PARTS OF SPEECH TAGGING USING A BAYESIAN APPROACH

Final Project Proposal

Bayesian Theory and Data Analysis

Due on April 28, 2016 at 02:15pm

Professor: Daniel Manrique-Vallier STAT S626

INDIANA UNIVERSITY

Angad Chandorkar | Roshan Kathawate | Santhosh Soundararajan

.

PARTS OF SPEECH TAGGING USING A BAYESIAN APPROACH

Team Members:

Angad Chandorkar anajchan@umail.iu.edu

Roshan Kathawate rkathawa@umail.iu.edu

Santhosh Soundararajan soundars@umail.iu.edu

Link to the Classifier Implementation:

¡Google Drive: Classifier Code(.py files)¿ - Softcopy mailed to Professor

1. AN EXECUTIVE SUMMARY

1.1 Objective & Problem to be Solved:

Our primary objective behind working on this project is to explore application of Bayesian Approaches to implement Classification on a real-world problem. The problem chosen for this project is Part-of-Speech Tagging. We will be using the following approaches for predicting the Part-Of-Speech Tag for an unknown word:

a. Naive Bayes Classification: While this is a simple approach which assumes conditional independence of the features, it still provides good results in the context of Part-of-Speech tagging. Our intention is to study the nature of the Prior and Likelihood probabilities in the Training Data and implement a Naive Bayes Classifier to compute the Posterior probability and evaluate the quality of results.

b. MCMC Classifier (Gibbs Sampling): Our objective is to understand the nuances of MCMC based sampling methods by implementing the Gibbs Sampler from scratch using Python. Here, we will use the Full Bayesian Network to construct a Gibbs Sampler which will sample from the posterior probability $P(POS\ Tag \mid Word)$ in order to predict the POS tag for a given input word.

1.2 Significance of the Problem:

a. It is a non-trivial task: Identifying a part of speech can be a non-trivial task even for humans. This is because the part of speech generally depends on the context of the word-usage in the sentence. Consider the usage of the word ***that*** in the following sentences. In each sentence, ***that*** has a unique POS-tag:

i. I know that he is honest. POS-tag : Preposition

ii. You cant go that far. POS-tag : Adverb

iii. Yes, that play was nice. POS-tag : Determinant

Thus to predict the POS-tag, we will have to consider the probabilistic relationships among between words in a sentence. We adopt this approach while implementing the MCMC based Gibbs sampler. However, we find that even if we dont account relationships between words of a sentence and rely only on the probabilistic occurrences of words in natural language, we can attain a decent accuracy given we have enough data. We use this approach in Naive Bayes Classification.

b. Traditional Rule-Based approaches will not scale:

Owing to the versatility of natural languages, it will be extremely time-consuming, difficult and an error-prone approach to create a rule-based classification approach. This is another reason why using a probabilistic approach to interpret the interdependencies of words in the language is a better approach.

1.3 Uses of POS Tagging:

i. Converting text to speech. E.g. to determine whether to pronounce construct as CON-struct(n.) or as cun-STRUCT(v.) based on context.

ii. Help in performing Linguistic Analysis. E.g. Create regular expressions for picking out base noun phrases.

iii. POS Tagger can be used as a preprocessor, for instance Text Indexing & Information Retrieval.

iv. POS-Tagger is used as a fundamental tool for building Corpus which is used as a benchmark for all other Text/Language-processing models.

2. DESCRIPTION OF THE DATA

We have used the BROWN CORPUS for creating the Training and Test Datasets provided by Brown University - Standard Corpus of Present-Day American English ($http : //www.nltk.org/nltk_data/$).

2.1 Overview:

a. We will be using the Brown-Corpus for creating the Training and Test Datasets (Data Description included in the section. The Training data comprises 1 million words (955,797 words) in 45k sentences (44204 sentences). Each word has a corresponding label representing its Part-of-Speech tag. Following is the format of the Training and Test datasets:

Nick **NOUN** hit **VERB** the **DET** barrel **NOUN** and **CONJ** threw **VERB**
himself **PRON** upon **ADP** the **DET** smaller **ADJ** man **NOUN**..

b. The classifiers will learn using the Training dataset and will predict the POS-tag for each word in the Test Dataset. The Test Dataset comprises 30,000 words and 2,000 sentences.

