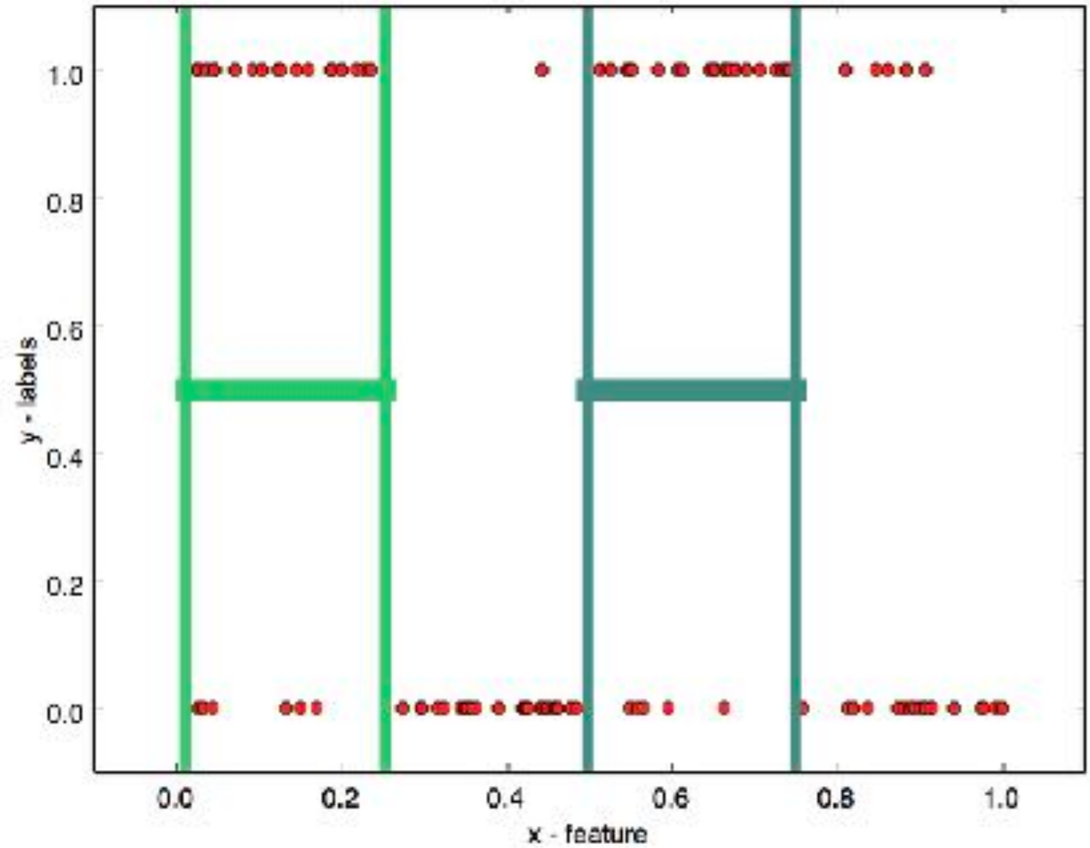
Machine Learning Ex. 2

Discussions and plots for the programming section

Ido Calman, Jonathan Somer - 24 November 2017

A. Plotting 100 points agains k=2 intervals

In this section we plot 100 random points and run the ERM on this data set. We can see that the segments chosen fit the labels. Segments correspond with “1-labeled areas”



B. Calculating the hypothesis with optimal true error:

There are effectively two kinds of segments in our case. Segments [0,0.25] and [0.5,0.75] which we will call “08 segments” because of a 0.8 chance for a point in these segments to be labelled 1. The other kind are respectively “01 segments”.
The Error is thus:

(length of intervals in 08 segments) * 0.2 +
(length of non-interval sections in 08 segments) * 0.8 +
(length of intervals in 01 segments) * 0.9 +
(length of non-interval sections in 01 segments) * 0.1

We will treat the first kind:
Let us name the “length of intervals in 08 segments” - ‘X’

So in 08 segments we have an error of:

$X*0.2 + (0.5 - X) * 0.8 = 0.2X + 0.4 - 0.6X = 0.4 - 0.4X$

The error is minimized by the maximal X, 0.5 and the total error is 0.2.

