

Assignment-2 (CS-419)

For training and prediction/inference with HMMs, you are free to use any software on Internet¹.

1. **Small Vocabulary Speech Recognition**²: The input data or the observed variable (x) is exactly in the format as discussed in lectures i.e., sequence of vectors. However, its a small vocabulary digit recognition problem, hence it makes sense to pose this problem as multi-class classification. Here is what you have to do: take training data for each digit (including male and female speakers) and train HMM models for each digit. Using these models, perform MAP labeling for the test data. Try various number of hidden states and structures for the Markov chain (left-right or allow all etc.). You can also use Table1 in <http://www.mghamdi.com/IPCVO8.pdf> to know the phonemes involved in the digits. Provide a comparison of training and test accuracies with the various parameters/structures you have tried.
2. Read the problem above and assume you are given the same dataset. Design and implement a predictive model that involves only one HMM (with phonemes as hidden states or 3 hidden states per phonemes etc., as discussed in lectures) rather than 10 HMMs as in the previous question. Again, try various meaningful choices for number of hidden parameters/structures and report training and test accuracies with each choice. Intuitively, which method, the above or this will perform better and why (list the trade-offs). Irrespective of your answer to this question, can you suggest improvements to the single-HMM model?
3. **Speaker Recognition**³: The input data or observed variable (x) is a sequence of vectors, hence HMM is one option to be used for modelling. Here is what you have to do: Using training data for each speaker, build a HMM model for each speaker. Using these 9 models, perform a MAP labeling of the test datapoints. Try various meaningful choices of number of hidden states and the structure for the Markov chain (left-to-right or allow all etc.). For the various choices tried, compare the training and test accuracies.

¹<http://torch.ch/torch3/> or <http://doc.gold.ac.uk/~mas02mg/software/hmmweka/> or many others.

²Dataset available at <http://archive.ics.uci.edu/ml/datasets/Spoken+Arabic+Digit>

³Dataset available at <http://archive.ics.uci.edu/ml/datasets/Japanese+Vowels>.

4. **Classification under Uncertainty:** Often real-world data is noisy. Here we focus on noise in x . There are many cases where some partial information about the noise in x is available:

- (a) Consider dataset at [http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)). Here for each feature, the mean, std-error⁴ and max. value are provided. One way to utilize this information is to estimate a beta distribution⁵ for each example.
- (b) Consider the gene expression dataset at [Available at http://www.ncbi.nlm.nih.gov/geo/](http://www.ncbi.nlm.nih.gov/geo/) with accession number GSE2187. Here, since the micro-array experiments are noisy, for each example x_i , three replicates are provided that represent three runs of the same experiment. One way to utilize this information is to perform MLE for Gaussian distribution⁶ using the three replicates and fit a Gaussian distribution at each example.

In either of the above cases, the original problem is turned into that where input is a distribution and corresponding output is the class-label. Here is what you have to do: build an SVM model using training data⁷. You only need a kernel that compares distributions. For this purpose use the probability product kernels described in <http://www.ai.mit.edu/projects/jmlr/papers/volume5/jebara04a/source/jebara04a.pdf> for exponential family distributions. With various values of ρ parameter in the kernel, compare the training and test accuracies achieved.

⁴http://en.wikipedia.org/wiki/Standard_error

⁵choose beta parameters a, b to best fit the given sample mean, sample std-error and max.value.

⁶Assume spherical covariance i.e., the covariance matrix is diagonal.

⁷Use pre-computed kernels in libsvm www.csie.ntu.edu.tw/~cjlin/libsvm/.