c. For our classification task, we will classify an input word to a limited set of 12 Part-of-Speech tags, represented by the set S below:

$$S = \{ADJ(adjective), ADV(adverb), ADP(adposition), CONJ(conjunction),$$
$$DET(determiner), NOUN, NUM(number), PRON(pronoun),$$
$$PRT(particle), VERB, X(foreignword), .(punctuationmark)\}$$

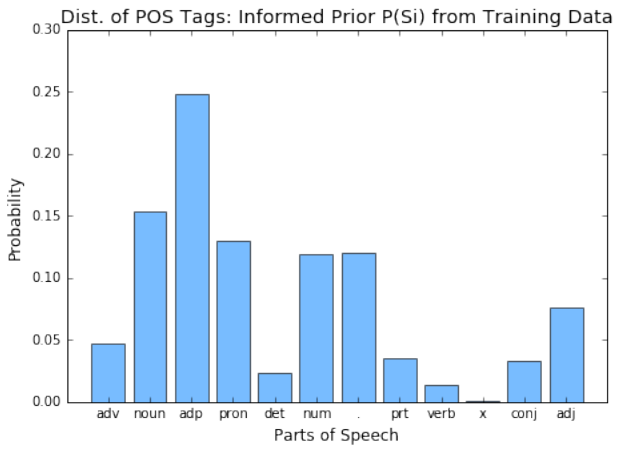
2.2 DATA DETAILS & DESCRIPTIVE STATISTICS:

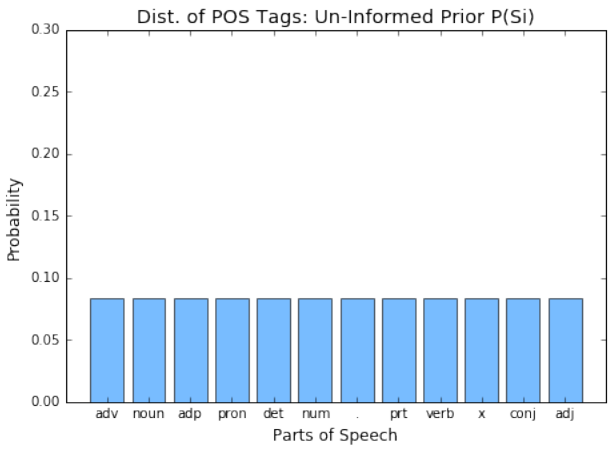
(NOTE: Informed priors are calculated using the Training dataset. In case of uninformed priors, we are using uniform ones, i.e. values are equally likely.)

2.2.1 Prior: We can compute the Prior, i.e Probability of a POS-Tag i.e $P(S_i)$ occurring in Natural Language. Following is a plot of:

