Final Project in the course Linear and Combinatorial Optimization

Classifying documents by language

First Name: Konstantinos

Last Name: Stavropoulos

Contents

[Summary 1](#_Toc172898483)

[Analysis 2](#_Toc172898484)

[Norm 1 4](#_Toc172898485)

[Norm infinity 10](#_Toc172898486)

[Bibliography 14](#_Toc172898487)

# Summary

There are languages that share exactly the same alphabet, so it is not easy to tell which language a text belongs to. In this paper we study this problem for Finnish and Swedish languages that share a common alphabet of 29 letters. Specifically, we are developing an approach that combines artificial intelligence methods with linear optimization to solve the problem. In fact, we compare two different methods of modeling the objective function, we evaluate the performance of the system in each method to predict language in unknown texts and present the results.

# Analysis

For each text we select certain quantitative characteristics that will then be used as indications to determine the language of the text. The most critical characteristics are the frequency of appearance of letters and the average length of words, which we choose to use. In fact, the Finnish and Swedish alphabets consist of 29 letters, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, Å, Ä, Ö.

It is worth noting that this problem has been inspired by the book by Sierksma and Zwols [1]. Of course, different languages have been used in this project , specifically Finnish and Swedish, which share the same alphabet and therefore it is not easy to separate them. We have also added to the characteristics proposed by the book the average length of words, which as we will see below is the defining feature in the model for norm 1.

So, for each text we calculate a vector of 30 elements **f**=[f1 f2 .. f30] ( the frequency of occurrence of letters of the alphabet and the average length of words). In total we have 9 Finnish and 9 Swedish texts. The first 6 of each language will be used to train our model and define the parameters and the remaining 3 of each language for testing. That is, to evaluate the system on real texts on which it has not been trained.

In fact, we use a linear classification function

g(fd)= i fid + b d=1,..,18 texts

wi : The weights of each attribute i=1..30

b: The polarization

Any text d that has a vector of attributes fd is assigned a scalar value by the classification function. The task is to find appropriate weights w and polarization b so that the separator g(f) correctly recognizes the texts.

Therefore, the decision variables of the model we have developed so far are the weights wi, i=1..30 and b.

We assume that when the value assigned by the separator is greater than or equal to unit, then the text d belongs to Finnish and when the value is less than or equal to -1 it belongs to Swedish

g(fd)Finnish

g(fd)

So the **constraints** of the problem are initially written as follows:

w1\* f1d + ... + w30\*f30d + b for each Finnish text

w1\* f1d + ... + w30\*f30d + b for each Swedish text

b,wi

It is worth mentioning that a value of 1 for a separation threshold can be chosen arbitrarily without affecting the problem. This is immediately apparent from the fact that if instead of 1 we set e.g. 10 then all the parameters w,b of the problem in the optimal solution would simply be scaled by 10 and the classifier would again recognize the same texts as Finnish and Swedish as if the threshold value was 1.

However, the possible separation functions can be many. We want to choose the one that maximizes the separation distance from the values in the individual texts. To visualize the meaning of the separation distance, suppose that there are only two attributes f1,f2 and g(f)= w1f1+w2f2+b ( each separator has different values of w1,w2,b and the seperatos are the lines g(f)=a=> w1f1+w2f2+b=a, where α in the example is random).

*A graph with red green and blue dots

Description automatically generated*

Image 1 Physical significance of separation distance

In the figure above, both separators correctly separate the set of points. However, the yellow H2 separator is better than the red one because the red is very close to the elements of the blue group. Therefore, it is possible that a new sample is very close to the blue elements but on the opposite side of the separator and will be wrongly categorized. In fact, in our problem the separation distance is defined similarly and concerns the 30-dimensional space.

The exact definition of separation distance is the minimum distance of the separator from group A and group B and it is shown [1] to be equal to

Width\_distance(w,b) =

We want to maximize the separation distance, i.e. minimize ||w||, so **the norm of w is the objective function,** which we want to minimize . Another reason why we want to have small values in the components of the weight vector is that if there is a component with a large value then the result will be determined almost entirely by it.

Therefore, the problem is modeled as follows

min ||w||

i fid + b for each Finnish text d

i fid + b for each swedish text d

b,wi

For each of the 30 components of w we define two auxiliary variables.

ui1= max{0,wi}, ui2 = max{0, -wi} ui1, ui2

Thus, we have

wi= ui1 - ui2 and |wi| = ui1 + ui2

For the objective function we then examine two norms, the first norm and the infinity norm and compare their performance.

