

Opinion Mining of M Learning Reviews using Soft Computing Techniques

A. Nisha Jebaseeli
Assistant Professor
Dept. of Computer Science
BDU Constituent College Lalgudi, Trichy

E. Kirubakaran, PhD.
Additional General Manager
Outsourcing Department
Bharat Heavy Electricals Ltd. Trichy, India

ABSTRACT

Internet has increasingly become the place for online learning, and exchange of ideas. The rapid development in wireless technology offering fast data transfer has lead to mobile device revolution. With the ease of access of mobile devices like mobile phones, PDAs, tablet PCs and high bandwidth through wireless, there is an upsurge of mobile learning or M-learning. It is important to know the opinion of users using m-learning platforms for developing and fine tuning of M-learning systems. The sheer volume of reviews found in the internet blog spot, bulletin board makes it difficult to track and understand customer opinions. Opinion mining also known as sentiment mining is an area of research which attempts at determining the opinion underlying a text written in natural language which summarizes the customer reviews and express whether the opinions are positive or negative. In this paper, we investigate the classification accuracy of machine learning algorithms for opinion mining of M-learning system review.

Keywords

Opinion Mining, M-learning, e-learning, Neural Network

1. INTRODUCTION

The rapid development in wireless technology offering fast data transfer has lead to mobile device revolution. With the ease of access of mobile devices like mobile phones, PDAs, tablet PCs and high bandwidth through wireless, there is an upsurge of mobile learning or M-learning. mobile learning can be defined as 'The exploitation of ubiquitous handheld technologies, together with wireless and mobile phone networks, to facilitate, support, enhance and extend the reach of teaching and learning' [1]. M-learning is part of e-learning, but focuses on creating new contexts for learning through interaction between people, technologies and settings, and on learning with an increasingly mobile society [2]. M-learners learn through a combination of conversation on mobile device, retrieving information and exploring virtual worlds. Mobile learners are able to learn about any subject, at any place and just when the knowledge is required, thus, widening the access of learning. The growth in mobile technology has led to support of graphics, animations and better data management, enriching conventional lessons. New M-learning systems are designed based on game-based learning [3], interactive location based guides [4], and ambient learning [5]. Organizations are able to update its sales/service personnel who are always on the move through M-learning systems.

The m-Learning advantages comparing to e-Learning are:

- it is location and time independent;

- most of mobile devices cost less than desktop PCs;
- ensures students' engage in M-Learning as students use mobile devices in everyday life;
- Using GPS technology the m-Learning can provide location dependent education.

Opinion mining (OM) also known as sentiment mining, is an area of research which attempts at determining the opinion underlying a text written in natural language [6]. This discipline is a cross of information retrieval and computational linguistics. Opinion mining identifies and extracts subjective information from the source materials. Opinions expressed in a set of source documents about an object is mined using OM; by extracting attributes of the object from the comments and determine whether the opinions are positive, negative or neutral. Opinions are important for most, during decision making process [7]. With the explosion of the internet and web, it is possible to find out opinions from a very large group of people through discussion forum, blogs and other social networks.

The opinions could be on consumer goods, book, movie, political issue and so on. These opinions have huge influence in shaping the opinions of other consumers. Companies and organizations, monitor these opinions or sentiment to find out how the product/service is perceived. But monitoring the web is a difficult job as there are many sources which contain huge amount of data. Thus the need of a system which can automatically analyze opinions to understand how a product or service is perceived is in great demand. So OM is used on the vast data to perceive whether the opinions are positive or negative and also to summarize [8, 9].

Topic related words are the key to classify documents into predefined topic classes, e.g., sports, art etc in text classification. Whereas, in opinion classification, topic-related words are not very important but, opinion words indicating positivity or negativity of the opinions are important, e.g., great, excellent, amazing, horrible, bad, worst, etc. Most of the methodologies for opinion mining apply some forms of machine learning techniques for classification. In opinion classification, for a set of text data D , classifiers group documents in the set D expressing positive or negative opinion on a specific object. Algorithms specifically designed for opinion classification exploits opinion words and phrases together with some scoring functions [9].

