A Sentiment Quantification Model using Gaussian Mixture Model and K-means Clustering Techniques

Smita Suresh Daniel [#1], Dr. Ani Thomas [#2], Dr. Neelam Sahu[#3]

[#,1,3]*Dept of Computer Science ,Dr. C.V. Raman University , Kota ,C.G ,India*

[#,2] *Dept of Information Technology , BIT Durg , Bhilai, C.G, India*

Abstract— **Quantifying sentiments of various aspects of a product is important in understanding the emotions of reviewers towards a product . Sentiment Analysis and Quantification Model (SAQM) is an unsupervised model which accurately clusters product reviews to estimate the prevalence of various sentiment classes for each aspect of a product in a review dataset using Gaussian Mixture Model and K-means clustering methods, the results of which are better than the existing Quantification methods.**

Keywords— **Quantification, Sentiment Analysis , Gaussian Mixture Model, Unsupervised model.**

I. INTRODUCTION

Sentiment Quantification is a new research topic in Machine Learning. While in sentiment classification we are interested in obtaining the class of individual observations, in quantification we focus on estimating the total number of instances that belong to each class. This minor difference has led to the development of several algorithms that corrects the consistent errors issued by a classifier [6] . For sentiment quantification it is very important to correctly estimate proportions of each sentiment expressed in the set of documents (quantification task) [7] than to accurately estimate sentiment of a particular document (classification).
Generally a classifier, trained on a dataset does not provide the true a priori probabilities of the target classes on real-world data as shown in figure 1.1 .

.

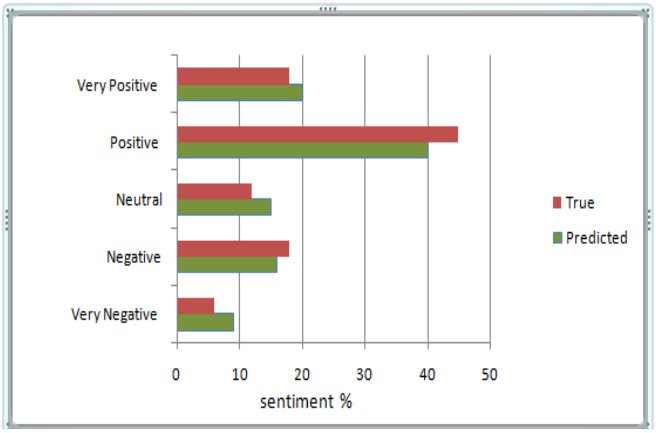

Figure 1.1 The difference between True and Predicted probabilities by a Classifier

This may result in poor quantification accuracy on the real-world dataset, as the classifier's decisions are based on the a posteriori probabilities of class membership, as they rely on the a priori probabilities of the training set [8]. Hence the outputs of the classifiers are to be corrected according to the new conditions [1][5].

Classifying and Counting and then adjusting the priory seems to be a practical solution for Quantification however research have shown that such a method generally produces poor quantification performances, it underestimates or over estimates the class prevalence as shown in [2] . Unsupervised model for sentiment classification is described in [9]. In this section a relatively new method to analyze and quantify sentiments based on Gaussian Mixture Model is formulated.

The main aim of this paper is to make more accurate predictions, in spite of inaccurate a priori probability estimations by a classifier . Here we present an unsupervised procedure to adjust the outputs provided by the K-means clustering model with respect to a new priori probabilities using Expectation Maximization(EM).

This iterative algorithm uses Gaussian Mixture Model(GMM) to maximizes the likelihood of the new data to a cluster . A statistical test is applied in order to find if the a priori class probabilities have changed from the training set to the real-word data after each iteration.

II. PROPOSED MODEL : SENTIMENT ANALYSIS AND QUANTIFICATION MODEL (SAQM)

Sentiment Analysis and Quantification Model (SAQM) is an analytical and an Unsupervised process which provides relevant information relative to the sentiment patterns and trend correlations to assist the company to rapidly change and improve their products by finding hidden truths.

This model uses Gaussian Mixture Model (GMM) algorithm to quantify things via the Gaussian probability density function (normal distribution curve). Gaussian Process Regression is used to describe the distribution of probability space. That is, each sentiment category is represented by a Gaussian Process, the weighted combination of all Gaussian Processes constitute GMM.

GMM is dependent on each sample for sentiment Quantification, the gradual iterative Estimation Maximization (EM) algorithm is implied to obtain optimal parameters, which increase the generalization ability of data processing and reduce time complexity of the method. In general, the model consists of three components:

1) GMM : The core algorithm of the model to perform unsupervised classification of multiclass sentiment;
2) EM : EM solves the selection problem of the initial value of GMM and obtains the finest parameters for it;
3) K-Means : This algorithm solves and provides the initial values of the selection problem of EM, regarding it as initial algorithm to obtain clustering numbers of EM algorithm.

K-means clustering is one of the methods of cluster analysis, the goal of which is to divide n observations in a dataset into K clusters in which each observation belongs to a cluster using the nearest means. Each cluster forms a circular shape and has a centroid which is updated iteratively using the mean distance value. When the data points of the observations are not within the circular shape, K-means fails to find the right clusters for it. The major drawbacks of this algorithm is :

1) The number of clusters K has to be provided .
2) It is incapable of handling noisy data and outliers.
3) It not suitable for determining clusters which are not with in the circular shape .

