Impact Factor (SJIF): 5.301

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444
Volume 5, Issue 5, May-2018

Mobile Data Mining Using Smartphone

Gaurav Kulkarni
Assistant Professor
Department of Computer Engineering
ITM Universe Vadodara, Gujarat, India
kulkarnigaurav@yahoo.com

Abstract— **More and more data mining applications are running on mobile devices such as 'Tablet PCs', smart phones and Personal Digital Assistants (PDAs). The ability to make phone calls and send SMS messages nowadays seems to be merely an additional feature rather than the core functionality of a smart phone. Smartphone's offer a wide variety of sensors such as cameras and gyroscope as well as network technologies such as *Bluetooth*, and *Wi-Fi* with which a variety of different data can be generated, received and recorded. Furthermore smartphones are computationally able to perform data analysis tasks on these received, or sensed data such as data mining. Many data mining technologies for smartphones are tailored for data streams due to the fact that sensed data is usually received and generated in real-time, and due to the fact that limited storage capacity on mobile devices requires that the data is analyzed and mined on the fly while it is being generated or received. For example, the *Open Mobile Miner (OMM)* tool allows the implementation of data mining algorithms for data streams that can be run on smartphones. However, to the best of our knowledge, all existing data mining systems for mobile devices either facilitate data mining on a single node or follow a centralized approach where data mining results are communicated back to a server which makes decisions based on the submitted results.**

Keywords—smartphones; computations; OMM; datamining.

I. INTRODUCTION

Data mining on mobile devices can be broadly classified into two categories: (1) *mobile interface*; and (2) *on-board execution*. In the former one, the mobile device is used as an interface to a data mining process that runs on a high performance computational facility, like the *cloud*. This represented the early history in the area of mobile data mining, with the *MobiMine* system [4] to analyze stock market share prices being the first realization of this category. In the on-board execution category, the mobile device is used to not only set the parameters and visualize the results, but also to run the data mining process. This category has been the result of the continuous advances in mobile devices like smartphones and tablet computers. Many of these devices are comparable in terms of computational power to computer servers that run data-intensive tasks a decade ago. In the following, we give representative examples of each of the two categories. At the time of development, *MobiMine* was a pioneering piece of work. Despite the fact that this client/server architecture with a thin client representing the mobile device was motivated by the limited power of such devices at the time, it has opened the door for further development that exploited the continuous advances in hardware of the handheld devices. The system paid special attention to how efficiently a decision tree. can be communicated over a channel with limited bandwidth; a real problem at the time, that now seems not to be of an issue. The motivation behind the develop-ment of *MobiMine* is to help businessmen on the move to analyze the stock market shares and take a decision without having to attend to their personal computers. A breakdown of tasks between the Personal Digital Assistant (PDA) and the server was designed - such that all the computationally intensive processes are performed at the server side. To ensure that communication will not fail the system, a Fourier transformation of the signal was used. *MobiMine* will be given special treatment in this book as a potential application area for *PDM*.

II. PROBLEM

Let X be the space of attributes and its possible values and Y be the set of possible (discrete) class labels. Each ubiquitous device aims to learn the underlying concept from a stream DS of labeled records where the set of class labels Y is fixed. How-ever, the feature space X does not need to be static. Let $X_i = (\mathbf{x_i}, y_i)$ with $x_i \in X$ and $y_i \in Y$, be the i^{th} record in DS. We assume that the underlying concept is a

function that assigns each record x_i to the true class label y_i. This function f can be approx-imated using a data stream mining algorithm to train a model m on a device from

the *DS* labeled records. The model m returns the class label of an unlabeled record $\mathbf{x}$, such that $m(\mathbf{x}) = y \in Y$. The aim is to minimize the error of m (i.e., the number of predictions different from f). However, the underlying concept of interest f may change over time and the number of labeled records available for that concept can sometimes be limited. To address such situations, we propose to exploit similari-ties in models from other devices and use the available labeled records from *DS* to obtain the model *m*. We expect *m* to be more accurate than using the local labeled records alone when building the model. The incremental learning of *m* should adapt to changes in the underlying concept and easily integrate new models. We assume that the models from other devices are available and can be integrated at anytime.

III. Implementation of Mobile Data Mining

The *MDM* framework requires the following software components: (1) an operat-ing system for the mobile devices, (2) a mobile agent platform to host the *MDM* agents outlined in the data mining software/libraries used to add functionality to *MDM*.