In this paper, we investigate classification of opinion mining not only based on opinion words but also corpus words which are frequently used in the documents under review. The documents are then classified using a neural network. The proposed method is evaluated using IMDB

dataset and also to find opinion for m Learning reviews which are extracted from customer's feedback. This paper is organized into the following sections. Section II briefly describes the materials and methods and classification algorithms, section III describes the results obtained and discusses the same.

2. MATERIALS AND METHODS

The proposed classification uses online movie reviews as data due to the availability of a large number of reviews available online. Bo Pang and Lillian Lee [10] provide collections of movie-review documents labeled with respect to their overall sentiment polarity (positive or negative). A dataset consisting of m learning reviews is also compiled. The m learning reviews are sourced from the feedback of customers of m learning software available in the android market.

The first step involves constructing a table using all words found in the input documents, which includes indexing and counting of documents and words, i.e., a matrix of frequencies of words that specifies the number of times that word occurs in each document. This basic process was further refined by use of stop word list and stemming of words; to exclude certain common words such as "the" and "a" (stop word lists) and to combine different grammatical forms of the same words such as "traveling," "traveled," "travel," etc. To eliminate frequently used words and infrequently used words for the data under consideration, the following rules were applied.

- Words occurring in more than 80% of the documents were ignored
- Words occurring in less than 30% of the documents were ignored

Based on the above rule the word frequency is created using the exclude list.

The terms document frequency is computed. In a set of documents X and a set of terms a , each document can be modeled as a vector v in the a dimensional space R^a , this is a vector space model. Let the term frequency be denoted by $freq(x, a)$, this expresses the number of occurrence of term a in the document x . The term-frequency matrix $TF(x, a)$ measures the association of a term a with respect to the given document x . $TF(x, a)$ is assigned zero if the document does not contain the term, and a number otherwise. The number could be set as $TF(x, a) = 1$ when term a occurs in the document x or uses the relative term frequency. The **relative term frequency** is the term frequency versus the total number of occurrences of all the terms in the document.

The term frequency is generally normalized by:

$$TF(x, a) = \begin{cases} 0 & freq(x, a) = 0 \\ 1 + \log(1 + \log(freq(x, a))) & otherwise \end{cases} \quad (1)$$

Another measure used is the **Inverse Document Frequency (IDF)**, it represents the scaling factor. If term a occurs frequently in many documents, then its importance is scaled down due to its reduced discriminative power. The $IDF(a)$ is defined as follows:

$$IDF(a) = \log \frac{1 + |X|}{x_a} \quad (2)$$

x_a is the set of documents containing term a .

The numerical data so obtained from the above process for each word is used to train the classification algorithm. Classification based on Random forest, Learning Vector quantization and Elman neural network are investigated.

Random Forest

Random forest algorithm for classification uses group of classification trees [11], which is not as rational as decision tree. The concept of learning used in random forest is that the integration of multiple unstable classifiers will improve the performance of the final classifier giving better overall performance than an individual classifier [12]. Random forests are a combination of tree predictors, where each tree is got from the random vector values sampled independently with same distribution. As the number of trees increases the generalization error for forests converges to a limit; depending on the strength of individual trees and correlation among the trees [13]. Random vectors generated direct the growth of each tree in the forest. If δ_k is the random vector generated for the k th tree, which is independent of other random vectors $(\delta_1, \dots, \delta_{k-1})$; the tree is grown using the random vector δ_k and the training set. The classifier resulting from the vector is $h(x, \delta_k)$ where x is the input vector. So a random tree consists of set of trees with classifiers got from independent identically distributed random vectors; and each tree casts a vote for the class at input x , as the number of trees increase, the generalization error converges to

$$P_{X,Y} \left(P_\delta(h(X, \delta) = Y) - \max_{j \neq Y} P_\delta(h(X, \delta) = j) < 0 \right) \quad (3)$$

Where Y, X are random vectors and $P_{X,Y}$ is generalization error probability over the X, Y space.

To improve accuracy, the correlation has to be minimized by the randomness used while maintaining the strength. Each node is built into tree using randomly selected inputs. Random forest is an effective tool in prediction.