Accordingly, we need to allocate clusters to these data points in a different way. So we use a statistical distribution model as a replacement for distance based model which adjusts the a posteriori probabilities provided by K means to new a priori class probability using GMM.

Initialization of K-Means

In this model, initially we apply K-means algorithm with a known vector (measure of different sentiment classes in a text) as initial mean value of each cluster. These were taken from the reviews whose sentiment values are known. Amazon reviews provides star rating from 1 to 5 ie emotions from very negative to very positive . From this , a rough estimate is sufficient to initialize the parameters of K-Means for each K sentiment cluster . K-means then uses the Euclidian distance formula to cluster the vectorised dataset into K number of clusters to calculate the initial priory of each cluster.

We know that there are certain number of Gaussian distributions in a mixture , and each of these distributions form a cluster. The data points belonging to a single distribution are grouped together. Let's say we have three Gaussian distributions GD_1, GD_2, and GD_3 for three sentiments like positive, negative , neutral points . These distributions have a certain mean (μ_1, μ_2, μ_3) and variance (σ_1, σ_2, σ_3) value respectively. For a given set of data points, GMM would identify the probability of each data point belonging to each of these distributions. The data points are then assigned to the closest centroid and a cluster is formed. The centroids are then updated and the data points are reassigned. This process goes on iteratively until the location of centroids no longer changes.

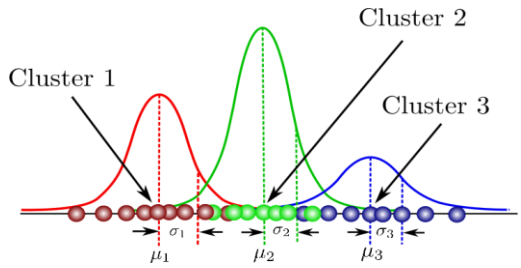

Fig 2.1 The Three Sentiment Clusters

This procedure also uses the Expectation Maximization algorithm. The posteriori probabilities of various classes provided by the K-means model are close to the observed probabilities for a dataset.

The Basic Concepts

Let D be the dataset that stores a large number of Amazon reviews , represented by D = {r_1, r_2, … r_i… r_n},

where r_i are the reviews . The number of reviews is defined as |D|. Each Gaussian distribution is called a Gaussian process which represent different sentiment class and the probability density function of GMM. We use GMM to construct the multi-class sentiment quantification model SAQM. Following are the modules prepared for sentiment quantification of D :

i Preprocessing and Detecting Review Sentiments

Each review contains casual languages, sarcasm, misspellings, excitements and are ambiguous so the dataset is cleaned, spellchecked and pre-processed and tokenized. Parts of Speech tagging is performed to reduce the original attribute set by removing irrelevant features with less frequencies. Also most of the adjectives, adverbs, and a small set of nouns and verbs can acquire semantic orientation for sentiments and they are to be preserved.

ii Categorization - Aspect Based

We need to find out the most frequently talked about aspect features of a product in the review-sets and categorize them. Attributes or aspects of a product are generally represented as nouns and extracting nouns and their relative frequencies in the dataset provide an insight about the various attributes of the product. We reduced the original attribute set by removing irrelevant features with less frequencies. The feature selection method computes a score for each individual aspect and then select the top ranked features as per that score using Term Frequency/Inverse Document Frequency. Reviews are categorized based on it.

iii Calculation of Sentiment Feature

Polarity of each word in a review is measured and a collective polarity is considered for each sentence. Mostly adjectives, adverb, and a small set of nouns and verbs can acquire semantic orientation for sentiments . Handling of Negation in Text**,** Negation identification and its scope within a piece of text are required in finding out the sentiments.

iv Creating Polarity List using Adjectives, Adverbs and Verbs as Sentiment Words

A polarity list also known as lexicon dictionary stores Adjectives, Verbs and Adverbs which generally contains sentiment words and are collected in text file using a POS(Parts of Speech) tagger and scores between 0 and ±2 are marked to them depending up on their intensity . Adverbs of degree that modify adjectives are also scored according to its intensity. Thus each vector has the sentiment values as (VeryNegative, negative, neutral ,.positive, VeryPositive and more).

Let m represents the number of multi-class sentiments in the dictionary. A review $r_i$contains p number of sentiments in it. We calculate the intensity and polarity to compute a feature value of each sentiment; Let V_{nm} be the sum ofintensity and polarity of sentiment, which denotes the feature value of sentiment m of a review r_n.

$V_{nm} = \alpha \times intensity_i + \beta \times polarity_i$

Where α, β denote adjustment coefficients, different values reflect the influence degree of intensity and polarity on V_{nm} ; $intensity_i$denotes intensity value of sentiment word i, and $polarity_i$denotes polarity value of sentiment word $i$in review.