Using the same logic we find that in 01 sections X = 0 minimizes error.

Thus, the optimal intervals are:

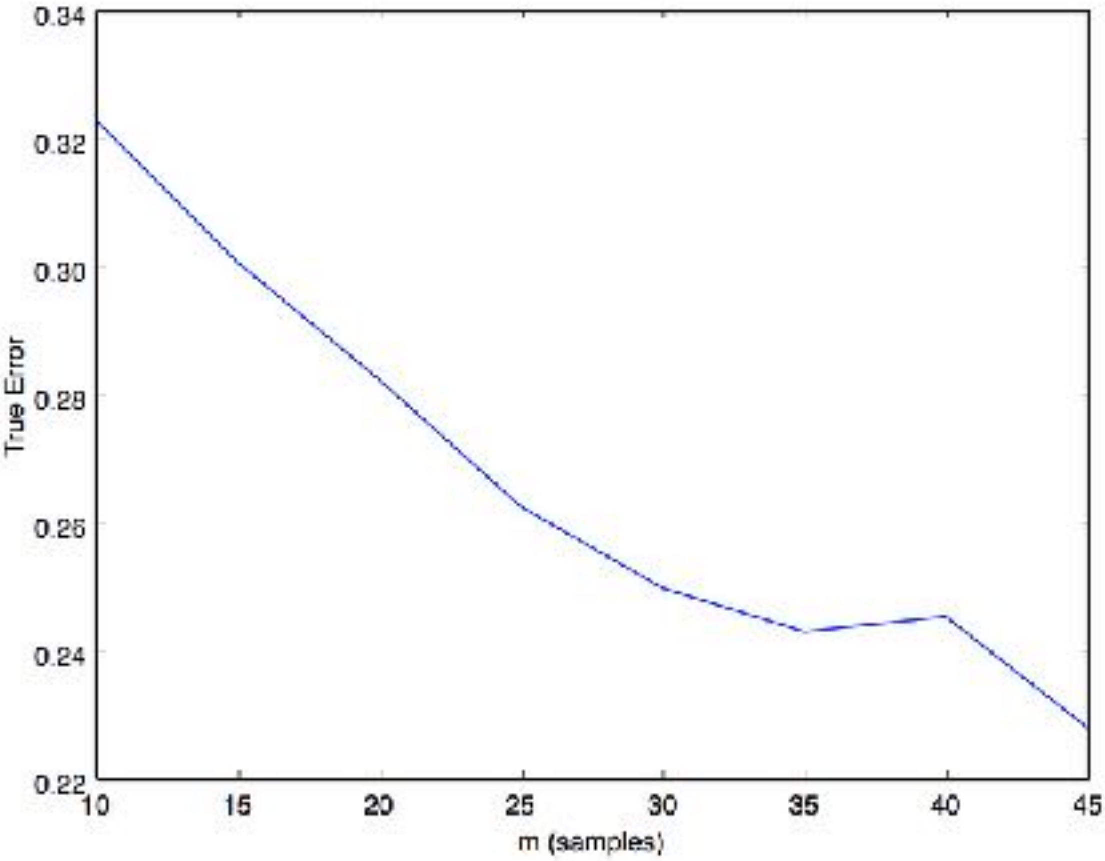
[0,0.25], [0.5,0.75]

This is very intuitive: have intervals in areas with a greater probability of labelling points!