## Norm 1

The first norm is defined as follows:

||w||=

The absolute value is eliminated with the auxiliary variables mentioned above. Therefore, the **decision variables are ultimately ui1, ui2 i=1..30 and b.** So the problem becomes the linear model

min

fid – fid ) + b for each Finnish text d

*fid – fid ) + b*  for each Swedish text d

ui1, ui2 b i=1..30

The linear separator is now

g(fd)= f + b

g(fd)

g(fd) *Swedish*

Then we develop a program that calculates fid vectors. Initially, for each text we measure the frequency of appearance of the letters. This is calculated as the quotient of the number of each letter and the total number of letters. We also calculate the average word length. This process results in a vector of 30 elements for each text. Each vector is stored in an F-matrix, which is two-dimensional and includes 18 (as many texts) 30-dimensional vectors. The code that implements this function is shown below.

A screenshot of a computer program

Description automatically generated

Image 2 Code for extracting the characteristics of Text

Moreover, to calculate the optimal values of the decision variables we use as mentioned above 6 Finnish and 6 Swedish texts while the rest are used to evaluate the model.

Therefore, we have 12 constraints and (2\*30)+1=61 decision variables, the

u11 u12, u21 u22, u31 u32, u4­1 u42 ..., u301 u302, b.

To solve the problem we use pymprog's solver. The code is shown below

A screenshot of a computer program

Description automatically generated

Image 3 Solving a problem with norm 1 as an objective function

The solve\_first\_norm function returns the optimal values of the decision variables that the linear separator will then use to evaluate unknown texts.

The optimal values are shown below in the order we write above, namely u11 u12, u21 u22, u31 u32, u4­1 u42 ..., u301 u302, b

A white background with dots

Description automatically generated

Image 4 Optimal values of decision variables as an objective function of norm 1

We notice that b=-16.6, u301 = 3.3 and all other values are 0. That is, the model considers only the average length of words and the polarization b. The minimum value of norm 1 is 3.3.

To verify the correctness of our model we will test it on texts that have not been used in training. 6 texts will be used, namely the files f7,f8,f9 contain Finnish texts and the files s7,s8,s9 Swedish texts. The code for evaluating the model is shown below.

A screenshot of a computer program

Description automatically generated

Image 5 Model evaluation code in unknown texts

If the value of the linear separator is greater than or equal to 1 then the text is classified as "Finnish", if it is from 0 to 1 it is classified as "Finnish probably", if it is less than or equal to -1 then it is classified as "Swedish" and if it is from -1 to 0 in "Swedish probably". As mentioned for evaluation we will use the files in the texts folder with names f7.txt, f8.txt, f9.txt and s7.txt, s8.txt, s9.txt containing respectively 3 Finnish and 3 Swedish texts. In case you want to test the model on your own data, you only need to save the Finnish or Swedish text you want to categorize in one of the above files.

The results of the evaluation are shown below.

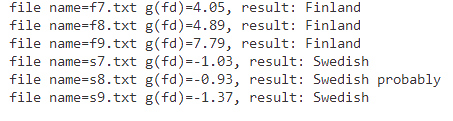


Image 6 Evaluation results based on norm 1

We notice that the model has very good accuracy. 5 out of 6 texts are evaluated correctly and with a large margin, while one text is evaluated correctly but without being absolutely sure because the value of the linear separator is -0.93, i.e. a little more than -1. We then use infinity norm as an objective function for comparison.

## Norm infinity

As we have mentioned, the purpose of the problem is to minimize the norm, min ||w||, to have a large separation distance in the data.

It turns out [2] that

p = max{ |w1|,.. ,|wm| }

So the objective function becomes, min ||w||= min ( max{ |w1|,.. ,|wm| } )

We set x= max{ |w1|,.. ,|wm| } and the point is to minimize x, min(x) .

So we need to add to the constraints of the previous model the following constraints:

|wi| => ui1+ui2 => ui1+ui2-x , i=1..30

x

Therefore, the problem becomes

min (x)

fid – fid ) + b for each Finnish text d

*fid – fid ) + b*  for each Swedish text d

ui1+ui2 => ui1+ui2-x

ui1, ui2, xb i=1..30

There is one more decision variable, x, so in total there are 61+1=62 decision variables. From the restrictions ui1+ui2-x are added 30 more restrictions so we have a total of 12 + 30 = 42 restrictions.