v Sentiment Vector Representation

Feature matrix of sentiments is obtained as follow:

$$Vnm = \begin{bmatrix} V_{11} & V_{12} & .... & V_{1m-1} & V_{1m} \\ . & . & .... & . & . \\ . & . & .... & . & . \\ V_{n1} & V_{n2} & .... & V_{nm-1} & V_{nm} \end{bmatrix}$$

vi Dimension Reduction of Sentiment Feature Matrix

A dimension reduction algorithm is used to reduce the dimension of $V_{nm}$from m to t. A low-dimensional sentiment feature matrix $V_{nt}$is as follows:

$$Vnt = \begin{bmatrix} V_{11} & V_{12} & .... & V_{1t-1} & V_{1t} \\ . & . & .... & . & . \\ . & . & .... & . & . \\ V_{n1} & V_{n2} & .... & Vnt-1 & V_{nt} \end{bmatrix}$$

vii Gaussian Process Regression

For training dataset D = $\{(x_i, y_i)\}$= (X, Y), where $x_i \in Rd$, X = $[x_1, x_2, \cdots x_n]$ denotes the vector matrix of d × n dimension, and $y_i \in R$is the output result. A set of random variables $\{f(x_1), f(x_2), \cdots f(x_n)\}$ can be formed by a given set X, which satisfy a joint Gaussian distribution), so the Gaussian process can be denoted by mean $m(x)$ and covariance function k(x,x') is :.

f(x) ~ G($m(x)$, k(x, x')) $= E[f(x)]$ cov(f(x),f(x')) $= k(x, x') = E[(f(x) - m(x)) - (f(x') - m(x'))]$

In GMM, each cluster corresponds to a probability distribution or Gaussian distribution. What we want to do is to learn the parameters of these distributions, which is the Gaussian's mean μ (mu), and the variance σ^2 (sigma Σ) .The mathematical form of the mulitvariate Gaussian distribution can be written as where $N(x \mid \mu, \Sigma)$.

Each Gaussian $N(x_i, \mu_c, \Sigma_k)$ is described as $\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma c|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x_i-\mu_c)^T|\Sigma c|^{-1}(x_i-\mu_c))$

- This is also referred to as the probability density function (pdf).
- Gaussian distribution is commonly referred to as the normal distribution, hence that's where the *N* comes from.
- *x* is the random observation for the distribution.
- The mean μ, controls the gaussian's "center position" and the variance σ^2, controls its "shape". To be precise, it is actually the standard deviation , i.e. the square root of the variance that controls the distribution's shape.

viii Gaussian Mixture Model of Sentiment

Each sentiment can be denoted as a function of GMM, that is,the linear combination of multiple GMMs compose the dataset.The dataset x is the linear combination of the independent GMM of all components $G = \{N_1, N_2, \cdots, N_K\}$, and $\pi_1, \pi_2, \cdots, \pi_k$ refer to the weight of all variables. Therefore, the probability density function of x can be represented as:

$p(x|\theta) = \sum_{k=1}^{K} \pi_k N(x_k \mid \mu_k, \Sigma_k)$

$\sum_{k=1}^{K} \pi_k = 1$ and $\theta = \{\pi_k, \mu_c, \Sigma_k\}$

Where N(·) denotes the probability density function of Gaussian Mixture Model; π_k is the weight for the Gaussian Process k.

The Algorithm : SAQM

Input : Training Sentiment Data $D_{Train} = \{r_1, r_2 \ldots\ldots r_n\}$
Testing Sentiment Data $D_{Test} = \{r_1, r_2 \ldots\ldots r_m\}$
Output : Quantification result of sentiment samples $R_{Test} = \{Q_1, Q_2 \ldots\ldots Q_m\}$

1. $D = \{r_1, r_2, \ldots\ldots r_n\}$ // Preprocess ,Reduce , calculate , sequence sentiments
2. $D'_{train} = VEC(D_{train})$ // Vectorize training data
3. $D'_{test} = VEC(D_{test})$ // Vectorization test data
4. begin Initialization θ^0, th, $t \leftarrow 0$ // Initialization
5. SAMQ_initials = K_means(D'_{train}) // K-means initialization of model with initial known mean value of each cluster
6. do $t \leftarrow t+1$
7. $J(\theta, \theta') = \theta(D_{train}, SAQM\_initials)$ // E-step of EM algorithm
8. $\theta^{t+1} \leftarrow$ Max $J(\theta, \theta')$ // M-step of EM algorithm
9. until $J(\theta^{t+1}, \theta') - J(\theta^t, \theta^{t-1}) <= th$ // Iteration ends
10. return $\hat{\theta} \leftarrow \theta^{t+1}$ // $\theta = \{\pi_i, \mu_i, \Sigma_i\}$
11. End
12. SAQM = GMM($D_{train}, \hat{\theta}$) // GMM construction
13. For i = 1 to m // Testing Samples
14. R = Quantify(D'_{test}, SAQM) // Quantify
15. Output(R_{test}) // Output results
16. End For