The optimal error is: 0.1*0.5 + 0.2*0.5 = 0.15

C. Plotting True and Empirical Error against m:

We expect the true error to decrease as the size of the sample increases, as we are learning on more data, thus the distribution of the data will be closer to the true distribution.
This is evidently the case in the plot below:

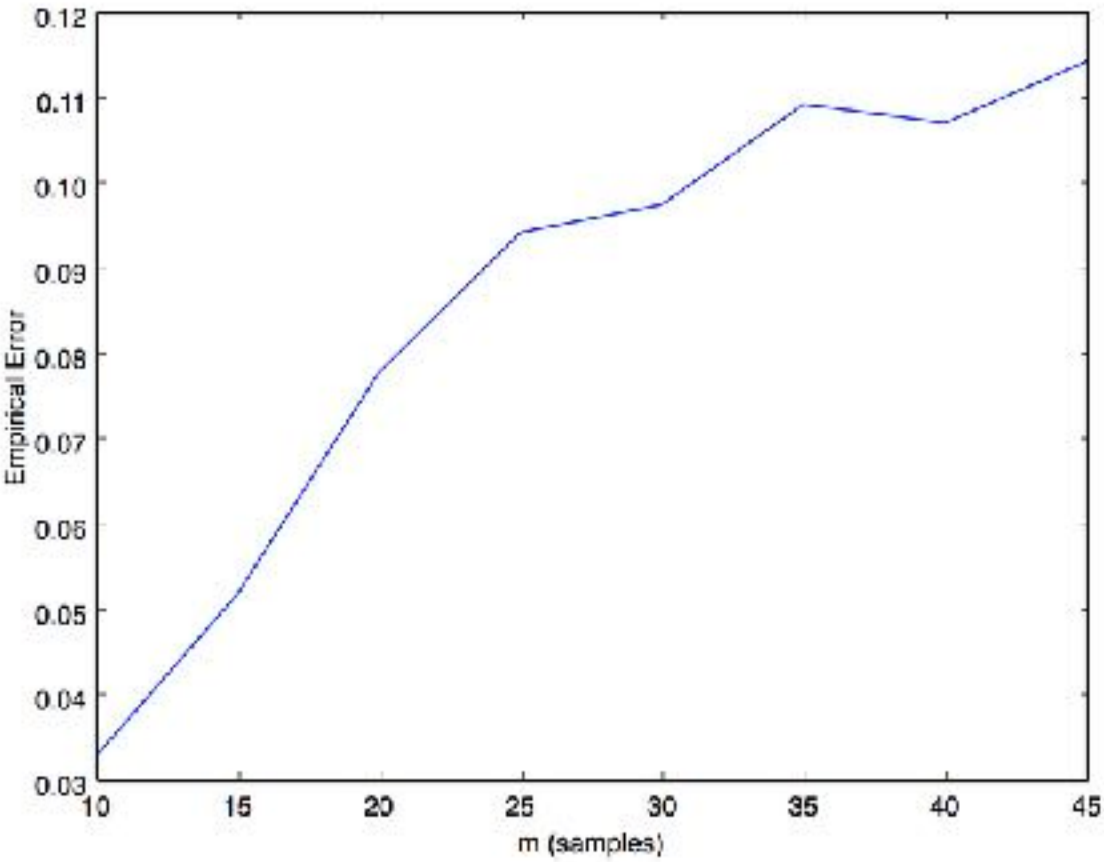


Regarding empirical error, in first glance you might expect something similar, but this is not the case.

Lets look at a trivial case with one point. It is obvious we can reach an empirical error of 0 in this case (either place an interval on it if it is labelled, else, do not place one). We can do the same with 2 points, and with a bit of thinking also for 3.

