Volume 5, Issue 5, MAY 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Filtering the Walls in Online Social Networks

Nisy John Panicker*, Laya Devadas
Department of CSE, College of Engineering
Munnar, Kerala, India

Abstract— The growth of Online Social Networks (OSN) has embarked a new method of interaction and communication among the human race. Individuals can keep in touch with his/her friends by exchanging information which includes text, image, audio and video files. The wide and dynamic nature of these data creates the need of Information Filtering. In the current scenario, OSNs provide less support to prevent unwanted messages being posted on one's wall. In this work, a new method of filtering the walls in OSN is put forward, thus avoiding unwanted messages being displayed on the users' wall. Both textual messages and images (based on their title) are taken into consideration. For this, Machine learning-based text classification technique such as Naïve Bayes classifier is used.

Keywords— Online Social Networks, Information Filtering, Data Mining, Text Classification.

I. INTRODUCTION

A. Online Social Networks

Men are by nature social beings who depend on others to meet their daily needs. With the growth of Internet, Online Social Networks have become a popular communication media. Internet today is not just an information access tool; but also an interaction mechanism used by individuals to exchange contents, opinions and information [1]. A social networking service is a platform to build social networks or social relations among people who, share activities, interests, backgrounds or real-life matters. A social network service consists of a representation of each user (a profile), his social links, and some additional services. Social networking is a web-based service that allows individuals to create a profile which is public, to create a list of users with whom to share connection, and view the connections within the system [2]. Popular social networking sites are Facebook, Google Plus, and Twitter etc.

Online Social Networks has now become a part and parcel of human life. Tremendous data including text, image, audio, and video are sent daily across Online Social Networks. According to Facebook statistics[1] average user creates 90 pieces of content every month, whereas more than 30 billion pieces of content (web links, news stories, notes, photo albums, etc.) are shared every month. The huge and dynamic volume of these data creates the need of Information Filtering. In OSNs, information filtering can be used as there is a chance of posting unwanted messages on public/private area, called walls.

An information filtering system is a system that removes redundant or unwanted information from an information stream using (semi)automated or computerized methods prior to presentation to a human user. Its main aim is the management of the information overload and increment of the semantic signal-to-noise ratio. For this the user's profile is compared to some reference characteristics. The characteristics may originate from the information item (the content-based approach) or the user's social environment (the collaborative filtering approach) [3]. Thus information filtering can be used to give users the ability to automatically control the messages written on their own walls. Today only a very little support is provided by OSNs to prevent unwanted messages on user walls. For instance, Facebook allows individuals to state who is allowed to insert messages on their walls (i.e. friends, friends of friends, or defined group of friends). After all, no content-based preferences are supported and therefore it is not possible to prevent undesired messages, like political or vulgar ones, no matter of the user who posts those [4]. There are number of techniques for information filtering which come under Data Mining or Knowledge Discovery from Data (KDD).

B. Data Mining

Data mining refers to extracting or "mining" knowledge from large amounts of data. Data mining is an essential step in Knowledge Discovery. Knowledge Discovery consists of an iterative sequence of the following steps:

- Data cleaning (to remove noise and inconsistent data)
- Data integration (where multiple data sources may be combined)
- Data selection (where data relevant to the analysis task are retrieved from the database)
- Data transformation (where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations)
- Data mining (an essential process where intelligent methods are applied in order to extract data patterns)
- Pattern evaluation (to identify the truly interesting patterns representing knowledge based on some interestingness measures)

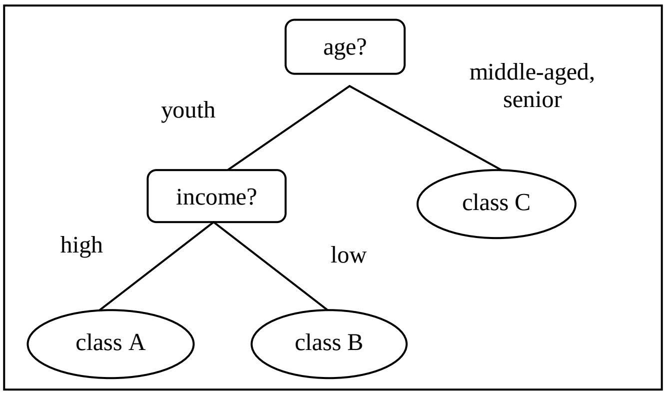
- Knowledge presentation (where visualization and knowledge representation techniques are used to present the mined knowledge to the user) [5].