Output of SAQM Model After Clustering

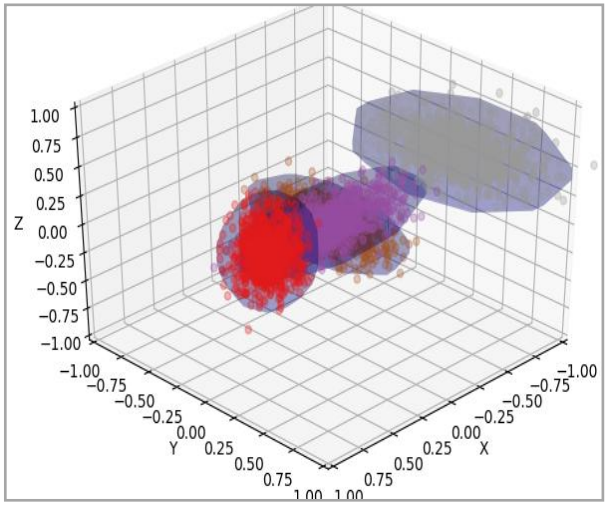

FIGURE 2.2 3D PATTERN OF SENTIMENT CLUSTERS

III EXPERIMENTAL EVALUATION OF SAQM

The traditional standard Quantification method Probabilistic Adjusted Classify and Count PACC [3] is used to compare and evaluate the standards of SAQM in the experiments. The SVM classifier is used for a priori probability estimation of true Positive, Negative and Neutral rates of a classifier using the confusion matrix. The difference between the training and the test ie, $\hat{p}(\delta_i|\omega_j)$, is the probability δi to classify an observation in class ω_i, wrongly as ω_j.. $\hat{p}_t(\delta_i|\omega_j)$ is estimated on the training set using Confusion Matrix . It is used to estimate the priori probabilities on a new data set using *n* linear equations

$\hat{p}(\delta_i) = \sum_{j=1}^{n} \hat{p}_t(\delta_i|\omega_j \hat{p}(\omega_j)$

Where i = 1, . . ., *n*

with respect to the $\hat{p}(\omega_j)$, where the $\hat{p}(\delta_i)$ is the difference. After computing $p(\omega_j)$, adjusting the priory to get the new a posteriori probabilities can be done by PACC. [3] .

For all experimentation Amazon reviews of Kindle Tablets of different Versions are used datasets are collected from www.kaggle.com (Kindle_Oasis, 'Kindle_fire1' and 'Kindle_pw1 ') .

i. Estimates Before and After Adjustments

Table 3.1 shows the Quantification rates before and after the probability adjustments by SAQM as well as PACC method when the true priors of the test set $p(\omega_1)$, which are unknown were used to adjust the classifier's outputs.

Table 3.1 Quantification Rates Before and After adjustments.

True priors$p(\omega_1)$	Percentage of Correct Quantification			
	No Adjustment using		After Adjustment by using	
	SVM	K-Means	SAQM	PACC
10%	90.1%	86.4%	93.2%	92.4%
20%	90.3%	87.2%	91.5%	91.2%
30%	88.6%	87.4%	89.3%	89.1%
40%	90.4%	86.4%	90.5%	90.7%
50%	87.0%	87.2%	87.6%	86.5%
60%	90.0%	87.4%	91.2%	90.3%
70%	89.2%	86.4%	89.5%	89.2%
80%	89.5%	87.2%	90 .6%	90.1%
90%	88.5%	87.4%	91.4%	91.7%

By looking at Table 3.1, we observe how adjustments made on the outputs of initial classifiers(K-means and SVM) results . In spite of the low classification accuracy by K-means , SAQM quantification accuracy is significant.

ii Robustness Evaluation of SAQM on Amazon Data

In this experiment a classifier's imperfect estimation of the a posteriori probability and the size of the training and the test dataset is used to estimate the new a priori probabilities are studied . Decreasing the size of the training dataset reduced the quality of classifier output. At the same time, the size of the test dataset was also reduced in order to study the effect of reducing the amount of data available to the SAQM and the SVM algorithms.

For each state we compared the outputs of (SVM based quantifier PACC) and (K-means based quantifier SAQM) on a known distribution. Here we could quantify the deviation with reference to the true a posteriori and evaluate the effect of decreasing the size of the training and test dataset on the a priori estimates .

Here the SVM classifier was trained on training set ($p_t(\omega_1) = 0.33 = p_t(\omega_2)) = p_t(\omega_3)$). An unequal test set (with $p(\omega_1) = 0.20$, $p(\omega_2) = 0.40$) and $p(\omega_3) = 0.40$) was scored by it. Now the size of training and test set were decreased to study the score obtained as shown in Table 3.2.

The score ($s(\mathbf{x}) = \hat{p}_t(\omega_1|\mathbf{x})$ obtained by SVM classifier and ($k(\mathbf{x}) = p_t(\omega_1|\mathbf{x})$) obtained by K-means on the test datasets before their outputs are adjusted are measured to compute the

absolute deviation $|k(\mathbf{x}) - s(\mathbf{x})|$).

Then, for each test set, the SAQM and the confusion matrix adjustment procedures(PACC) were applied to the outputs of the K-Means and SVM classifiers to estimate the new a priori probabilities and compare it with the true priory .

Table 3.2 : Average Results for the Estimation of the Priors, using Training and Test sets of Different Sizes

Training set Size (#ω1,#ω2, #ω3)	Test set Size (#ω1,#ω2, #ω3)	Mean absolute deviation $\lvert k(\mathbf{x}_n) - s(\mathbf{x}_n)\rvert$	Estimated prior for ω1 ($p(\omega1) = 0.20$) by using	
			SAQM	PACC
(500, 500,500)	(200, 400,400)	0.107	22.0%	24.7%
	(100, 200,200)	0.110	21.6%	24.5%
	(40, 80,80)	0.104	20.4%	23.5%
	(20, 40,40)	0.122	26.7%	22.7%
(250, 250,250)	(200, 400,400)	0.139	22.0%	24.7%
	(100, 200,200)	0.140	21.6%	24.5%
	(40, 80,80)	0.134	20.4%	23.5%
	(20, 40,40)	0.167	22.7%	26.7%
(100, 100,100)	(200, 400,400)	0.183	24.1%	27.5%
	(100, 200,200)	0.185	24.4%	28.2%
	(40, 80,80)	0.181	23.5%	27.3%
	(20, 40,40)	0.180	26.6%	29.2%
(50, 50,50)	(200, 400,400)	0.202	24.9%	28.5%
	(100, 200,200)	0.199	25.3%	29.0%
	(40, 80,80)	0.203	24.3%	27.6%
	(20, 40,40)	0.189	22.3%	26.0%