fig(a): Informed Prior computed using the Training Dataset

fig(b): Uninformed Prior where each POS-Tag is equally likely





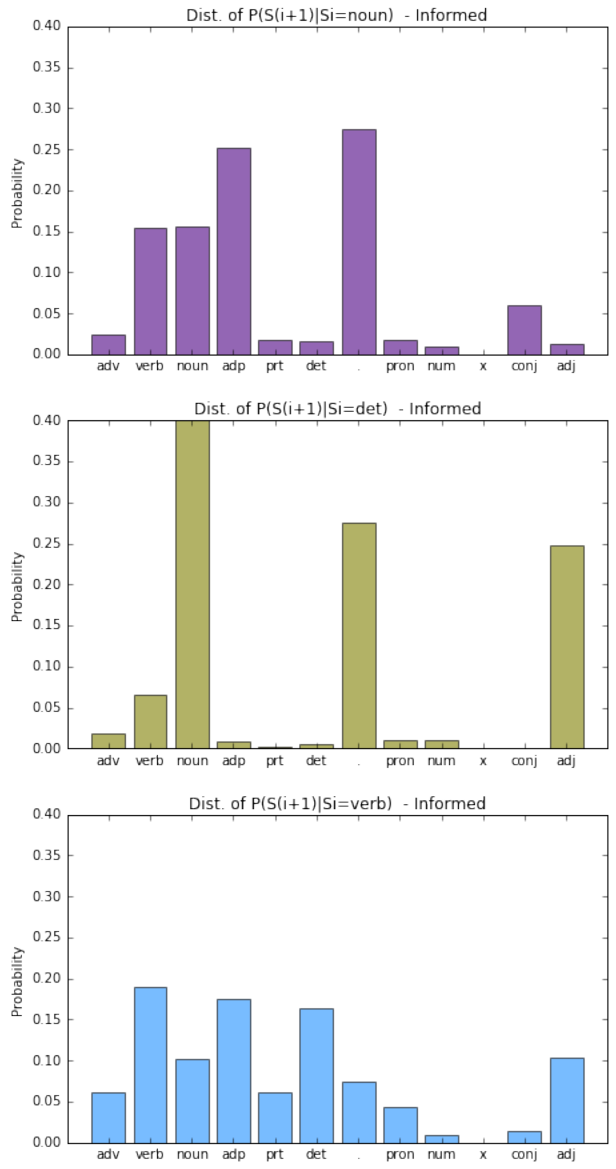
. **fig(a)** **fig(b)**

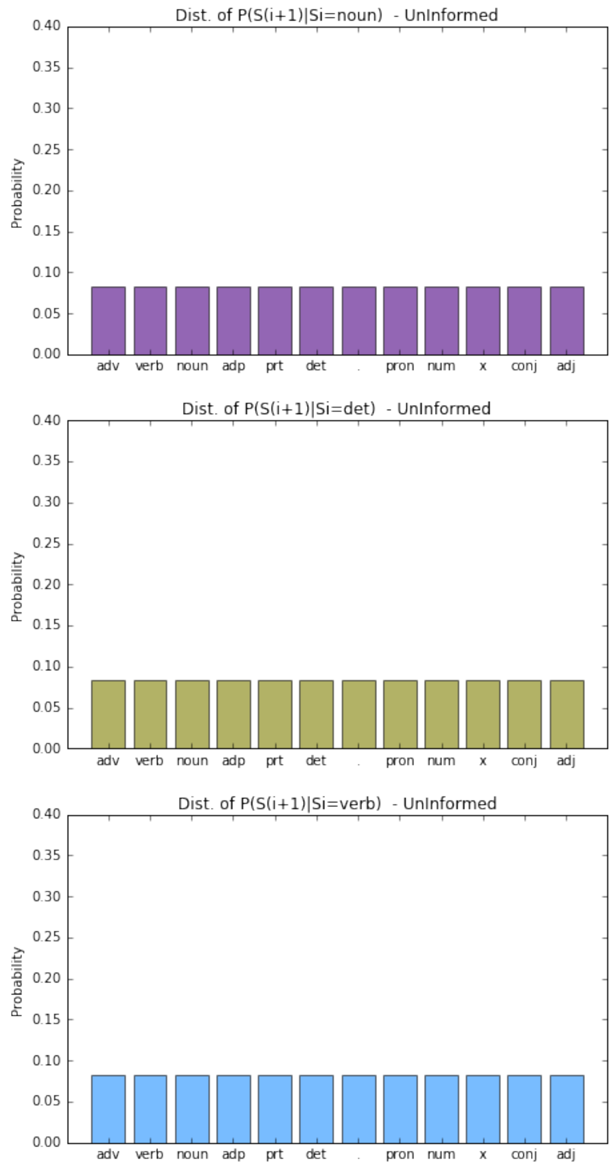
2.2.2 Prior Transition Probabilities: We can also compute the Prior Transition Probabilities, i.e Probability of a POS-Tag given the POS-Tag of the word preceding it, $P(s_{i+1}|s_i)$. Following is a plot of:

fig(c): Informed Transition Probabilities $P(s_{i+1}|s_i)$ computed using the Training Dataset.

fig(d): Uninformed Transition Probabilities $P(s_{i+1}|s_i)$ where each POS-Tag is equally likely.

in fig(c) and fig(d), we plot only values of s_i, where s_i = **[Noun, Det, Verb]**



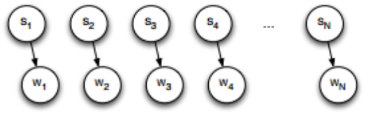


. **fig(c)** **fig(d)**

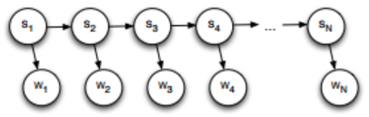
3. BAYESIAN STATISTICAL ANALYSIS OF DATA

