CS6370 Programming Assignment - 1

Sabarinath N P (CS10B020)

Introduction

The objective of the assignment can broadly divided into two folds:

- **Parameter Estimation**: In this section, the concept parameter estimation is studied under different scenarios in unison with the famous inference problem, deducing the nature of messages as spam or ham. The different cases experimented are,
  - Maximum Likelihood Estimation(MLE) using distributions $\mathcal{L} \sim Multinomial(n_1, n_2, .., n_k, N)$ and $\mathcal{L} \sim Bernoulli(i, p)$ as likelihood probabilities.
  - Bayesian Estimation(BE) using distributions $p \sim Dir(\vec{\alpha})$ and $p \sim Beta(\alpha, \beta)$ as prior beliefs.
- **Sampling Methods**: Here, we delve into different methods to sampling from any arbitrary distribution. The two main methods explored are *inverse-cdf* and *rejective* sampling.

Data Set Processing

The given data set contains emails with subject and body. Each word is represented by numbers. This essentially cuts down the possibility of removing stop words for the purpose of feature selection. The main characteristics of the data set are

- 1099 emails in total divided into 10 folders
- Vocabulary of size 24654 words
- Number of ham emails is greater than number of spams but the difference is small enough to avoid any sampling (over/under) techniques.

In each iteration of cross validation, different folders are combined together to form training and test set. But the fact all emails of a given folder are chosen together either in train or test set, inspires us to use the following datastructure so as to reduce the space and time complexity.

- For Mulitnomial likelihood, we need number of times a given word occurs as spam and ham. We store it as count_multi[f][w][c] which gives number of times a given word w occurs in c class message in the folder f.
- For Bernoulli likelihood, we need number of spam or ham emails in which a given word occurs. We store it as count_bern[f][w][c] which gives number of emails of class c belonging to the folder f contains the word w.

Owing to small size of the data set, the above storage is feasible. The entire data set is processed only once for the entire assignment and the data structures generated are used for all the queries.

Theoritical Formulation

MLE

According to Bayes' rule,

$$p(\theta/\mathcal{X}) = \frac{p(\mathcal{X}/\theta)p(\theta)}{p(\mathcal{X})}$$

where $p(\mathcal{X}/\theta)$ is the likelihood and $p(\theta)$ is the prior.

Now, we can derive the MLE parameter θ using the following equation,

$$\hat{\theta}_{ML} = \underset{\theta}{\operatorname{argmax}}\, \mathcal{L}(\theta/\mathcal{X}) = \underset{\theta}{\operatorname{argmax}} \sum_{x \in \mathcal{X}} log\ p(x/\theta)$$

To estimate the parameter, we just equate the first derivate of $\mathcal{L}$ to zero.

$$\frac{\partial \mathcal{L}(\theta/\mathcal{X})}{\partial \theta} = 0$$

After smoothing the parameters take the following form, which cannot be considered as a pure MLE anymore.

$$\hat{p}_{ML} = \begin{cases} \frac{n_{ct}+1}{N_c+2} & \text{if } \mathcal{L} \sim \textit{Bernoulli} \\ \\ \frac{T_{ct}}{T_c+|V|} & \text{if } \mathcal{L} \sim \textit{Multinomial} \end{cases}$$

where n_{ct} is the number of emails of class c containing the word t. N_c is the number of emails of class c. T_{ct} is the number of times the word t occurs in emails of class c. we can defind T_c as $\sum_{t'} T_{ct'}$. $|V|$ is the size of the vocabulary.

MAP

Note that while doing ML estimate we assumed uniform prior over the parameter p. But if we assume a prior with hyper parameters $\vec{\alpha}$, it becomes MAP estimate

$$\hat{\theta}_{MAP} = \underset{\theta}{\text{argmax}}\, \mathcal{L}(\theta/\mathcal{X})p(\theta) = \underset{\theta}{\text{argmax}}\{\sum_{x\in\mathcal{X}} \textit{log } p(x/\theta) + \textit{log } p(\theta)\}$$

By equating the derivative of posterior to zero, we get the following MAP estimates for the parameter p given the hyper parameters $\vec{\alpha}$

$$\hat{p}_{MAP} = \begin{cases} \frac{n_{ct}+\alpha}{N_c+\alpha+\beta} & \text{if } p \sim \textit{Beta} \\ \\ \frac{T_{ct}+\alpha_c}{T_c+\sum_{i\in C}\alpha_i} & \text{if } p \sim \textit{Dirichlet} \end{cases}$$

The above MAP is for the $p(x/c)$. Using similar arguments we can derive the MAP estimate of $p(c)$ in terms of number of spam and ham emails and its hyper parameters $\vec{\alpha'}$. With the MAP estimate of $p(c)$ and $p(x/c)$ we can obtain the inference $p(c/\mathcal{X})$.

Experiments

Spam Filter

Two independent experiments are carried out to analyze MLE and MAP estimates. In both the cases 5-fold cross validation is performed over 5 different combinations of the given data set folders. For the case of *Multinomial* likelihood we obtain the following values,

Fold	Accuracy	Precision	Recall	F-measure
0-7 Train 8-9 Test	0.968	0.950	0.979	0.944
2-9 Train 0-1 Test	**0.981**	**0.969**	**0.989**	**0.979**
4-1 Train 2-3 Test	0.954	0.930	0.968	0.948
6-3 Train 4-5 Test	0.958	0.957	0.947	0.952
8-5 Train 6-7 Test	0.968	0.949	0.979	0.964

For the case of *Bernoulli* likelihood we obtain the following values,

Fold	Accuracy	Precision	Recall	F-measure
0-7 Train 8-9 Test	0.900	0.987	0.783	0.873
2-9 Train 0-1 Test	**0.940**	**0.988**	**0.875**	**0.928**
4-1 Train 2-3 Test	0.940	1.000	0.864	0.927
6-3 Train 4-5 Test	0.899	0.986	0.781	0.872
8-5 Train 6-7 Test	**0.940**	**0.977**	**0.885**	**0.928**

As expected, multinomial likelihood performs better than bernoulli likelihood in inferencing. This is primarily attributed by the fact that bernoulli distribution does not distinguish whether a word occurred once or more than once in a spam or ham email.