Learning Vector quantization (LVQ)

Learning vector Quantization (LVQ) is a neural net that combines competitive learning with supervision [14]. The basic LVQ approach is quite intuitive. It is based on a standard trained SOM with input vectors $\{x\}$ and weights/Voronoi vectors $\{w_j\}$. The new factor is that the input data points have associated class information. This allows us to use the known classification labels of the inputs to find the best classification label for each w_j , i.e. for each Voronoi cell. For example, by simply counting up the total number of instances of each class for the inputs within each cell. Then each new input without a class label can be assigned to the class of the Voronoi cell it falls within. The problem with this is that, in general, it is unlikely that the Voronoi cell boundaries will match up with the best possible classification

boundaries, so the classification generalization performance will not be as good as possible. The obvious solution is to shift the Voronoi cell boundaries so they better match the classification boundaries.

The basic LVQ algorithm is a straightforward method for shifting the Voronoi cell boundaries to result in better classification. It starts from the trained SOM with input vectors $\{\mathbf{x}\}$ and weights/Voronoi vectors $\{\mathbf{w}_j\}$, and uses the classification labels of the inputs to find the best classification label for each \mathbf{w}_j . The LVQ algorithm then checks the input classes against the Voronoi cell classes and moves the \mathbf{w}_j appropriately:

1. If the input \mathbf{x} and the associated Voronoi vector/weight \mathbf{w}_j (i.e. the weight of the winning output node $I(\mathbf{x})$) have the same class label, then move them closer together by $\Delta \mathbf{w}_j(\mathbf{x})(t) = \beta(t)(\mathbf{x} - \mathbf{w}_j(\mathbf{x})(t))$ as in the SOM algorithm.
2. If the input \mathbf{x} and associated Voronoi vector/weight \mathbf{w}_j have the different class labels, then move them apart by $\Delta \mathbf{w}_j(\mathbf{x})(t) = -\beta(t)(\mathbf{x} - \mathbf{w}_j(\mathbf{x})(t))$.
3. Voronoi vectors/weights \mathbf{w}_j corresponding to other input regions are left unchanged with $\Delta \mathbf{w}_j(t) = 0$.

where $\beta(t)$ is a learning rate that decreases with the number of iterations/epochs of training. In this way we get better classification than by the SOM alone.

Neural Network

Neural network are based on the principles of biological neural networks and can perform tasks that a linear program finds difficult. Neural networks can be considered as a system that processes an input data to provide an output by optimizing the learning rule. In a neural network, the desired or target response is set at the output and is used to compute the error from the difference of the actual output to the desired output. The error is fed back to the system so that the weights can be adjusted till the desired output is obtained. Advantages of neural networks are that it can be used on noisy data, able to classify patterns on which it has not been trained. Neural network algorithm can be used even when the relationship between attributes and classes are not well defined. There are many kinds of neural network algorithms, popularly used is the backpropagation algorithm.

The backpropagation algorithm uses a multilayer feed-forward neural network. A multilayer feed-forward neural network is made of three layers, input layer, hidden layer and output layer as shown in Figure 1. The input layer is made up of units with inputs to the network matching to the attributes measured for each learning document. These inputs are weighted and fed to the second layer, the hidden layer. The outputs of the hidden layer are input to the units in the output layer. The output layer finally gives out the network's prediction.

Each output unit takes a weighted sum of the outputs from units from the previous layer. A nonlinear function is applied to the weighted input. Multilayer feed-forward neural networks can also model the class prediction as a nonlinear combination. Backpropagation works by repeatedly processing a training set, comparing the network's prediction with the actual known target value. The target value is known as the class label of the training set. The weights are modified to minimize the error between the network's prediction and the actual target value. The modification is done in the backward direction that is from output layer to input layer. So when the weights converge, the learning process stops.

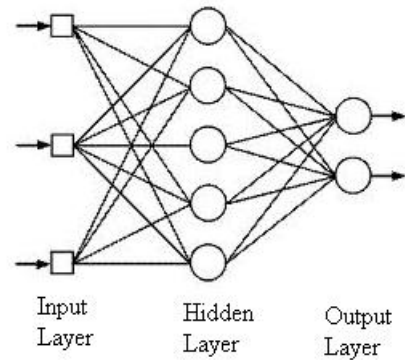


Fig. 1 A multilayer feed-forward neural network.

In Elman network positive feedback is used to construct memory in the network as shown in Fig. 2. The network has input hidden and output layers. Special units called context units save previous output values of hidden layer neurons. Context unit values are then fed back fully connected to hidden layer neurons and thus they serve as additional inputs to the network. Networks output layer values are not fed back to network. The Elman network has a high depth, low resolution memory, since the context units keep exponentially decreasing trace of past hidden neuron output values.

