Assignment 2, Part 2 for CSCI B551

Saurabh Mathur, Shivam Rastogi, Virendra Wali

October 26, 2018

Naive Bayes

Pre-processing

We applied minimal pre-processing so as to preserve the original words used in the tweets. Specifically, we applied the following transformations:

1. Convert the tweet text to lowercase.
2. Remove URLs. Most URLs occured only once.
3. Remove special characters and unicode symbols. Most of the tweets seem to be in English. So, we removed punctuation marks like comma and full stop and only kept words which were sequences of alphabets, numbers and some symbols like (').

Training the classifier

Our naive bayes classifier models two things - the distribution of locations and the distribution of words.

1. The prior on locations models the distribution of locations and is computed from the training data as

$$P(Location = l_i) = \frac{\#\ of\ tweets\ having\ their\ location\ as\ l_i}{total\ \#\ of\ tweets}$$

2. The distribution of words associated with each location is modeled as a bayesian unigram model with a multinomial distribution and a dirichlet prior. Prediction is done using the predictive distribution method. So, the likelihood is given by

$$P(w_1, w_2, \ldots, w_K | Location = l_i) = \prod_{k=1}^{K} \frac{m_{ik} + \alpha_{ik}}{N_i + \alpha_{i0}}$$

Where,
m_k is defined as

$$m_{ik} = \#\ of\ times\ kth\ word\ occurs\ in\ tweets\ from\ location\ l_i$$

N_i is defined as

$$N_i = the\ total\ \#\ of\ words\ in\ tweets\ from\ location\ l_i$$

and, α_i is the parameter of the dirichlet prior. It is a vector having one entry for each unique word in tweets for location l_i. α_{ik} is the entry of α_i corresponding to the kth word. α_{i0} is defined as

$$\alpha_{i0} = \sum_{a \in \alpha_i} a$$

Predicting a label

For a given set of words $W = \{w_1, w_2, \ldots, w_K\}$, the naive bayes prediction is given as

$$Prediction = \underset{i}{\operatorname{argmax}}\, P(l_i|W)$$

where the posterior $P(l_i|W)$ is estimated as

$$P(l_i|W) \sim \prod_{k=1}^{K} P(w_k|l_i)P(l_i)$$

Implementation details

1. For simplicity, we have set each α_{ik} corresponding to each word in each location as the same value (say a). This value was selected by selecting a value (from a set of values) that maximized the sum of evidence values of all locations. The simplified evidence function for a location l_i can be given as

$$Evidence = \frac{\Gamma(N_i a) \prod_k \Gamma(a + m_{ki})}{\Gamma(N_i a + N_i) \prod_k \Gamma(a)}$$

2. As the number of words in training set increases, the probability computations yield increasingly small values. This causes loss of precision. So, we used log Probabilities and log Evidence instead of the actual values. Since log is a monotonically increasing function, this transformation would not have an impact on the results.

Future Work

1. Instead of using a multinomial distribution, the distribution of words can be modeled using a bernoulli distribution. In such an implementation we would estimate the probability of each word in the vocabulary occurring and the word not occurring.

2. Instead of using raw counts for each word, normalized counts such as by tf-idf(term frequency-inverse document frequency) can be used.

3. Separate α values should be maintained for each location, since the distribution of words would vary according to the location.

4. The prior on locations can be computed from outside the training data (For ex. derived from the relative twitter activity for the cities). This would have a regularizing effect on the model.

References

1. Evidence maximization in bayesian unigram modeling was taught in Prof. Khardon's Machine Learning class.

2. Pattern Recognition and Machine Learning by Christopher M. Bishop.