The deviations as seen in Table 3.2 comprehends that decreasing the size of training dataset degrades the estimation of the a posteriori probabilities (an increase of absolute deviation) of about 0.10 between large, i.e. N_t = 1500, and small, i.e. N_t = 150, training dataset sizes). Of course, the prior estimates degraded accordingly, but only slightly. SAQM seems to be more robust than PACC (confusion matrix method) as it overestimated the priory ($p(\omega_1)$) by 4%, while the PACC method overestimated by 6.2 %. In fact decreasing the test set size had no much effect on the results.

iv Comparison of Quantification Rates of SAQM and Other Models

Figure 3.3 shows the Quantification rates of different quantifiers before and after the output adjustments . It also illustrates the degradation in quantifier estimations due to the decrease in the size of the training datasets . we can notice a difference of about 15% between large, i.e. N_t = 1500, and small, i.e. N_t = 150, training dataset sizes).

For this experiment, two basic assumptions which were considered :

1. The a predicted probabilities of each sentiment classes provided by the K-Means classifierwere adequately close observed probabilities.
2. The training dataset were selected such that the within class probability weights do not change.

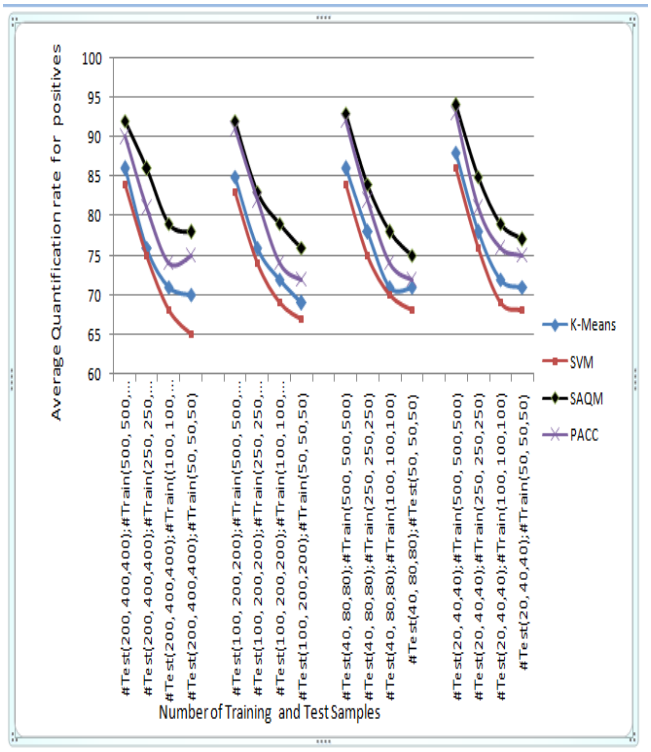

Figure 3.3. Comparison of Quantification Rates

In Figure 3.3 Quantification rates are obtained on the Kindle Fire dataset. Results are estimated for four different conditions:

1. Without adjusting the SVM classifier output ;
2. Adjusting the SVM classifier priory by using the PACC method
3. Output by using Kmeans Clustering algorithm.
4. Adjusting the K-Means clustering algorithm (i.e. After adjustment by GMM); output by SAQM.

The results are plotted for different sizes of both the training and the test set

The Quantification results of SAQM is better than the other methods even though there is not much difference . ie 0.9 % approximatly

.

v. Effect of Increasing the Number of Training Data on the SAQMApproach.

To study the effect of increasing the number of training data on SAQM we gradually increased the training data from 200 to 2000 and evaluated using Kindle_fire dataset and calculated the recall and precision on all classes. We obtain an average F-measure percentage of quantification with a different number of training samples, which is shown in Figure 3.4. and found that the training data of minimum 500 samples are enough for the model to properly quantify the data into three clusters. The larger value of samples means more features can be selected with a higher fitting value but the curve flattens after 800 samples. In Figure 3.4, F-measure increases to 82%, the F-measure achieves the highest point with 800 samples. In concise, the select 500 training samples are sufficient to train the proposed model.

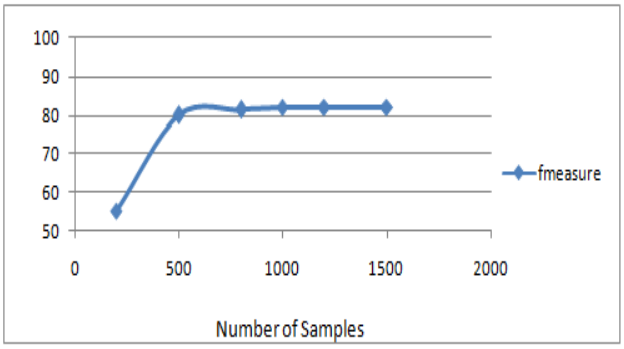

Figure 3.4 . Effect of the number of training samples on F-measure

vi Gradual Optimization Process of the SAQM Approach.

