Training Dataset for Character Recognition
Pattern Recognition - EQ2340 Assignment 4

Bram Leenders   Deepa Krishnamurthy
leenders@kth.se   deepa@kth.se

October 25, 2015

1 CODE VERIFICATION

The verification was done by checking if the output of the function (`c` and `alfaHat`) matches the given values of `c` and `alfaHat` for the example given in section A.3 of the Exercise Project (page 238). The corrected code passed the test case.

We identified the following errors:

1

Original: `T = size(pX)`
Corrected: `T = size(pX, 2)`
Type of flaw: Logical and Runtime
Comments: T is duration of the observed pattern sequence and hence it is a scalar value and not a vector.

2

Original: `alfaTemp = zeros(numberOfStates);`
Corrected: `alfaTemp = zeros(numberOfStates, 1);`
Type of flaw: Logical
Comments: alfaTemp is a temporary value should get reset after calculating alfaHat for every time step 't'. Also, since it is reset everytime it will be a row vector with number of states as its size. Without this correction, we get incorrect dimensions of alfaHat and hence alfaHat in general becomes incorrect.

3

Original: `alfaTemp(j) = alfaTemp(j)/c(2)`
Corrected: `alfaTemp(j) = alfaTemp(j)/c(t,1);`
Type of flaw: Logical

Comments: Incorrect c and alfaHat values and dimension. alfaTemp has to be normalised with c at that time instance 't'. This is because, alfaHat for a time instance 't' is the normalized state probabilities given the observations at the time instance 't'.

4

Original: `alfaHat = [alfaHat alfaTemp']`
Corrected: `alfaHat = [alfaHat alfaTemp]`
Type of flaw: Runtime
Comments: Error : Dimensions of matrices being concatenated are not consistent.

5

Original: `c(max(rows,columns)) = 0.0581]`
Corrected: `c(max(rows,columns)+1) = 0.0581`
Type of flaw: Logical
Comments: For a finite duration HMM, we need to include the special exit condition $S_{(T+1)}$ and hence the exit probability is is for $c_{(T+1)}$.

6

Original: `c(max(rows,columns)+1) = 0.0581;`
Corrected: `c(max(rows,columns)+1) = alfaTemp'*A(:,columns);`
Type of flaw: Logical
Comments: The value is hardcoded (and thus passes the tests) instead of calculated.

2 CHARACTER DATABASE

We have added the following 15 characters to our training dataset: A, B, C, i, O, Q, T, U, V, W, 1, 2, 4, ! and +.
For each character, we made 15 samples so we have a total of 225 samples.

3 WHY THESE CHARACTERS?

We chose these characters for the following reasons:

- A, B and C simply because they are the first three characters.

- T and + because they are specifically mentioned in one of the previous exercises as difficult to distinguish.

- C, O and Q because the "roundish" shape looks visually similar, but they have some small differences (not a full circle for the C, and an extra line for the Q).

- U, V and W because they are somewhat similar in shape too, especially U and V.

- i and ! because the basic shapes are the same (a line and a dot); only the relative position of the dot distinguishes the two.

- 1, 2 and 4 for the sake of having some numbers too. Also, the 1 (being a single line) is similar to the ! and i but doesn't have the dot.

For each character, we used the same drawing movements. For example; when drawing a "T", we first draw a line from the left to the right (the top line) and then a line from the centre top to the bottom.

We only varied angles, drawing speed, size and position (offset) of the characters. We think that this should be sufficient for recognizing one persons handwriting.

Most people will draw a C starting at the top, and ending at the bottom. Of course, it would be totally possible to train our HMM on the reverse movement too (it should result in the same figure), but we don't think it's likely anyone draws it that way.