3.1 Modelling The Data

To implement Naive Bayes Classification and Markov Chain Monte Carlo Sampling, we will be modelling the data using the following Bayesian Network models.



a. Simplified Bayesian Model



b. Full Bayesian Model

In both the cases, we will have two sets of N Random Variables:

- Hidden Random Variable S: $\{s_1, s_2, ...s_N\}$ represent the part of speech tags, s_i being the tag for the i^{th} word w_i. And $s_i \in S$, which is the set of 12 POS-tags described in 2.1(c).
- Observed Random Variable W: $\{w_1, w_2, ...w_N\}$ represent words, w_i being i^{th} word and $w_i \in \{w | w \text{ is a word in English Language}\}$

Important Probabilistic Relationships between the Random Variables:

- **P**(s_i)**:** Represents the probability of occurrence of a POS-tag. This is also known as a **Prior**. We can model the Prior on as an informed Prior by using the Training Data - fig(a) or model it as an uninformed one where each Tag is equally likely - fig(b).
- **P**$(s_{i+1}|s_i)$**:** Represents the probabilistic relationship between the POS Tag t_{i+1} of an observed word w_{i+1} given that the POS Tag t_i of preceding word w_i is t_i. Based on the Tag t_i for word w_i, we can compute a distribution for the Tag t_{i+1} of the next word w_{i+1}. For e.g. we can determine that nouns will be generally followed by verbs rather than adjectives. This value can represent a **prior belief**, known as a **Transition Probability**. Again, we can use the Training data to formulate an informed opinion about the Transition Probability - fig(c) or we can formulate an uninformed opinion where each tag is equally likely - fig(d).
- **P**$(s_i|w_i)$**:** Represents the probability of a Tag t_i given a specific word w_i. This value is called **Likelihood or the Emission Probability**. The value of Likelihood is computed using the Training Data. Intuitively, we can say that each word will have a probability distribution for all the possible POS tags it can take on. Using this distribution we can determine that the word cat has a higher probability of being a noun than being a verb.

3.2 BAYESIAN INFERENCE:

Using the models, we can implement part-of-speech tagging. A sentence comprising N words will be represented as a Bayes Net model with 2N nodes. Using the N observed words W from the sentence, we perform Bayesian inference to estimate the POS tags in S. Following are the approaches for classification we use:

3.2.1 NAIVE BAYES CLASSIFIER:

For the Naive Bayes Classifier, we will use the Simplified Bayesian Network defined earlier (fig a). The Classifier will be trained first on the basis of the Training data and will then Predict the POS-tags of all words in the Test data. Following is the overview of both these phases :

TRAINING PHASE:

Firstly, using the Training data we will learn Likelihood and Prior beliefs. We can also model our Prior beliefs as being Uninformed.

- **Prior** $P(s_i)$: This is the global Prior Probability of POS-Tags. We will use both Informed and Uninformed priors. The informed prior will comprise values of $P(s_i), s_i \in S$ learned from the Training data while the uninformed prior will comprise values of $P(s_i), s_i \in S$ where every Tag is equally likely.
- **Likelihood** $P(w_i|s_i)$**:** We will compute the value of Likelihood from the Training data.

TESTING PHASE:

Given the values of Prior and Likelihood, the classifiers task is to compute the Posterior probability $P(s|w_i)$, and predict the Unobserved Variable (POS Tag) s_i for input w_i.

- **Posterior** $P(s_i|w_i)$**:** The classifier predicts the most probable POS-tag s_i from the posterior in the following manner:

$$s_i = argmax_{s_i \in S}\ P(s_i|w_i) \qquad (i)$$

In Equation-(i), we can write $P(s_i|w_i)$ using Bayes Theorem as follows:

$$P(s_i|w_i) = \frac{P(s_i) * P(w_i|s_i)}{P(w_i)}$$

Since $P(w_i)$ will remain constant, we can rewrite the above equation as:

$$P(s_i|w_i) = P(s_i) * P(w_i|s_i) \qquad (ii)$$

i.e. $Posterior \approx Prior * Likelihood$

Rewriting Equation - (i) using Equation - (ii), we get:

$$s_i = argmax_{s_i \in S}\ P(s_i) * P(w_i|s_i) \qquad (iii)$$

Therefore, using the Naive Bayes Classifier, we can compute the most probable tag $s_i \in S$ given a word w_i in Test Data by formulating the Posterior $P(s_i|w_i)$ on the basis of the Prior and Likelihood probabilities per Equation - (iii).