When considering the MAP estimates, we get the following results for different hyper parameters $\vec{\alpha}$. Let α and β be the hyper parameters of $p(c)$ and α' and β' be the hyper parameters of $p(x/c)$. Experiments are run for different values of hyper parameters and the best values in-terms of f-measure is reported in the following tables. For *Dirichlet* prior, we get the following values when $\alpha' = 5$ and $\beta' = 5$ and the performance is independent of hyper parameters of $p(c)$ given α' and β',

Fold	**Accuracy**	**Precision**	**Recall**	**F-measure**
0-7 Train 8-9 Test	0.927	0.955	0.876	0.913
2-9 Train 0-1 Test	**0.968**	**0.989**	**0.937**	**0.962**
4-1 Train 2-3 Test	0.936	0.976	0.875	0.923
6-3 Train 4-5 Test	0.890	0.986	0.760	0.858
8-5 Train 6-7 Test	0.936	1.000	0.854	0.921

For *Beta* prior, we get the following values when $\alpha^{'} = 1$ and $\beta^{'} = 5$ and the performance is independent of hyper parameters of $p(c)$ given $\alpha^{'}$ and $\beta^{'}$,

Fold	**Accuracy**	**Precision**	**Recall**	**F-measure**
0-7 Train 8-9 Test	0.656	0.560	1.000	0.718
2-9 Train 0-1 Test	0.657	0.561	1.000	0.719
4-1 Train 2-3 Test	0.627	0.539	1.000	0.700
6-3 Train 4-5 Test	**0.694**	**0.588**	**1.000**	**0.741**
8-5 Train 6-7 Test	0.659	0.561	1.000	0.719

When experimenting by fixing the $\beta^{'}$ and varying $\alpha^{'}$ values, it is observed that when dirichlet prior is employed, fmeasure increases with decrease in the difference between $\alpha^{'}$ and $\beta^{'}$. Similary when beta prior is used, fmeasure increases with decrease in $\alpha^{'}$ for given $\beta^{'}$.

The performance of the above two estimates (MLE and MAP) is compared with the performance of Naive Bayes Classifier of WEKA. Input to WEKA is provided in the required *arff* format for each of the 5 different cases. The performance of weka classifier does not match with my inferencing mechanism in following cases. The table below provides the mismatched values obtained by the WEKA Naive Bayes classifier.

Fold	**Accuracy**	**Precision**	**Recall**	**F-measure**
$\mathcal{L} \sim$ *Bernoulli*: 0-7 Train 8-9 Test	0.895	0.986	0.773	0.867
$\mathcal{L} \sim$ *Bernoulli*: 8-5 Train 6-7 Test	0.931	0.965	0.875	0.918
$\mathcal{L} \sim$ *Multinomial*: 2-9 Train 0-1 Test	0.977	0.959	0.989	0.974

PR-Curve

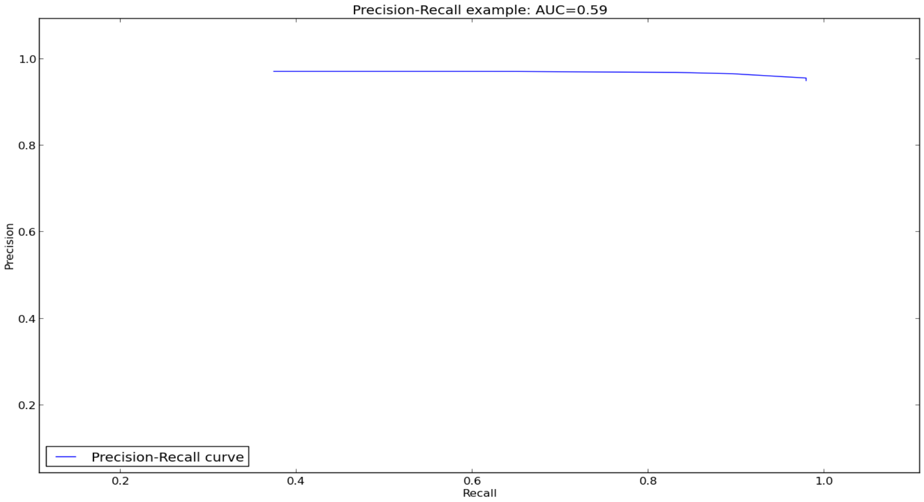
The best model based on the above experiments in the MLE model using Multinomial likelihood on the Train set 2-9 and Test set 0-1. The PR-Curve for the same using different threshold is given below. The formula used for checking for spam is

$$score_{spam} - score_{ham} \geq |score_{spam}| * threshold$$

where *threshold* is varied between 0 and 1.

Sampling

In the section, two types of sampling mechanisms, namely inverse-cdf and rejective sampling, are experimented with different types of probability density functions. There are no conclusive results to report in this regard.



Conclusion

In this assignment we experimented and understood how parameter estimation works using bayesian spam filter. We have analyzed the performance of spam filter by varying the train and test set for both MLE and MAP and by varying the hyper parameters in the case of MAP. We observed that Multinomial likelihood performed better than Bernoulli in the case of MLE, as expected. Though in general, MAP performs better than MLE, it depends on the hyper parameters heavily as observed from the experiments.