To visualize the differences in drawing one character repeatedly, we include the following three graphs. Each graph shows the direction element
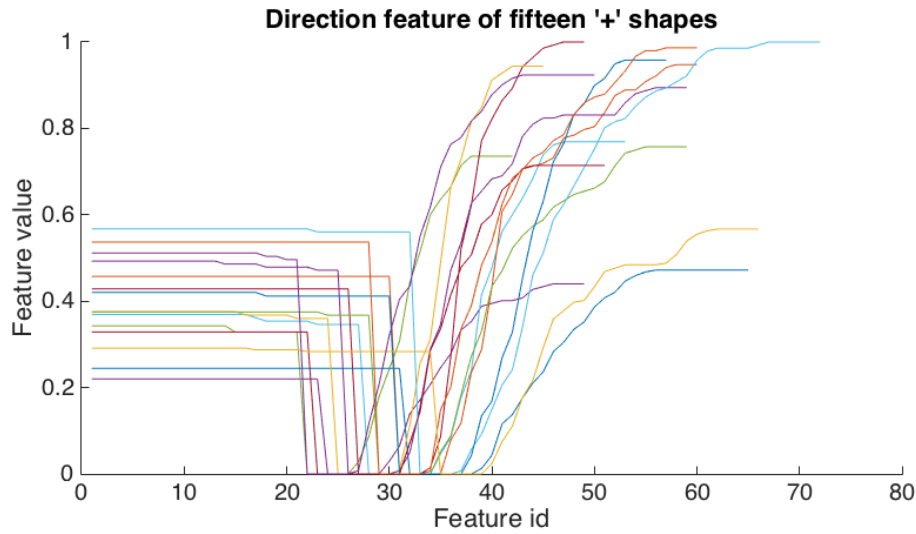

Figure 3.1: Direction vector of fifteen drawings of '+'
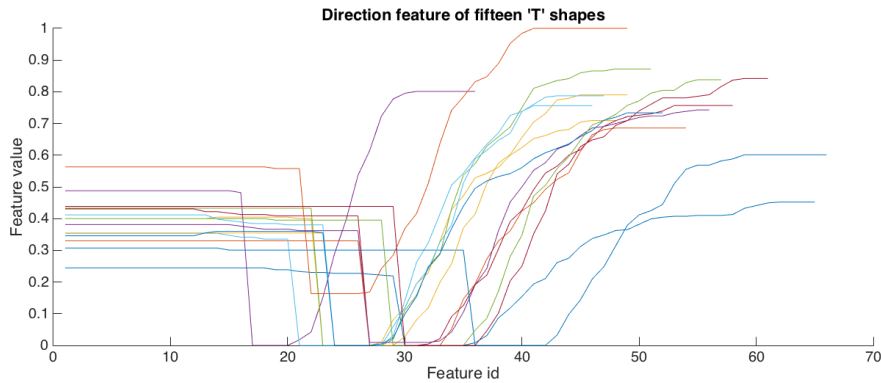

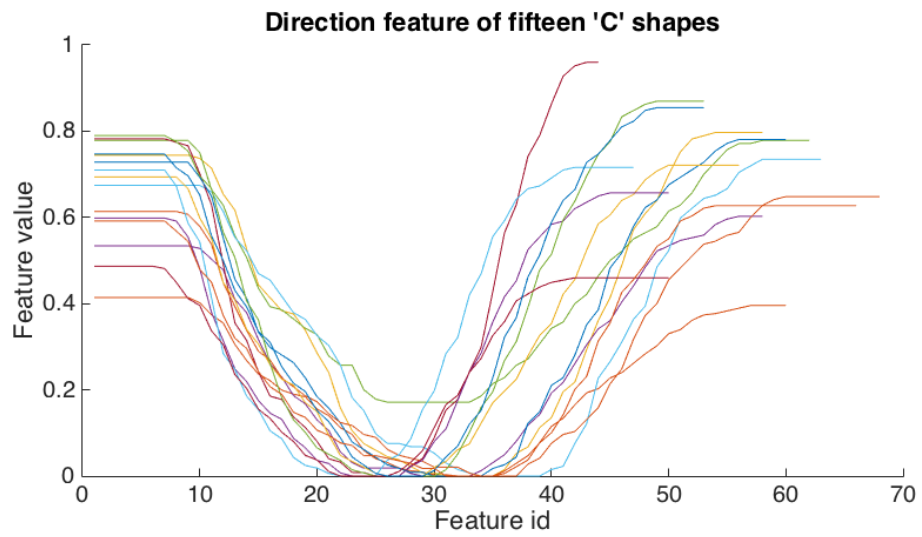Figure 3.2: Direction vector of fifteen drawings of 'T'

Figure 3.3: Direction vector of fifteen drawings of 'C'