ADVANCED METHODS IN NLP
EXERCISE #3 SOLUTION

Uri Avron [uriavron@gmail.com] [308046994]
Ofri Kleinfeld [ofrik@mail.tau.ac.il] [302893680]
Ido Calman [calman.ido@gmail.com] [308353499]

May 11, 2018

Question 2

(b) The accuracy in most frequent model on dev set is **92.31%**

Question 3

(b) We tune the λ_i parameters using a grid search. Our optimal parameters are: We arrive at the following results:

λ_1	λ_2	λ_3	Accuracy
0	0	1	0.923
0	0.1	0.9	0.936
0	0.2	0.8	0.941
0	0.3	0.7	0.944
0	0.4	0.6	0.946
0	0.5	0.5	0.947
0	0.6	0.4	0.949
0	0.7	0.3	0.950
0	0.8	0.2	0.951
0	0.9	0.1	0.951
0	1	0	0.615
0.1	0	0.9	0.938
0.1	0.1	0.8	0.942
0.1	0.2	0.7	0.945
0.1	0.3	0.6	0.947
0.1	0.4	0.5	0.949
0.1	0.5	0.4	0.950
0.1	0.6	0.3	0.951
0.1	0.7	0.2	0.952
0.1	0.8	0.1	0.952
0.1	0.9	0	0.615
0.2	0	0.8	0.943
0.2	0.1	0.7	0.946
0.2	0.2	0.6	0.949
0.2	0.3	0.5	0.950
0.2	0.4	0.4	0.951
0.2	0.5	0.3	0.952
0.2	0.6	0.2	0.953
0.2	0.7	0.1	0.953
0.2	0.8	0	0.615
0.3	0	0.7	0.947
0.3	0.1	0.6	0.949
0.3	0.2	0.5	0.950

0.3	0.3	0.4	0.952
0.3	0.4	0.3	0.953
0.3	0.5	0.2	0.954
0.3	0.6	0.1	0.954
0.3	0.7	0	0.788
0.4	0	0.6	0.949
0.4	0.1	0.5	0.951
0.4	0.2	0.4	0.953
0.4	0.3	0.3	0.954
0.4	0.4	0.2	0.954
0.4	0.5	0.1	0.955
0.4	0.6	0	0.788
0.5	0	0.5	0.951
0.5	0.1	0.4	0.952
0.5	0.2	0.3	0.954
0.5	0.3	0.2	0.954
0.5	0.4	0.1	0.955
0.5	0.5	0	0.615
0.6	0	0.4	0.952
0.6	0.1	0.3	0.954
0.6	0.2	0.2	0.954
0.6	0.3	0.1	0.955
0.6	0.4	0	0.788
0.7	0	0.3	0.954
0.7	0.1	0.2	0.954
0.7	0.2	0.1	0.955
0.7	0.3	0	0.789
0.8	0	0.2	0.954
0.8	0.1	0.1	0.955
0.8	0.2	0	0.789
0.9	**0**	**0.1**	**0.955**
0.9	0.1	0	0.789
1	0	0	0.615

(c) Viterbi accuracy is: **95.55%.** with: $\lambda_1 = 0.9, \lambda_2 = 0, \lambda_3 = 0.1$
Our pruning policy works as follows: on hmm_train we construct the emission tag count dictionary. For each encountered word in the sentence we hold the list of associated tags. Now, in test time we use this dictionary to only iterate over associated tags which we saw in training time.

Question 4

(c) We made the following optimizations in order to make Viterbi evaluation work faster:
- In training time, we maintain a dictionary **S** where each encountered word points to a list of tags which it was associated with. Then in test time, we only evaluate **u, t** such that appear in the dictionary and so we do not run over the whole tag set.
- We perform only **one** logistic regression prediction per word in sentence **k** with a matrix that contains all relevant **u, t**. This way we take advantage of a prediction in a parallel fashion in sci-kit implementation.
- Finally, we use **vec.transform** parallel implementation for all relevant **u, t** instead of extracting features one by one.

(d) The accuracy on development set is: Greedy: **95.46%** Viterbi: **96.03%**

(e) A few examples of sentences which Viterbi errs on:

"... members of the House Ways and Means Committee introduced..."
In here Viterbi predicted **Ways** and **Means** to be **NNPS** which stands for proper noun, plural. The actual tag is **NNP** which is proper noun, singular. This kind of mistake seems legitimate because the words end with an "s" which normally indicates a plural form. In here the words are used as names of events and thus are labeled as singular.

"PainWebber also was able to gear up quickly thanks to..."
In here Viterbi predicted **gear** to be **NN** which is non, singular or mass. The actual tag is **VB** which is verb, base form. This mistake makes sense because the phrase has a verb which is "was", and an algorithm could fairly expect the "to" proposition word to be such that reflects the next word ("gear") to be the subject of the verb.