The code for finding the optimal values of the decision variables with objective function the infinity norm is shown below

A screenshot of a computer program

Description automatically generated

Image 7 Solution by objective function of infinity norm

The code is almost identical to that for an objective function of norm 1, except that because another decision variable is added, the vector of objective coefficients has 62 elements.

The results are shown below.

A white background with black text

Description automatically generated

Image 8 Problem solving with an objective function of the infinity norm

We observe that the optimal value of the objective function is z=1.62 and b=-8.2 and x=1.62. In the figure above, below the value of the objective function , the 62 decision variables are listed in the order u11 u12, u21 u22, u31 u32, u4­1 u42 ..., u301 u302, b, x.

We notice that there are many nonzero coefficients, which means that the linear model now looks at many features of the texts. Also, whereas with norm 1 there was a decision variable with an optimal value of 3.3, with the infinity norm all decision variables have lower values. This is because the objective function is min(x)= min ( max{ |w1|, .. ,|wm| } )= min( max{ u1\_1+u1\_2,.. ,u30\_1+u30­\_2 } ). Therefore, the objective function in the infinity norm "seeks" to keep all variables ui low, while in norm 1 it "seeks" their absolute sum to be small.

The optimal values of the decision variables are then used by the model to predict the class 6 of unknown texts, using the same code as before. The results are displayed for both cases of the objective function below.

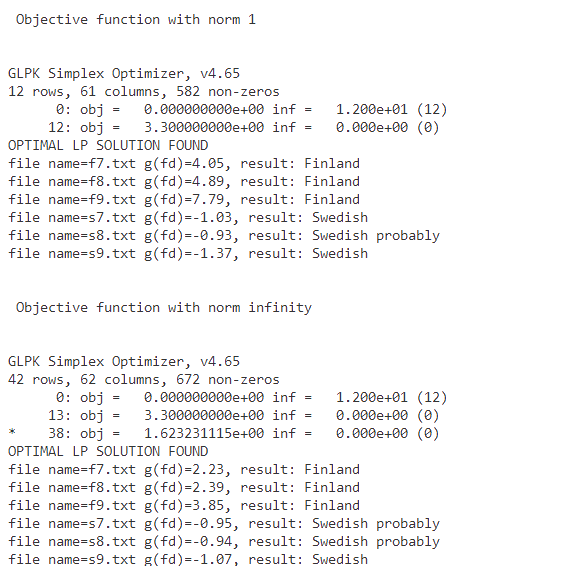


Image 9 Results of the two cases

We observe that the objective function with an infinite norm recognizes, as in the previous case, all unknown texts correctly. However, although for a small difference, there are two cases where the model is not completely certain about the categorization of texts, whereas with norm 1 this was the case in only one case. Therefore, although the infinity norm looks at more features, in this implementation it has a slightly worse precision than norm 1. This is because norm 1 allowed the model to assign a large weight to a single feature that turns out to be particularly important, the average length of words, since the rest were assigned a small weight, while the infinity norm makes it difficult to have any large weight.

Finally, we mention that in both models all nonzero coefficients in the objective function are 1 and it makes no sense for them to undergo any perturbation since they resulted from the way the model is constructed. Also, sign restrictions do not make sense to suffer any disturbance since necessarily the frequency of occurrence of a letter is a non-negative number. Finally, the vector b does not make sense to undergo any perturbation since it was defined by the way the model is constructed and is not subject to perturbation. Besides, as explained above, the value chosen as the separation threshold does not matter and does not affect how the unknown texts will be categorized since any random number could be selected and the optimal values of the decision variables would be scaled accordingly.

In conclusion, the linear separator (classifier) we implemented can predict with great accuracy the language of a random text. In fact, as parameters we defined the frequency of occurrence of the letters of the alphabet and the average size of the words, which, as we saw in norm 1, was the main characteristic for determining the language of the text. We then compared two options for the objective function, norm 1 and norm infinity, and found that both categorize the texts correctly. Finally, we concluded that norm 1 for this implementation is slightly more precise than the infinity norm.

# Bibliography

[1] LINEAR AND INTEGER OPTIMIZATION Theory and Practice Third Edition, Gerard Sierksma and Yori Zwols, page 433

[2] LINEAR AND INTEGER OPTIMIZATION Theory and Practice Third Edition, Gerard Sierksma and Yori Zwols, page 435

[3] <https://en.wikipedia.org/wiki/Linear_classifier>