A. The Mobile Operating System

Currently the smartphone industry is undergoing a major shift, where new operating systems (*OS*) have emerged, namely *iOS* and *Android* , which are currently pushing their rapid development. Some years ago the usage of a smartphone would not have seemed appropriate for an ordinary user, as it offered some complex functionality that would barely be useful and would have imposed a steep learning curve for the user. The introduction of these operating systems changed the smartphone's popularity. Thanks to an improved *Graphical User Interface (GUI)*, ease of usage, af-fordable prices, better hardware and the existence of useful applications, contributed to the success of smartphones for the everyday usage. The other type of phones,

commonly known as *feature phones*, have less computational power and offer less functionality with the reasoning that a mobile phone is to be used mainly for calls and messages and probably a few extras.

The main advantage of this type of phones is that they are cheaper than smartphones and offer better battery performance, pro-viding more usage time between charges. It is very common to find a customised *OS* on these mobile phones as it has to be tuned for the specific hardware for better performance. This in turn makes it difficult to write programs that work on different mobile phone platforms if they are not running the same *OS*. Thus these phones usually support *Java ME (Micro Edition)* in order to run third party programs. The biggest problem with *Java ME*, is that it was designed for computational constrained devices, so it attempts to support only the bare minimum functionality while avoiding more complex tasks.

The idea for *MDM* is to have a multi-platform implementation, capable of running seamlessly across different mobile devices. But nevertheless, a mobile *OS* has to be chosen as the first prototype. For testing purposes, low costs are preferred, plus at least an *OS* with a decent market share and enough power to support data mining.

In this case *Android* seems to be the best choice, as it is open source, offers a free software development kit and according to a *Nielsen* study shown in Figure 1.1, it has the biggest acquisition share as of September 2011 in the United Kingdom. Nonetheless, other *OS*s should not be neglected. *Android* has simply been chosen as the first implementation target. Potential applications for *MDM* are manifold and comprise amongst others the areas of financial investments, the health sector, public safety and defense. It introduces possible application scenarios in these areas, however, many more are possible, especially in science, such as human behavior detection, user behavior detection in order to detect the use of the phone by a potential thief, or detection of potential for mass panics on large public events.

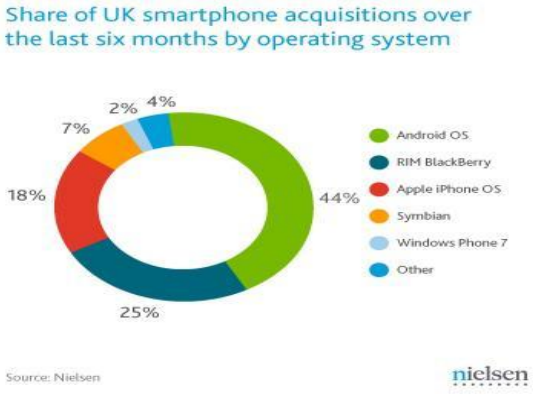

Figure 1.1

B. The Mobile-Agent Platform

Mobile agent technology has been around for over a decade, which has allowed it to evolve and to offer different approaches. During this time some frameworks have been developed to ease the use of this technology and to ensure the interoperability.

C. MDM Addressing MobiMine Shortcomings

MDM could build a local watch list using the local share subscription and subscrip-tions of different brokers. The local watch list is created by a local *Agent Miner (AM)* that could implement the algorithms of *MobiMine* or different more customized ones global influence could be incorporated by sending a *Mobile Agent Decision Maker (MADM)* that consults local *AM*s in order to find global relationships to the subscribed data. Also users may have subscribed to overlapping shares. Lets say broker *X* is interested in shares A, B, C and user *Y* is interested in shares B,D,E. Now *X* sends an *MADM* to all participating mobile devices among which is *Y* 's mobile device. The *MADM* discovers that *Y* 's *AM* has put B and E into the watch list, thus the *MADM* may advise *X* to put B into its watch list as *X* is interested in share B.

D. MDM as a Decision Support System for Investments

If a broker is interested in a new share, but has no experience in investing in this particular share, or the market in which the share is traded, then s/he may use *PDM* in order to support its investment decision. For example, in a *PDM* network of bro-kers, a broker could run a classification system wrapped in an *AM*, if the broker is certain about his share.

IV. Experimental validation in Mobile Data Mining

Having discussed in sufficient details our *Coll-Stream* technique extending *MDM* to deal with the concept drift problem, we conducted experiments to test the proposed approach's accuracy in different situations, using a variety of synthetic and real datasets. The implementation of the learning algorithm was developed in *Java*, using the *MOA* [8] environment as a test-bed. The *MOA* evaluation features and some of its algorithms were used, both as base classifiers to be integrated in the ensemble of classifiers and in the experiments for accuracy comparison.

A. Datasets

A description of the datasets used in our experimental studies is given in the following.