To verify that the gradual optimization of the SAQM approach is effective with multi-class classification, the results of precision and recall before and after optimization is shown in Figure 3.6 and 3.5 respectively. Recall is the measure of the sensitivity or completeness of a quantifier model. Higher recall values has less number of false classes. Bur lower recall values have more false classes.

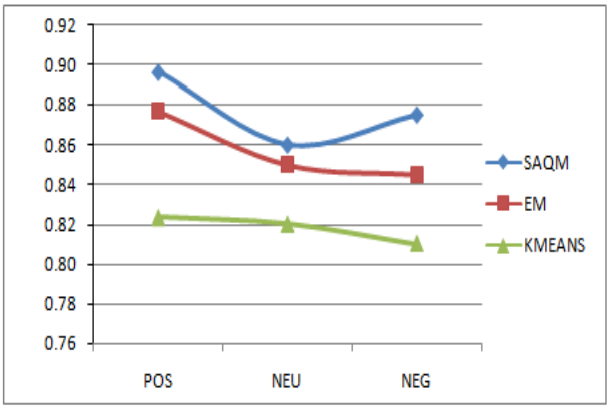

Figure 3.5 . Recall of Different Methods in the Process of SAQM

As seen in Figures 3.5 and 3.6, the recall of a positive class is generally high while precision of neutral class is relatively low. The reason may be attributed to misclassifying neutral class. In other word, the methodology is failing to identify neutral sentiment.

The Accuracy of SAQM = 88 %

The SAQM algorithm realize simple clustering by inputting the number of cluster as K values to K-means

model, then the EM algorithm obtain optimal parameters of the clusters by iterative processing, which is used by GMM to provide the best results.

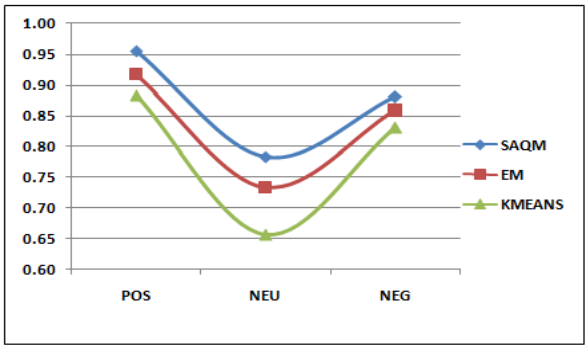

Figure 3.6 . Precision of Different Methods in the Process of SAQM

The results suggest that proposed methodology is better than the previous methodologies. Combining Kmeans and GMM resulted in higher precision, recall and F-measure values.

IV . OBSERVATIONS AND ANALYSIS

Performances of various sentiment classification and quantification approaches were tested on text and their analysis based on the study is mentioned below .

i SAQM Tests on Amazon Data

Three Amazon data sets were used to study the difference between priory estimation by initial classifiers and their output readjustments by SAQM and PACC quantifiers. A training set constituted of 200 reviews of each sentiment class (labelled) was selected in each data set and used for training SVM as in [5] and the same set (unlabelled) were fed to K-Means. To widen the difference between the class distributions in the training (0.33, 0.33, 0.33) and the test sets, we obtained a class distribution of ($p(\omega1) = 0.20$, $p(\omega2) + p(\omega3) = 0.80$) for the test set by adding and dropping some of the reviews . Two different training and test dataset were created, and different tests were executed for each training dataset. Average quantification rate for positive sentiments in each condition is observed .

Table 4.1 : Quantification Results on Real Datasets

Dataset	True priors	Priors estimated by		Correct Estimation	
				After adjustments by using	
		K-Means	SVM	SAQM	PACC
Kindle_fire1	20%	24%	23%	78.0%	74%
Kindle_pw1	20%	19%	26%	90.6%	92%
Kindle_Oasis	20%	22%	21%	75.4%	73.2%

Table 4.1 shows the a priori probabilities estimated by means of the SVM and the K-Means as well as the quantification rates before and after the probability adjustments.

These results show that the prior estimates of SVM were generally better than the K-means (except for 'Kindle _Oasis'). However , these priors were better adjusted by SAQM even when K-means gave less accuracy.

V. APPLICATIONS OF SAQM

i. SAQM For Problem Extraction

To detect the sentiment issues of reviews related to various aspects of the product Kindle fire for the Kindle_fire dataset we categorized five most prominent aspects into five different files after preprocessing and using Term Frequency weights to indicate the relative importance of aspects extracted. Based on these five categories and the relative positive proportions from each of the categories were compared using the two approaches for quantification. Sentiment Quantification on these files resulted in graphical representation in Figure 5.1.