3.2.2 GIBBS SAMPLING using MARKOV CHAIN MONTE-CARLO:

Here, we will use the Full Bayesian Network defined earlier (fig b). This technique is based on the following observations:

TRAINING PHASE:

- We can generate a distribution of all possible parts-of-speech tags for given a word, i.e. using Training data we can compute the Likelihood (Emission Probabilities) given by $P(w_i|s_i)$.
- We also know that using the Markov Chain property that a POS-Tag for a given word will be dependent only on the previous state, i.e. previous POS-Tag. We can represent the distribution of possible POS-Tags for a word given the POS-Tag for the previous word using $P(si+1|si)$ where $s_i \in S$. This is Transition Probability and can be considered as our Prior belief. For an informed prior, we can compute these Transition Probabilities from the Training data - fig(c) and for an uninformed prior - fig(d), we can set the Transition Probabilities to be equally likely.

TESTING PHASE:

Now, using the Emission and Transition probabilities, we can emulate the The Posterior Distribution of the POS-Tags and sample the most appropriate POS-Tag for a given word. Following is a mathematical overview of the process:

Considering the Full Bayesian Network (fig b), our objective is to use Gibbs sampling to the sample the probable tags $\{s_1, s_2, ...s_n\}$ for a given word sequence $\{w_1, w_2, ...w_n\}$ in a sentence of Test data. That is,

Sample $s_1, s_2, ...s_n$ from Posterior, P$(s_1, s_2, ...s_n \mid w_1, w_2, ...w_n)$

Using Bayes Law, we can write the Posterior distribution as:

$$P(s_1, s_2, ...s_n \mid w_1, w_2, ...w_n) = \frac{P(w_1, w_2, ...w_n \mid s_1, s_2, ...s_n)P(s_1, s_2, ...s_n)}{P(w_1, w_2, ...w_n)}$$

From this equation, we can ignore the denominator. Also, per the Full Bayesian Network model, the probability of a word is only dependent on its tag, therefore we can write the above equation as:

$$P(s_1, ..s_n|w_1, ...w_n) = P(w_1|s_1)\ P(w_2|s_2)......P(w_n|s_n)\ P(s_1, s_2, ...s_n)$$

Further, owing to the Markov assumption applicable to the Full Bayesian Network, we can write

$$P(s_1, s_2, ...s_n) \approx P(s_1)\ P(s_2|s_1)\ P(s_3|s_2)....P(s_n|s_{n-1})$$

Substituting this in the above equation, we get

$$P(s_1, s_2, ...s_n \mid w_1, w_2, ...w_n) = P(s_1) \prod_{i=1}^{n} P(w_i|s_i) \prod_{i=1}^{n-1} P(s_{i+1}|s_i) \qquad (iv)$$

Now, using in Equation (iv) to represent the Posterior, we can use Gibbs Sampling to estimate the POS-Tag sequence $s_1, s_2, ...s_N$ for a given word sequence $w_1, w_2, ...w_N$. Thus we sample $s_1, s_2, ...s_N$ from the Posterior represented by below equation:

$$P(s_1) \prod_{i=1}^{n} P(w_i|s_i) \prod_{i=1}^{n-1} P(s_{i+1}|s_i) \qquad (v)$$

ALGORITHM IMPLEMENTATION - Gibbs Sampling using MCMC:

I. Factoring in the burn-in period, we iterate over each sentence in the Test-Data multiple times. At the end of this set of iterations, we will have predicted the tag sequence for each sentence.

II. In a given sentence, for each word, we compute the distribution for the Posterior using the formula derived in Equation -(i). Here, we consider each one of the twelve tag values present in set of global tags - S. As per the equation, we use training data to compute $P(w_i|s_i)$ and for each word, the value for $P(s_i|s_{i-1})$ is computed based on the value of the tag of the previous word. For the first word of the sentence, we just consider the prior probability P(Tag s_i being the tag of the first word of the sentence).

III. Once this Posterior Distribution is computed, we use a random number generator to generate a random number and map it to the Cumulative Probability Densities of each tag per the computed Posterior Distribution. Thus, we obtain the desired value of tag s_i given word w_i.

