A Hidden Markov Model for *BIO Tagging*

By Cherubin Manokaran

Introduction:

Hidden markov models are probabilistic and generative. Their use of hidden or unobserved variables enable more effective data fitting and generalization. Their application is clear for noun phrase chunking using *BIO tagging* where the three tags are hidden states and the words and its features such as part of speech are observations. Here, a hidden markov model is implemented using both supervised and semisupervised training and both training methods are compared. In addition, the results of different feature sets are also compared. As described in methods, supervised training occurs with labeled data while unsupervised training with unlabeled data using the forward-backward and EM algorithms. Semisupervised training combines both labeled and unlabeled data. Classification, though, implements the viterbi algorithm in all cases.

Methods:

A classifier was built with a dictionary of indices for all features. In addition, a matrix with one axis corresponding to all features and another corresponding to all labels, a matrix of transition probabilities and vectors for both initial and final probabilities were constructed. For supervised training, these matrices were populated with the probability of a given sequence and its individual observations for possible hidden state sequences based on observed counts. Using these emission and transition probabilities and the viterbi algorithm, the best label sequence was inferred given a sequence of observations for new data. A variant of the viterbi algorithm, the forward algorithm was also implemented for later when training unsupervised using the forward-backward algorithm.

For unsupervised training, the expectation was not computed based on counts like above; instead, it was determined using the forward-backward algorithm. A model was initialized with certain parameters, uniformly in this case, and unlabeled data was used to determine a marginal probability for each individual state as opposed to the most likely sequence of states which the viterbi algorithm was previously implemented to determine. In addition, the purpose was quite different. The forward-backward algorithm was implemented for training purposes to optimize parameters based on unlabeled data. After maximizing, the emission and transition probabilities and viterbi algorithm can be used decode a sequence of observations and infer the most likely sequence of states, or labels and tags.

Such training alone produces extremely mediocre results. Instead of initializing a model randomly or uniformly, it was initialized with labeled data first. The above forward-backward and EM algorithms were then implemented to maximize parameters. The combination of labeled and unlabeled data with the unlabeled data serving to maximize and smooth the data set, will improve accuracy. It will provide the most rigorous model with significant generalization.

Results:

1. Words

	B	I	O
B	9903.0	5618.0	1015.0
I	882.0	3834.0	614.0
O	1488.0	5035.0	19157.0

	Precision	Recall	F1
B	0.598875	0.806893	0.687493
I	0.719325	0.264651	0.386941
O	0.745989	0.921630	0.824560

Accuracy: 0.691835

When training the hidden markov model with labels, or tags, and with words as the only features, the number of correctly predicted “B” and “O” tags, as represented in the confusion matrix, are particularly high. Specifically, the “B” and “O” tags had approximately 80 and 92% recall, respectively. The “I” tag, though, had very low recall. A similar trend can also be observed with F1 scores. But based on precision, the “B” tag produced a larged percentage of false positives.

2. Part of Speech

	B	I	O
B	9870.0	8127.0	576.0
I	1208.0	5626.0	310.0
O	1195.0	734.0	19900.0

	Precision	Recall	F1
B	0.531417	0.804204	0.639953

I	0.787514	0.388348	0.520179
O	0.911631	0.957375	0.933943

Accuracy: 0.744458

Training with the part of speech generated similar results. “B” and “O” tags had the highest F1 scores, but the “B” tag also generated the most false positives. The “I” tag also had a much higher F1 score as a result of relatively higher precision and recall levels.

3. Words and Part of Speech

	B	I	O
B	10093.0	5544.0	982.0
I	921.0	4120.0	629.0
O	1259.0	4823.0	19175.0

	Precision	Recall	F1
B	0.607317	0.822374	0.698671
I	0.726631	0.284393	0.408791
O	0.759195	0.922496	0.832917

Accuracy: 0.702225

Training with words and part of speech produced results were even more similar to those of training with words. The F1 scores, precision and recall were all about 1-3% higher.

Overall, it appears that model struggles the most with determining words in the middle of noun phrases and the least with others. Nevertheless, Training with part of speech appears to produce the best results, because it generate an overall accuracy of approximately 74%. Per expectation, training with words resulted in the least accuracy, because the words alone provide the least amount of information. The first task is being able to locate the nouns such that the noun’s position in a noun phrase phrase can then be determined. Training with words increases the difficulty of the first task and, in turn, the second task. Unexpectedly, however, models trained with part of speech performed better those trained with both words and part of speech.

Although both words and part of speech provide the most information, the two pieces of information also result in very sparse features that can lead to overfitting and lack of generalization. For this reason, training hidden markov models with part of speech for BIO tagging generates the most accuracy.

Several smoothing techniques were attempted. Feature count tables were initialized as ones arrays. This demonstrated the most improvement in results. In addition, features with low frequencies were ignored. This alone improved results but when also initializing all feature counts to one failed to bring any significant improvement. The purpose of initializing feature counts to one is to avoid non-zero probabilities that will decrease accuracy. Removing low frequency features will select for features with non-zero probabilities; therefore, both techniques were not required.

Unsupervised training was implemented as described in the methods section. After computing forward and backward values, expectation-maximization was performed based on the initialized the model. This model can be initialized randomly, uniformly or by training with labeled data. After training, new data was classified using the viterbi algorithm. The program has been having issues with underflow and extremely small probabilities.