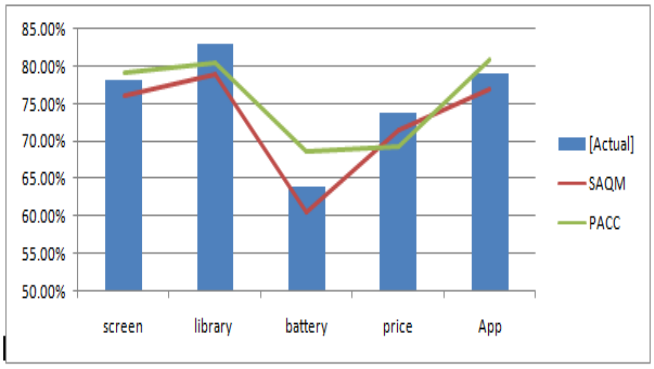

Figure 5.1 . Quantification by Different Methods

Analysing Figure 5.1 it is found that the Battery aspect of the Kindle is showing less positive sentiments. So an indepth analysis was done on this category to study the reason behind why many people are opposing it. On Quantifying this particular category again in the same process we get the results in figure 5.2 and 5.3.

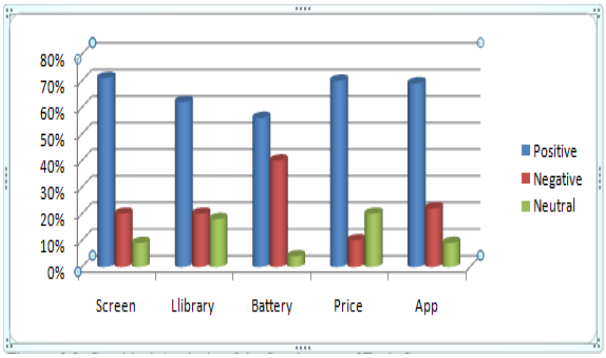

Figure. 5.2 Graphical Analysis of the Sentiments of Each Category

Here we come to the conclusion that reviewers are referring to the battery replacement and battery getting heated during their usage. Similarly Sentiment Analysis can be done for all other categories also.

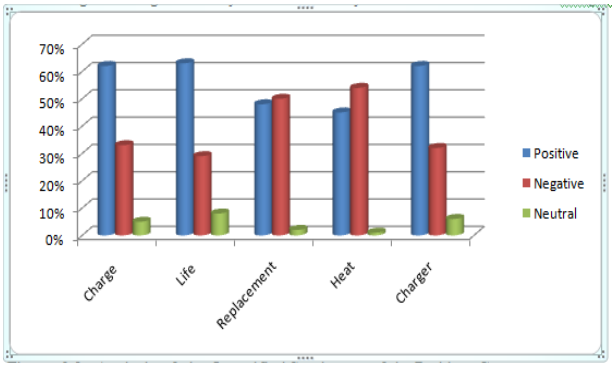

Figure.5.3 Analysis of the Quantified Sentiments of the Problem Category

The calculated mean error = 2.8 %

ii Using SAQM for Multi Class Text Quantification.

For detailed quantification of sentiments in a dataset it is necessary to choose a dataset with a variety of sentiments, we used Kindle_Kids data set containing 800 sample reviews to be clustered into 5 emotions as Neutral, Happy, Good, Sad and Angry. . Table 5.1 shows a brief summary of its sentiments.

Table 5.1 The Summary of Emotion Categories of Kindle_Kids dataset

Emotion	No	Happy	Good	Sad	Angry
Quantity	14	343	252	170	21

Calculating the error estimation for n samples as Mean Squared Error (MSE) which is the difference between actual and estimated quantification.

$$\text{MSE} = \frac{1}{K}\sum_{i=1}^{k}(x_i - t_i)^2 = 12.6$$

where K is the number emotions (clusters) i.e. 5, x_i is the estimated sentiment score of each category and t_i is the real score derived from human evaluation.

iii Tracking Sentiments Over Time:

Detecting sentiments towards an aspect of a product and tracking its development during a period of time is a Quantification application. The below graph shows tracking sentiments towards two aspects, that is Price and Color-Adjustable features for the Amazon Product Kindle_Oasis from Jan 2019 to June 2019. It is very important for companies to track and identify the sentiment fluctuations for their products, so as to take immediate actions in case of any negative sentiment emergence. Sentiment detection over time has been applied using SAQM and the result is shown in Figure 5.4 .

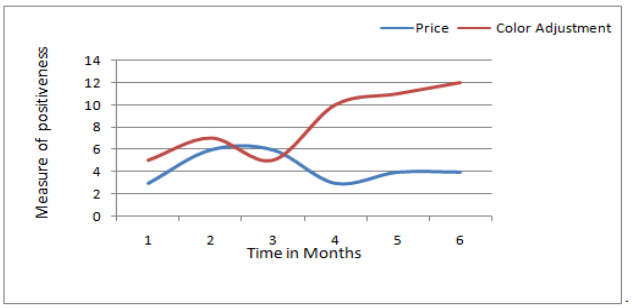

Figure 5.4 Tracking Sentiments Over time

It is obvious from the graph that that the color-adjustable feature is having a high response while the price rise was not at all appreciated by the customer reviews .

VI. RESULTS AND DISCUSSIONS

i. Statistical Significance Test

A pair wise two tailed Wilcoxon signed-ranks [10] is performed on nine Amazon datasets using KLD measures . It is a non-parametric alternative to the paired t test, where Wilcoxon test calculates the absolute differences in the performance measures of two algorithms for each dataset and then ranks the positive and the negative differences after comparing them.

If the computed p value is less than the significance level $\alpha = 0.05$, the null hypothesis is rejected else the difference of the two algorithms is statistically significant.

