Sentimental analysis for yelp dataset

Bhaskar Jupudi NJUPUDI@UCSC.EDU
Trivikram Bollempalli TBOLLEMP@UCSC.EDU
Chandrahas CJAGADIS@UCSC.EDU
Karthikeyan KARTHIK@UCSC.EDU
GitHub Link HTTPS://GITHUB.COM/JUPUDIBHASKAR967/CMPS-242-PROJECT.GIT

Abstract

In this project, we aim to perform sentiment analysis i.e., classifying whether the review is postive or negative using the yelp dataset based on reviews and ratings. The classification problem can be solved by a set of algorithms. Every algorithm has its own advantages and disadvantages in terms of accuracy and model complexity.

For example, Naive Bayes classifier is faster to compute than Logistic Regression classifier for huge datasets. But the disadvantage with the former is that it assumes that features are independent where as the latter has no such assumptions which can lead to better precdiction. Our work mainly concentrates on implementing these two classifiers and techniques to make them perform much better. We have adopted multi-processing for feature extraction to make it way faster and also implemented two different approaches of Logistic regression for both binary and multi-class classfication. We have also implemented Naive Bayes classifier. Finally, we contrast these two algorithms based on time taken for execution and performance metrics like accuracy, precision and recall.

1. Problem statement

2. Feature Extraction

We have extracted the features for respective algorithms using our own constraints and formulations instead of using count vectorizer. The reason we designed this because countvectorizer method is raising a run-time exception for a large dataset (>300k reviews).

,

2.1. Binary classification

For logistic regression, we need to form a data vector from a bag of words which contains the word count. To compute this, first we extracted 4/5th of our total dataset (which is our training data) and extracted the words and their respective counts into a dictionary. Out of all these words, we have used three constraints to restrict the number of bag of words. One, if a word occurs for 'x' times in a positive review then we considered that word into bag of words only if the same word occurs for less than 4/10th times in the negative review and vice-versa. Second, total word count for a specific word should be more than 14 (arbitrary choice and can vary to adjust the accuracy). Third, we check if length of the word is more than 2 characters thereby restricting the number of features to 8895. The intuition behind this is,

2.2. Multiclass classification

3. Model Formulation

From Feature Engineering, we get input data and its labels. we will use this to train our logistic regression algorithm. We have implemented both logistic regression for two classes and also multi-class logistic regression.

3.1. Binary Logistic Regression

The probability of a class in logistic regression with two classes is given by:

We estimated the parameters using Maximum Log Likelihood estimation. For N data points, the likelihood function is :

Taking log on both sides and negation of it, we get the below negative log likelihood function:

E(w) is our cost function which we want to minimize to estimate W parameters. The gradient for the cost function is given by:

3.2. Multiclass Logistic regression

The probability of a class in logistic regression with multiple classes is given by: p (Ck —x) = yk () = exp (ak) P j exp (aj) where ak = ln p (x—Ck) + ln p (Ck) We estimated the parameters using Maximum Log Likelihood estimation. For N data points, the likelihood function is : = Y N n=1 Y K k=1 y tnk nk Taking log on both sides and negation of it, we get the below negative log likelihood function: X N n=1 X K k=1 tnk ln ynk E(w) is our cost function which we want to minimize to estimate W parameters. The gradient for the cost function is given by: E (w1, . . . , wK) = X N n=1 (ynj tnj) n We use this gradient in minimizing the cost function.

3.3. Optimization techniques

4. Evaluations

We performed all our experiments on a server that has 24 physical cores (with hyperthreading 2) and 128GB of DRAM.

5. Results

5.1. Effect of parallelism

Table 1. Execution time for extraction of features in Logistic regression classifier.

PARALLELISM	SIZE	FEATURES	TIME
NO	100K	9049	65M36.271S
NO	50K	5323	18M32.441S
YES	100K	9049	7M8.291S
YES	50K	5323	2M36.947S

Table 2. Parallelism vs countvectorizer()

METHOD	FEATURES	TIME
PARALLEL	17083	42M27.394S
COUNTVECTORIZER	10K	RUN-TIME ERROR

Table 3. Performance analysis of LR and NB for binary classification

CLASSIFIER	FEATURES	TIME	ACCURACY	PRECISION	RECALL
LR WITH GRADIENT	9049	102M39.110S	84.81	FILL	FILL
LR WITH SGD	13084	13M30.578S	86.87	[89.41, 80.76]	[91.75, 76.11]
NAIVE BAYES	199118	1M34.385S	82.98	[86.80, 73.96]	[88.73, 70.34]

Table 4. Performance analysis of LR and NB for multiclass classification

CLASSIFIER	FEATURES	TIME	ACCURACY	PRECISION	RECALL
LR WITH GRADIENT	–	102M39.110S	84.81	FILL	FILL
LR WITH SGD	–	13M30.578S	86.87	[89.41, 80.76]	[91.75, 76.11]
NAIVE BAYES	219383	3M29.505S	60.60	[58.5, 33.9, 34.2, 44.6, 75.6]	[77.8, 10.1, 17.2, 55.3, 74.3]

References

Langley, P. Crafting papers on machine learning. In Langley, Pat (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.