognizer.m`. Those places are identified, in the comments, as Code Sections 1, 2 and 3. You will need to use those identifications later on.

The code that is provided already contains all the loops that are needed, as well as a few more commands. You will need to add the code that performs the calculations for the recognizer itself, using the data produced by the *first part* of the program (described above), as well as some data that are computed by already existing code of the *second part* of the program. Take into account the following indications:

- The basic structure of the *second part* is as follows:

  - There is an outermost loop, which repeatedly asks for input text and then proceeds to classify it.
  - That loop contains a loop on the languages.

– The loop on the languages contains a loop on all the trigrams of the input text. In the beginning of this loop, the trigram that is to be processed in the current iteration is placed in the variable `trigram`, and the number of occurrences of that trigram in the training data for the current language is placed in the variable `trigramcount`.

- In Code Section 3, the final results of the calculations that you perform should be placed in an array called `scores`, of size 4, with an element for each language. For example, `scores(4)` should contain the score for English. The scores should be computed so that a higher score corresponds to a language that is more likely to be the one in which the input text was written.
- The end of the *second part* of the code already contains the instructions that will find the language with the highest score and output the results. The program outputs the scores of the various languages, followed by the identification of the language that has the highest score, and by the *classification margin*, which is the difference between the two highest scores.

3.1.1 Practical assignment

1. Complete the code given in the file `languagerecognizer.m`. Transcribe here the code that you have added to the program. Clearly separate and identify Sections 1, 2 and 3 of the added code.

Section1:

```
Prob_lang(languageindex) = log(double(total_counts(languageindex)+60^3))-log((sum(total_counts)+4*60^3));
```

Section2:

```
appearances(trigramindex) = log(trigramcount+1) - log(double(total_counts(languageindex)+60^3));
```

Section3:

```
scores(languageindex) = sum(appearances) + (Prob_lang(languageindex));
```

2. Once you have completed the code and verified that the recognizer is operating properly, complete the table given below, by writing down the results that you obtained for the pieces of text that are given in the first column.

   The last piece of text is intended to check whether your recognizer is able to properly classify relatively long pieces of text. It is formed by the sentence "I go to the beach. " repeated ten times (in the table, the piece of text is abbreviated). Note that the given sentence has a blank space after the period, so that the repeated sentences are grammatically correct. You may use copy and paste operations to ease the input of this piece of text.

Text	Real language	Recognized language	Score	Classification margin
O curso dura cinco anos.	pt	pt	-155.7	0.73
El mercado está muy lejos.	es	es	-184.8	19.29
Eu vou à loja.	pt	fr	-112.0	3.62
The word é is very short.	en	en	-197.64	0.32
I go to the beach. ... I go to the beach.	en	en	-1327.8	260.87

3. Give a detailed comment on the results that you have obtained for each sentence.

See Below

Análise preliminar das frases:
“O curso dura cinco anos.”:
A linguagem reconhecido nesta frase é a correta embora o valor da margem de classificação seja baixo, sendo que a classe que obteu a classificação mais próxima foi a linguagem espanhola, esta semelhança tem a sua origem em que as duas línguas têm raízes em comum muito perto entre si, por isso para que seja possível distinguir entre si é preciso que existam ocorrências de palavras exclusivas de língua para língua para que se consigam “afastar” as classificações da frase, os valores de classificação também se vão distancia consoante o tamanho do texto a analisar. Neste caso a maioria dos trigramas são comuns nas duas linguagens, sendo difícil distinguir entre o português e o espanhol.

“El mercado está muy lejos.” :
Neste caso a linguagem reconhecida pelo classificador corresponde à linguagem real, e o valor da margem de classificador é consideravelmente elevado, ou seja, com este texto o classificador tinha informação para “afastar a concorrência” sendo que o texto a analisar possui característica comuns na linguagem espanhol que são raros nas outras linguagens, fazendo diminuir a pontuação das outras linguagens quando analisaram trigramas característicos da linguagem espanhola:
Como caso de estudo, por exemplo o trigrama “muy”, nos ficheiros utilizados para o treino do classificador aparece 355233 vezes no espanhol, 3103 no francês, 1218 no português e 10 no inglês. Não tomar estes resultados como uma representação total do texto a ser analisado, mas consegue-se encontrar uma preferência evidente com a língua espanhola.

“Eu vou à loja.”:
Aqui a linguagem reconhecida não corresponde à real sendo que o classificador considerou que estava perante um texto na língua francesa quando na realidade se encontrava perante a portuguesa. No entanto podemos ver que a língua com melhor cotação a seguir à francesa foi a portuguesa, no entanto o valor da margem de classificação não é baixo. Sendo que neste caso estas duas línguas também têm origem no latim as suas raízes também são comuns sendo que existem palavras/(silabas). Parte da possibilidade de o classificador tenha errado pode vir do caso do facto de o texto a ser analisado ser relativamente pequeno(contém apenas 12 trigramas), este conjunto de condições causa que o classificador não irá conseguir encontrar trigramas comuns numa linguagem que seja raro noutra, sendo que após várias ocorrências da situação agora definida o classificador irá isolar o candidato mais adequado ao texto a ser analisado. Como a maioria dos trigramas são comuns na duas linguagens se ocorre algum trigrama muito comum numa linguagem específica poderá fazer “tip the scale” para uma das linguagens. Como é o caso do trigrama “ à ” que ocorre no francês 12 milhões de vezes e no português 500 mil vezes.

“the word é is very short.”:
Certo
Se não pusermos os ponto final o classificador considera o texto como, sendo que com a adição do ponto final o classificar vai passar a analisar o trigrama “rt.”, que vai tomar valores muito baixos no “dicionário portugues”(A DATA NÃO CORRESPONDE COM A SUPOSIÇÃO) português dizer que se este for o caso, não é comum palavras em português que acabem em t e,.
O trigrama “ é ” é muito comum no português 1966239 e não existe no inglês. Sendo que o score do português deste trigrama “é i” = 55737 e em inglês = 57. Sendo que estes trigramas vão causar que a classificação geral do texto não irá conseguir isolar a língua inglesa como a língua correspondente do texto introduzido.
Sendo que neste texto a margem do classificador é muito baixo, indicande que é preciso um sample size superior para que se apresentem resultados mais fiáveis.

“I go to the beach … I go to the beach.“ a frase está repetida 10 vezes
Neste caso como o texto que temos para analisar é consideravelmente superior logo vai apresentar uma maior margem de classificação , no entanto não nos encontramos perante um bom exemplo pois como o conteúdo vai estar repetido, se a frase original, neste caso “I go to the beach. I ” Possuir uma margem de erro pequena pode querer indicar que está mal classificada logo, quando repetida poderá levar a um margem superior, podendo estar errado. No entanto a frase original neste caso possui evidentemente um terno que é característico da língua inglesa com um valor elevado de presença e evidente em comparação com as outras línguas às quais estamos a comparar

Perante a situação que nos encontramos a repetição de uma frase não vai causar um aumento exponencial do valor do score pois este não esta a “tomar atenção à ocorrência de vários trigramas na mesma frase, congunto de trigramas não tem infuencia nenhuma para o score total, por exemplo co existência de I e the na mesma frase (pedaços da frases reconhecidos como comuns na língua inglesa) não vão gerar um score suplementar se aparecem mais vezes sendo apenas contabilizados o valor da apparence no dicionário usado para testes.”

???????? o “the” e “ i “ é mais comum nas outras línguas que no ingles???????