1. http://www.facebook.com/press/info.php?statistics

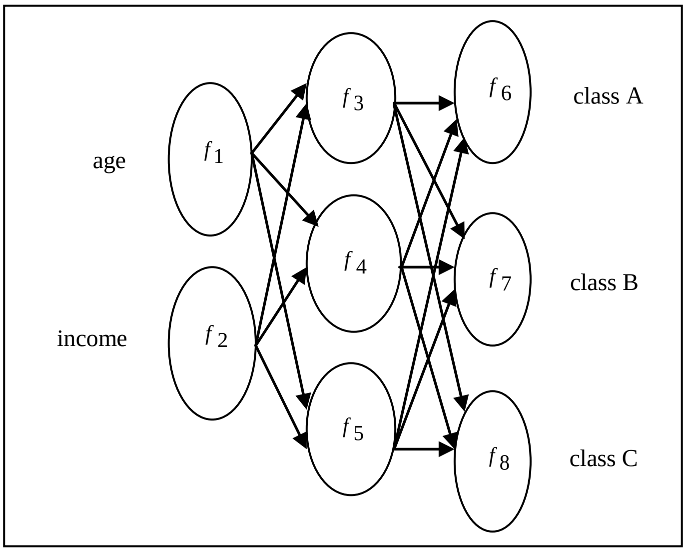
1) Classification: One of the functionalities provided by data mining is Classification. Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts for the purpose of being able to use the model to predict the class of objects whose class label is unknown. The derived model is based on the analysis of a set of training data (i.e. data objects whose class label is known). "How is the derived model presented?" The derived model may be represented in various forms, such as classification (IF-THEN) rules, decision trees, mathematical formulae, or neural networks (Fig. 1).

age(X, "youth") AND income(X, "high")	class(X, "A")
age(X, "youth") AND income(X, "low")	class(X, "B")
age(X, "middle-aged")	class(X, "C")
age(X, "senior")	class(X, "C")

(a)



(b)



(c)

Fig. 1 A classification model can be represented in various forms, such as (a) IF-THEN rules, (b) a decision tree, or a (c) neural network.

- A decision tree is a flow-chart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leaves represent classes or class distributions. Decision trees can easily be converted to classification rules.
- Support Vector Machines is a promising new method for the classification of both linear and nonlinear data.
- A neural network, when used for classification, is typically a collection of neuron-like processing units with weighted connections between the units.

- Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class. Bayesian classification is based on Bayes' theorem [5].

2) *How does Classification work?* Data Classification is a two-step process. In the first step, a classifier is built describing a predetermined set of data classes or concepts. This learning step (or training phase), where a classification algorithm builds the classifier by analyzing or "learning from" a training set made up of database tuples and their associated class labels. As the class label of each training tuple is provided this step is also known as supervised learning. In the second step, the model is used for classification. The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier.

II. RELATED WORKS

The main aim of this paper is the design of a system providing content-based message filtering for OSNs, using Machine Learning techniques. Related works in the field of content-based message filtering are as follows:

A. Adomavicius and G. Tuzhilin [6] surveyed on the field of recommender systems and described the current generation of recommendation methods that are usually classified into the following three main categories: content-based, collaborative, and hybrid recommendation approaches. In content-based recommendations, the user will be recommended items similar to the ones the user preferred in the past. The user will be recommended items that people with similar tastes and preferences liked in the past in collaborative recommendations. Hybrid approaches combine collaborative and content-based methods.