IV. We now proceed to repeat the same set of steps for sampling from the posterior distribution for the next word till the end of the sentence.

V. This process is repeated for each sentence till we complete the burn-in period.

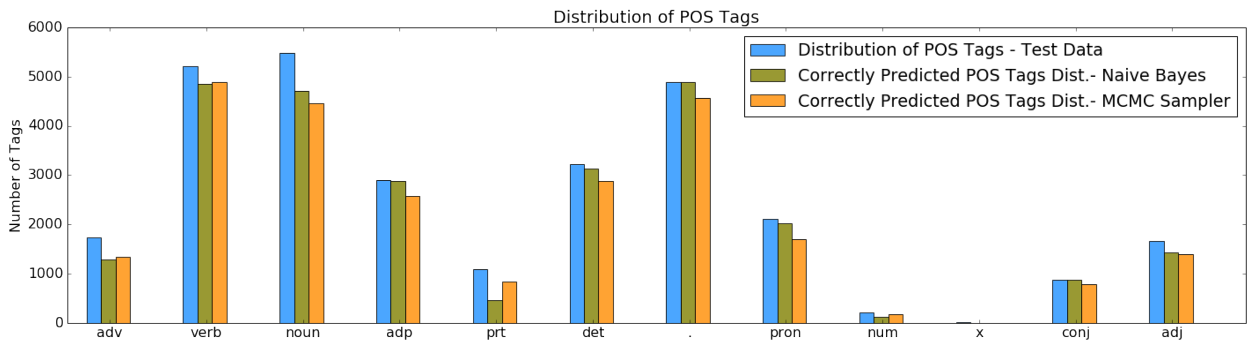
VI. The accuracy of predictions for each sentence is now computed and then we proceed to sample the POS tags for the next sentence.

3.3 RESULTS OF IMPLEMENTING POS-TAGGING FOR BOTH APPROACHES:

The results for Naive Bayes Classification and using Gibbs Sampling based MCMC classification are as follows:

#	APPROACH	PRIORS	ACCURACY on TEST DATA	
			Correct POS -Tags Predicted	Correct Sentences Predicted
1	Naive Bayes Classification: Use Posterior to find the most Probable POS-Tag for a given word using equation - iii	Informed (fig-a)	90.54%	33.15%
		Uninformed (fig-b)	91.29%	36.67%
2	Gibbs Sampling: Sample POS-Tag for a given word from the Posterior represented by equation - v	Informed (fig-c)	86.92%	4.25%
		Uninformed (fig-d)	72.45%	1.35%
Total			29442 Words	2000 Sentences

Visualizing Results: Comparison of number of correctly predicted words and Tags from Test Data.



4. IMPLICATIONS OF THIS STUDY:

- For Parts of Speech Tagging, Baseline accuracy is 90%. This is obtained by tagging every word with its most frequent Part-of-Speech and tagging unknown words by Nouns.
- As seen from our results, Naive Bayes Classification gives results which are slightly better than the baseline accuracy. Whereas the results given by the sampling approach are not as good.

- A possible way to increase accuracy is how we treat unknown words in the Test data, i.e. how to predict for a word not observed in the Training Data. Using some rudimentary techniques, we can show that accuracy of classification can shoot up to:

    - 93-94% for Naive Bayes Classification
    - 89-90% for Gibbs Sampling based on MCMC approach

5. FURTHER QUESTIONS RAISED BY THIS STUDY:

- While in case of both the approaches used, training requires a trivial amount of time, from Equation-V, we can observe that given a set of K tags and N words, we will have to evaluate K^N sequences to use sampling in order to predict the tags. This will be inefficient as the amount of data goes on increasing. We can explore more efficient ways of solving the same problem which are based on the Hidden Markov Models, for e.g. Viterbi Decoding.

- A possible way to increase the accuracy of prediction using the MCMC based Gibbs Sampling approach is to apply a Maximum A Posteriori (MAP) approach to the output of the sampled POS-Tag prediction.

- Currently in the Full-Bayesian Network approach we are considering a bi-gram Markov Model. Alternatively, we can also explore if we can use an n-gram model and how this approach will impact accuracy.

- In case of an Uninformed Prior using Naive-Bayes Classification, we see that the accuracy is actually higher. The cause behind this needs to be investigated.