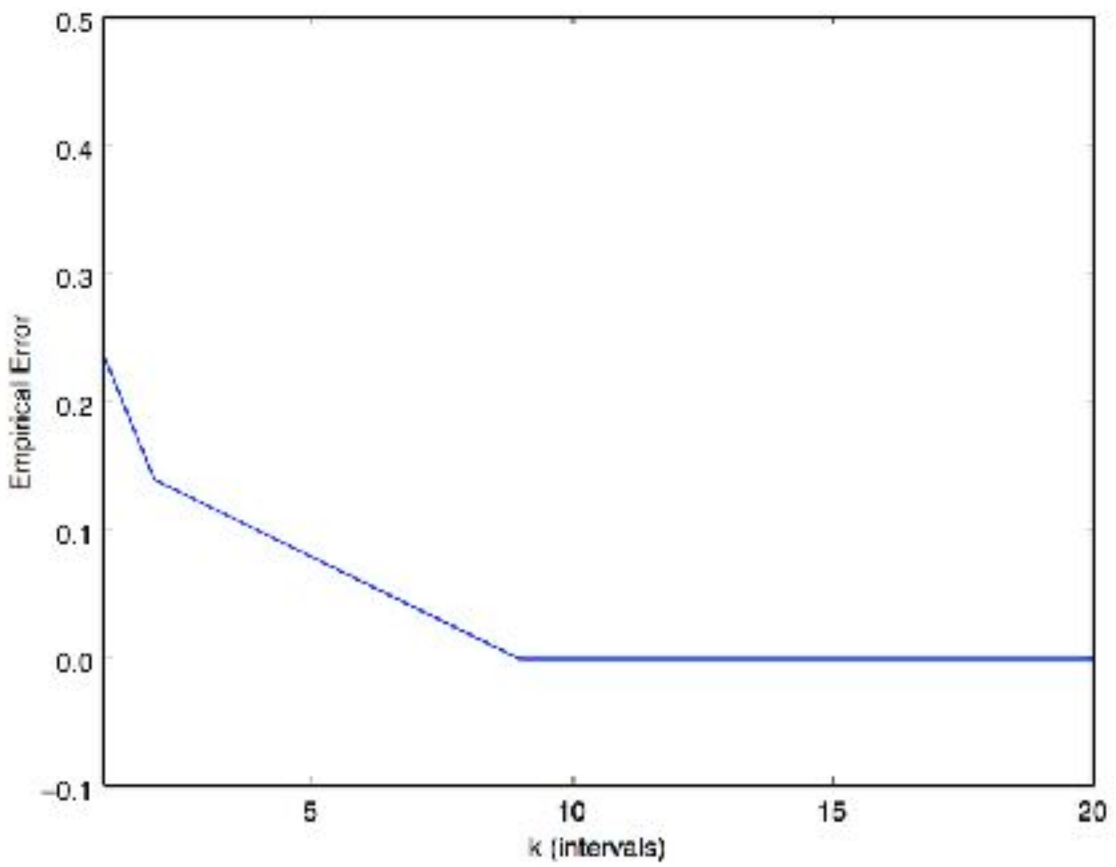
But, as the number of points increases we will find that we have to place intervals such that points that are not labelled lie inside the intervals, and we will miss some labelled points. This is due to the more complex patterns of intertwined labelled and non labelled points.

We expect the error on both plots to tend to 0.15 with as m increases, and this seems to be the case exactly! Recall that 0.15 is the optimal true error, and that we were able to achieve this with k=2.



D. Plotting True and Empirical Error against k:

We expect the empirical error to decrease as k increases. The algorithm gains much flexibility from the added intervals. To understand this think of having - one interval per point. It is obvious you can reach 0 error in this case.
We can see that the empirical case reaches 0 error in 10 intervals and then stagnates. This is as expected.



In contrast, the true error increases as K increases (for K > 2 that is). This is due to the overly-complex hypotheses generated by large K. This is also known as overfitting. The large K enables the algorithm to generate a hypothesis that is very specific to the data, but misses the underlying pattern.