R.J. Mooney and L. Roy [7] proposed that Recommender systems improve access to relevant products and information by making personalized suggestions based on previous examples of a user's likes and dislikes. Most existing recommender systems use social filtering methods that base recommendations on other users' preferences. By contrast, content-based methods use information about an item itself to make suggestions. This approach has the advantage of being able to recommend previously unrated items to users with unique interests and to provide explanations for its recommendations. They described a content-based book recommending system that utilizes information extraction and a machine-learning algorithm for text categorization. LIBRA is an initial content-based book recommender which uses a simple Bayesian learning algorithm and information about books extracted from the web to recommend titles based on training examples supplied by an individual user. Initial experimental results demonstrate that this approach can produce accurate recommendations.

P.W. Foltz and S.T. Dumais [8] and P.S. Jacobs and L.F. Rau [9] proposed in content-based filtering, each user is assumed to operate independently. As a result, a content-based filtering system selects information items based on the correlation between the content of the items and the user preferences as opposed to a collaborative filtering system that chooses items based on the correlation between people with similar preferences.

M.J. Pazzani and D. Billsus [10] discussed algorithms for learning and revising user profiles that can determine which World Wide Web sites on a given topic would be interesting to a user. They described the use of a naive Bayesian classifier for this task, and demonstrated that it can incrementally learn profiles from user feedback on the interestingness of Web sites. Furthermore, the Bayesian classifier may easily be extended to revise user provided profiles. In an experimental evaluation they compared the Bayesian classifier to computationally more intensive alternatives, and showed that it performs at least as well as these approaches throughout a range of different domains. In addition, they analyzed the effects of providing the classifier with background knowledge in form of user defined profiles and examined the use of lexical knowledge for feature selection. They found that both approaches can substantially increase the prediction accuracy.

III. PROPOSED METHOD

The Machine learning technique used for filtering unwanted messages from OSN user walls is Naïve Bayes classifier. In machine learning, naive Bayes classifier is a probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes classifier is highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression [11], which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, in our case labels to the messages typed on user walls. The 'naïve Bayes' assumption states that the probability of each document is dependent on the document class but it's independent of the document's context and position. The probability of a document *d* being in a class *C* is computed using Bayes' theorem as:

$$P(C \mid d) \propto P(C) \prod_{1 \le k \le n_d} P(t_k \mid C) \qquad (1)$$

where n_d is the length of the document, $P(t_k \mid C)$ is the conditional probability of term t_k occurring in a document of class C, $P(C)$ is the prior probability of C.

The main goal of Naïve Bayes classification is to find the 'best' class. The best class is the most likely or maximum a posterior (MAP) class C_{map}, which is computed as:

$$c_{map} = \arg\max_{C \in K} P(C) \prod\nolimits_{1 \le k \le n_d} P(t_k \mid C) \qquad (2)$$

where K is the set of classes.

Multiplying lots of small probabilities can result in floating point underflow. Since *log(xy) = log(x) + log(y)*, it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities. As log is a monotonic function, the class with highest final log probability score is still the most probable. In practice, the classification rule C_{map} is computed as:

$$C_{map} = \arg\max_{C \in K} \left[\log P(C) + \sum_{1 \le k \le n_d} \log P(t_k \mid C) \right] \quad (3)$$

Each conditional parameter $\log P(t_k \mid C)$ is a weight that indicates how good an indicator t_k is for *C*. The prior $\log P(C)$ is a weight that indicates the relative frequency of *C*. The sum of log prior and term weights is then a measure of how much evidence there is for the document being in the class. The class with the most evidence is selected.

In this work, the messages typed on a person's wall are classified as either 'good' or 'spam' using the Naïve Bayes classifier. If the resulting class is good, then the message is displayed at the receiver's wall else it is blocked or filtered out from the public wall. The same technique is used for filtering images based on their titles.