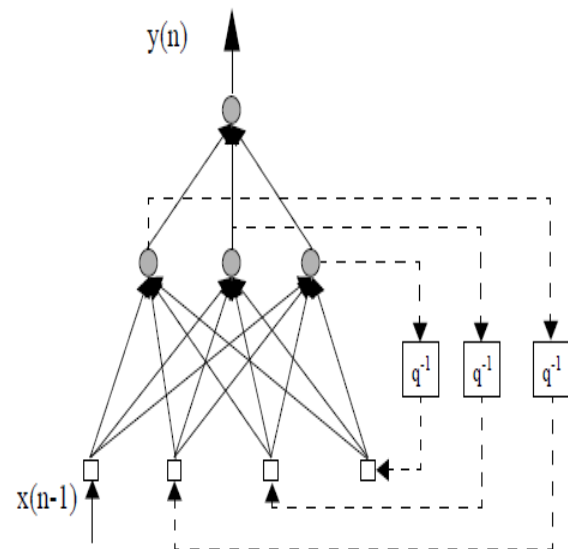


Figure 2: A Elman Neural Network with one input and one output.

3. RESULTS AND DISCUSSION

The classification accuracy of various Algorithms with respect to both the IMDB dataset and M- Learning dataset is shown in Table 1 and Figure 3.

Table I: Classification Accuracy

Algorithm used	Classification Accuracy %	
	IMDB Dataset	M-Learning Dataset
Random forest	68	65
LVQ	72	56.67
Proposed Elman Neural Network	88	76.67

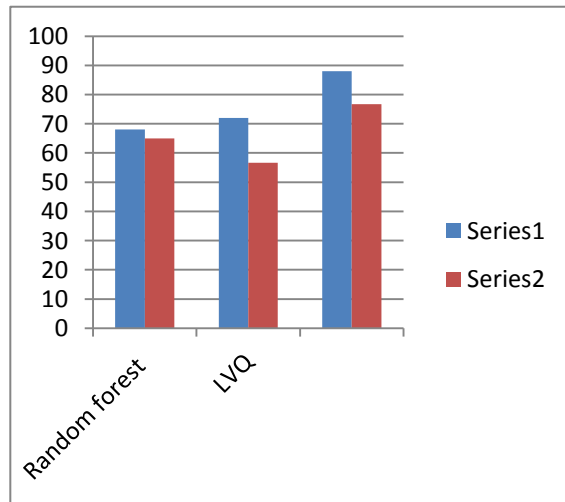


Figure 3: Classification Accuracy of various Algorithms.
(Series 1 represents the IMDB dataset results and Series 2 represents the M- Learning dataset results)

Table 2 shows the Precision, Recall and F Measure of the different algorithms used for both the dataset and Figure4 show the Precision and Recall and Figure5 shows the F Measure of the same.

Table 2: Precision, Recall and F Measure of the different Algorithms

		Algorithms used		
		Random Forest	LVQ	Elman
Precision	IMDB Dataset	0.687	0.823	0.88
	M learning Dataset	0.696	0.629	0.667

Recall	IMDB Dataset	0.68	0.72	0.88
	M learning Dataset	0.65	0.567	0.8
F Measure	IMDB Dataset	0.679	0.7	0.88
	M learning Dataset	0.648	0.574	0.727