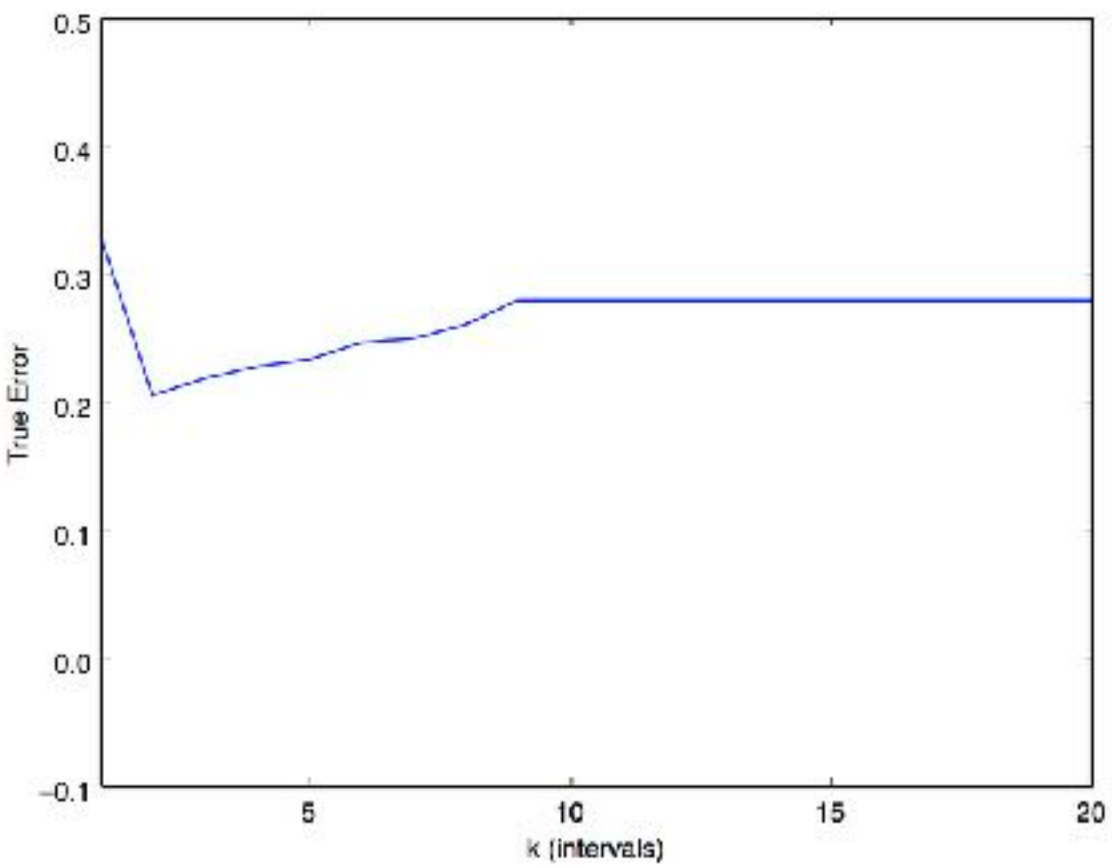
We can see on the plot that that this is the case for 2 < k <=10.

But why does this graph stop growing at k=10?

We suspected that because the algorithm can reach optimal empirical error for k=10, the rest of the intervals are 0 intervals that do not have an effect on the true error.

This was exactly the case! For K=15 the algorithm returned only 10 intervals:

[(0.0043495015640306334, 0.026736647878617403),(0.066487174918874459, 0.094925286092722549),(0.15227047153507717, 0.1680597728442475),(0.1847862391592926, 0.24982310863596896),(0.50710053589544857, 0.53830513050402207),(0.54212834027200263, 0.55488038484296354),(0.55981390551741661, 0.56546426506839276),(0.59435486808720506, 0.6821039658169572),(0.7069314503930304, 0.74167920559959599),(0.79627414966679022, 0.817968366120299)]



E. Finding Optimal K with Holdout Validation:

We know that an optimal error is reached with k=2.
Using holdout validation we are returned the optimal K for a specific sample and test data. This does not guarantee an optimal K.
As we can see in the plot below, during this specific example we have found K=3 to be optimal. Although K != 2 we can see that the segments "behave as 2 segments" as we expect. The algorithm forms 2 segments while avoiding 0 labelled points as much as possible.

This was plotted with m=50 as in part D. For larger m we would expect 2 segments consistently.