Table 6.1 Results of Wilcoxon RankedTests

	CC	PCC	ACC	PACC	SAQM
CC	-	<	≈	<	<
PCC	>	-	>	<	<
ACC	>	<	-	<	<=
PACC	>	<=	>	-	≈
SAQM	>	>	>=	≈	-

'>' sign means that the algorithm on the row is statistically significant and is betterthan the one on the column; '<' means the less significant ; '= ' means there is both are statistically same ; '_' not applicable.

From the results of Wilcoxon test in Table 6.1 we can comprehend the difference between various quantification algorithms [1] from a statistical point of view.

ii. Complexity of SAQM :

SAQM is a combination of two algorithms ie. K-Means and GMM. Calculation of complexity consist of the following variables. Let n be number of samples, k is the

number of Clusters and m is the number of sentiment classes in a review .

Time-Complexity :For every iteration of K- means there are:

- Calculation of distances : To calculate the distance from a point to the centroid, squared Euclidean distance is taken. So two subtractions, one summation, two multiplications and one square - root operations which equals 6-operations are needed .
- Comparisons between distances
- Calculation of centroids (cluster centre-points)

Therefore, for every iteration,

Total no. of operations = $6[I * k * m * n] + [I * (k-1) * m * n] + [I * k * ((m-1) + 1) * n]$

If the algorithm converges in I iterations then complexity is calculated as $O(I * k * m * n)$.

For large data-sets where k << m & n << m, the complexity is approximately O(m)

Since the initial parameters are obtained by K-means algorithm , so it takes a maximum of only 7 to 8 iterations by EM to converge. The number of iteration depends on the starting parameters supplied to the K-Means. m is cubic in the number of dimensions because there is a matrix inversion step while computing the E-step. Specifically, while calculating the density of each point we need to invert the covariance matrix The complexity of GMM algorithm is $O (i*n*k*m^3)$ where n is the number of samples, k is the number of Gaussian Clusters and m is the Problem Dimension , i the number of iteration the EM algorithm is using to converge. Thus the time complexity of SAQM is $O(i * n * k * m^3)$

The time taken to process n=2000 samples with k = 3 and m= 5 approximately 1.6 minutes for SAQM.

Space-Complexity :To store the data points and centroids the complexity is $O((m+k) * n)$. Where n is number of samples, k is the number of Clusters and m is the number of sentiment classes in a review .

CONCLUSION

A generative and unsupervised quantification methodology of multi-class sentiment called SAQM was formulated. GMM clusters it into distinct Gaussian components, which further realize effective and accurate quantification of multi-class sentiment. This methodology take advantage of generative approach that does not need to set up any parameters and labelled training dataset except for intial random means for clusters .

The EM combined with GMM algorithm obtains the optimal parameters for the SAQM methodology through progressive iteration and self updation process. Thus the SAMQ approach has low time complexity and high sentiment quantification accuracy.

Although the SAQM approach solves the limitation of existing methods, there are still some drawbacks: the computation complexity of the model is increased as dataset becomes larger, thereby resulting in convergence rate instability.

SAQM successfully detects most important aspects of the product about which the customers have referred in their text reviews and quantifies its sentiments into many categories, thus mining the important information contained in it. It depicts the actual problem faced by the customers correctly.

Future studies have numerous avenues to explore like improving the stability of SAQM convergence rate with big data and then further improve its accuracy. Also, more research is required about relevant features of the sentiment samples with in-depth data analysis from different backgrounds and sources also dealing with Sarcasm in reviews and anaphora Issues .

References

[1] Daniel Suresh Smita , Thomas Ani, Sahu Neelam (2019), "Sentiment Quantification Approaches For Customer Reviews Of Amazon Products " , International Journal of Advanced Science and Technology – Engineering Research Support Society- SERSC Australia , Vol. 28 No 11 (2019) **ISSN:** 2005-4238 (Print) **ISSN:** 2207-6360 (Online)

[2] Gao W, Sebastiani F (2015) Tweet sentiment: from classification to quantification. In: Proceedings of the 7th international conference on advances in social network analysis and mining (ASONAM 2015), Paris, France. 97–104.

[3] Gao W , Sebastiani F (2016) ," Tweet Sentiment: From Classification to Quantification", Springer-Verlag Wien .

[4] Forman G (2008) Quantifying counts and costs via classification. Data Min Knowl Discov 17(2) :164–206 .

[5] Saerens M, Latinne P, Decaestecker C (2002) Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. Neural Comput 14(1) : 21–41.

[6] Pang B. , Lee L. (2008) , Opinion mining and sentiment analysis , Found . Trends Information Retrival , vol. 2, no.(1-2), : 1-135.

[7] Barranquero J, Dı´ez J, del Coz JJ (2015) .Quantification-oriented learning based on reliable classifiers. Pattern Recognition. 48(2) : 591–604.

[8] Bella A, Ferri C, Herna´ndez-Orallo J, Ramu´rez-Quintana MJ (2010) Quantification via probability estimators. In: Proceedings of the 11th IEEE international conference on data mining (ICDM 2010), Sydney, AU : 737–74

[9] Xu liwei, Oiu jiagnan ,Unsupervised multiclass Sentiment classification Approach,Article*in* Knowledge Organization· Jan 2019 DOI: 10.5771/0943-7444-2019-1-15

Wilcoxon F (1945) Individual comparisons by ranking methods.Biom Bull 1(6) :80–83.