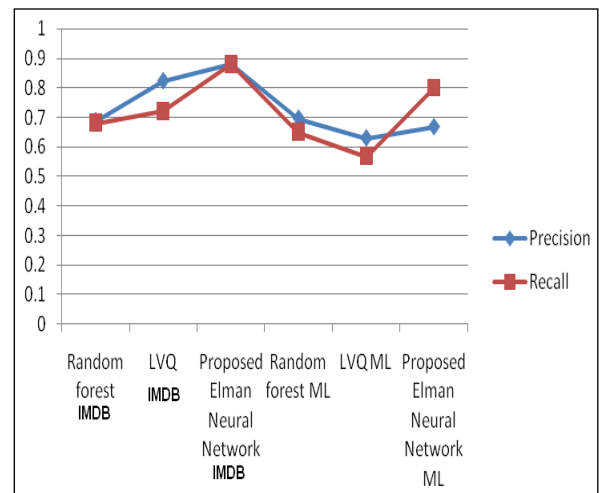


Figure 4: Precision and Recall of the different Algorithms

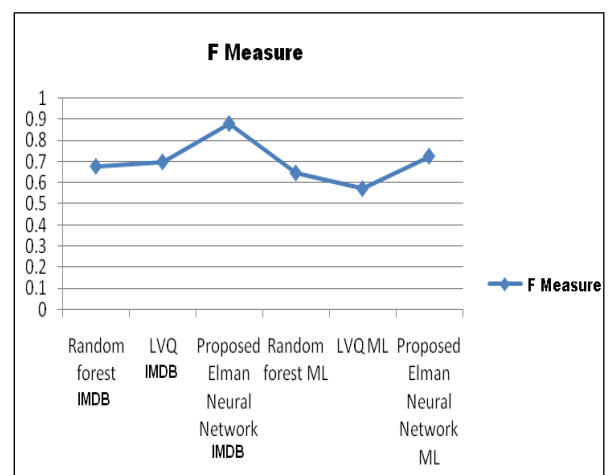


Figure 5: F Measure of the different Algorithms

4. CONCLUSION

In this paper, it was proposed to investigate the efficiency of Random Forest classifier, LVQ and proposed Elman Neural

Network to predict opinions as positive or negative specifically for M Learning systems and evaluated against IMDB dataset. Equal number of positive and negative opinions was obtained from [6]. A dataset consisting of M Learning reviews is also compiled. The M Learning reviews are sourced from the feedback of customers of M Learning software available in the android market. The data was preprocessed by removing commonly occurring words and rarely occurring words. Accuracies in the range of 76.67 to 88% were obtained.

5. REFERENCES

- [1] MoLeNET (2010). Modernising education and training. Mobilising technology for learning, Available at <https://crm.lsnlearning.org.uk/user/order.aspx?code=100103>
- [2] Sharples, M. (2009) Methods for Evaluating Mobile Learning. In G.N. Vavoula, N. Pachler, and A. Kukulska-Hulme (eds), *Researching Mobile Learning: Frameworks, Tools and Research Designs*. Oxford: Peter Lang Publishing Group, pp. 17-39.
- [3] Schwabe, G., & Goth, C. (2005). Mobile learning with a mobile game: design and motivational effects. *Journal of Computer Assisted Learning*, 21(3), 204-216.
- [4] Damala, A., & Lecoq, C. (2005). Mobivisit: Nomadic Computing in indoor cultural settings. A field study in the museum of Fine Arts, Lyon. In X. Perrot (Ed.), *ICHIM International Cultural Heritage Informatics Meeting September 21-23, 2005, Paris, France*
- [5] Rogers, Y., Price, S., Fitzpatrick, G., Fleck, R., Harris, E., Smith, H., et al. (2004, June 1-3). Ambient wood: designing new forms of digital augmentation for learning outdoors. Paper presented at the 2004 conference on Interaction design and children: building a community (IDC 2004), Maryland, USA.
- [6] Bing Liu . Exploring User Opinions in Recommender Systems. Proceeding of the second KDD workshop on Large Scale Recommender Systems and the Netflix Prize Competition, Aug 24, 2008, Las Vegas, Nevada, USA.
- [7] Peter D. Turney and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346.
- [8] Dave, D., Lawrence, A., and Pennock, D. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. *Proceedings of International World Wide Web Conference (WWW'03)*, 2003.
- [9] Turney, P. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *ACL'02*, 2002.
- [10] MovieReviewData <http://www.cs.cornell.edu/People/pabo/movie-review-data/>
- [11] L. Breiman, "Random forests," *Mach. Learning*, vol. 45, pp. 5–32, 2001
- [12] Y. Saeys, T. Abeel, and Y. Van de Peer, "Robust feature selection using ensemble feature selection techniques," in *Proc. ECML/PKDD, Part II (LNAI 5212)*, 2008, pp. 313–325.
- [13] Freund, Y. and Schapire, R. [1996] Experiments with a new boosting algorithm, *Machine Learning: Proceedings of the Thirteenth International Conference*, pp. 148-156.
- [14] M. T. Hagan, H. B. Demuth, and M. Beale, "Neural Networks Design", PWS Publishing, 1996