A. Parameter Estimation

The prior $P(C)$ is computed as:

$$P(C) = \frac{N_C}{N} \quad (4)$$

where N_C is the number of documents in class *C* and *N* is the total number of documents.

The conditional probability $P(t_k \mid C)$ is computed as:

$$P(t_k \mid C) = \frac{T_{Ct}}{|V|} \quad (5)$$

where T_{Ct} is the number of occurrences of t_k in class *C* and *V* is the whole vocabulary.

B. Time Complexity of Naive Bayes

In training mode, the time complexity is $O(|D| L_{avg} + |C||V|)$ where *D* is the training set, L_{avg} is the average length of a training document, *C* is the set of classes and *V* is the vocabulary. $O(|D| L_{avg})$ is the time it takes to compute all counts and $O(|C||V|)$ is the time it takes to compute the parameters from the counts. Generally the time complexity is just $O(|D| L_{avg})$ as $|C||V| < |D| L_{avg}$.

In testing mode, the time complexity is $O(|C| L_t)$ where L_t is the average length of a test document. Thus, Naive Bayes is linear in the size of the training set and the test document. Hence it is optimal.

C. Advantages of Naive Bayes

- Classification results of naïve Bayes (the class with maximum posterior probability) are usually fairly accurate.
- Very fast as learning is done with one pass of counting over the data.
- Low storage requirements.
- Robust to irrelevant features.
- Very good in domains with many equally important features.
- A good dependable baseline for text classification.

IV. CONCLUSIONS

The basic functionalities of an Online Social Network have been created. Only registered users can have access to the network. There is a provision for new registration too. The basic functionalities include setting of profile picture, wall picture, uploading text and image onto friends' walls, creating events, adding and viewing friends list, managing account settings etc.

In the next step, Naïve Bayes classifier along with Weka is used for text classification thereby unwanted messages not get displayed on a person's wall. Images transferred are also filtered according to their titles given.

The future plan contemplates blacklisting of message creators who send unwanted messages continuously to a particular person.

REFERENCES

[1] Heinrichs J.H, Jeen-SU Lim, Kee Sook Lim (2011), Influence of social networking site and user access method on social media evaluation. *Journal of Consumer Behaviour*, Vol. 10 pp 347- 355.

[2] Social networking service From Wikipedia, the free encyclopaedia. Available: http://en.wikipedia.org/wiki/Social_networking_service

[3] Information filtering system From Wikipedia, the free encyclopaedia. Available: http://en.wikipedia.org/wiki/Information_filtering_system

[4] Marco Vanetti, Elisabetta Binaghi, Elena Ferrari, Barbara Carminati, and Moreno Carullo," A System to Filter Unwanted Messages from OSN User Walls" *IEEE Transactions On Knowledge And Data Engineering*, Vol. 25, No. 2, February 2013.

[5] Jiawei Han and Micheline Kamber, *Data Mining Concepts and Techniques*, 2nd ed.

[6] A. Adomavicius and G. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 6, pp. 734-749, June 2005.

[7] R.J. Mooney and L. Roy, "Content-Based Book Recommending Using Learning for Text Categorization," *Proc. Fifth ACM Conf. Digital Libraries*, pp. 195-204, 2000.

[8] P.W. Foltz and S.T. Dumais, "Personalized Information Delivery: An Analysis of Information Filtering Methods," *Comm. ACM*, vol. 35, no. 12, pp. 51-60, 1992.

[9] P.S. Jacobs and L.F. Rau, "Scisor: Extracting Information from On-Line News," *Comm. ACM*, vol. 33, no. 11, pp. 88-97, 1990.

[10] M.J. Pazzani and D. Billsus, "Learning and Revising User Profiles: The Identification of Interesting Web Sites," *Machine Learning*, vol. 27, no. 3, pp. 313-331, 1997.

[11] Russell Stuart and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed., Prentice Hall, ISBN 978-0137903955, 2003 [1995]