B. Stagger

This dataset was introduced by Schlimmer and Granger [5] to test the *STAGGER* concept drift tracking algorithm. The *STAGGER* concepts are available as a data stream generator in *MOA* and have been used as a *benchmark* dataset to test concept drift [5]. The dataset represents a simple block world defined by three nominal attributes *size*, *color* and *shape*, each with 3 different values. The target concepts are:

- $size \equiv small \wedge color \equiv red$
- $color \equiv green \vee shape \equiv circular$
- $size \equiv (medium \vee large)$

C. SEA

The *SEA* concepts dataset was introduced by Street and Kim [13] to test their *Stream Ensemble Algorithm*. It is another *benchmark* dataset as it uses different concepts to simulate concept drift, allowing control over the target concepts in our experiments. The dataset has two classes {class 0, class 1} and three features with values between 0 and 10 but only the first two features are relevant. The target concept function classifies a record as class 1 if $f_1 + f_2 \leq \theta$ and otherwise as class 0. The features f_1 and f_2 are the two relevant ones and θ is the threshold value between the two classes. Four target concept functions were proposed in [13], using threshold values 8, 9, 7 and 9.5. This dataset is also available in *MOA* as a data stream generator, and it allows control over the noise in the data stream. The noise is introduced as the *p*% of records where the class label is changed.

C. Web

The *webKD* dataset contains web pages of computer science departments of vari-ous universities. The corpus contains 4,199 pages (2,803 training pages and 1,396 testing pages), which are categorized into: *project*; *course*; *faculty*; *student*. For our experiments, we created a data stream generator with this dataset and defined 4 concepts, that represent user interest in certain pages. These are:

- $course \vee project$
- $faculty \vee project$
- $course \vee student$
- $faculty \vee student$

D. Reuters

The *Reuters* dataset is usually used to test text categorization approaches. It con-tains 21,578 news documents from the *Reuters* news agency collected from its newswire in 1987. From the original dataset, two different datasets are usually used, R52 and R8. R52 is the dataset with the 52 most frequent categories, whereas R8 only uses the 8 most frequent categories. The R8 dataset has 5,485 training docu-ments and 2,189 testing documents. In our experiments from R8, we use the most frequent categories: *earn* (2,229 documents), *acq* (3,923 documents) and *others* (a group with the 6 remaining categories, with 1,459 documents). Similar to the *Web* dataset, in our experiments, we define 4 concepts (i.e., user interest) with these categories

V. CONCLUSION

In this monograph, we detailed the development of our *Mobile Data Mining (MDM)* framework. *MDM* is set to serve the next generation of applications in predictive analytics targeting users of smart handheld devices. In its current implementation, the *MADM* visits all available *AM*s, however, this may be impracticable if the number of *AM*s is very large. Currently, a mechanism is being developed for *MADM*s according to which the *MADM* can decide when to stop consulting further *AM*s. A possible stopping criteria could be that a certain time has elapsed or the classification result is reliable enough. Also the rating system outlined above can be used to determine an order in which *AM*s are visited. If there are time constraints the *MADM* may prioritise more reliable *AM*s.

VI. FUTURE WORK

The current implementation of the *MADM* agent assumes that the local *AM*s are of good quality, and thus in the case of classification of unlabeled data instances, it is assumed that the weights are calculated correctly and truly reflect the *AM*s classification accuracy. This assumption may be true for the *AM*s we developed in-house, which we used for the evaluation, but third party implementations may not be trusted. For this reason, a rating system about *AM*s is currently being developed based on historical consultations of *AM*s by the *MADM*. For example, if the *MADM* remembers the classifications and weights obtained from *AM*s visited and the true classification of the previously unknown instances is revealed, then the *MADM* could implement its own rating system and rate how reliable an *AM's* weight was in the past. If an *AM* is rated as unreliable, then the *MADM* may even further lower its weight. However, it is essential that this rating system is also able to loosen given ratings, as the *AM's* performance might well change if there is a concept drift in the data stream. In order to detect such concept drifts, it is necessary that *AMs* that have a bad rating are still taken into consideration, even if it is with a low impact due to bad ratings.

We have just outlined a possible rating system for classification *AMs*. How-ever, rating systems for other less generic *MDM* agents such as the *GDF* and *LFA* agents outlined in the applications of *MDM* remains an open area to be explored.*MDM* is a new niche of distributed data mining. The current implementation of *MDM* focuses on classification techniques, however, there exist many more data mining technologies tailored for data streams and mobile devices. For example, there are stream mining techniques that classify unlabelled data streams [40, 75] which could be introduced into *MDM*.

REFERENCES

[1] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)
[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.
[3] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[4] Kargupta, H., Hoon Park, B., Pittie, S., Liu, L., Kushraj, D.: Mobimine: Monitoring the stock market from a pda. ACM SIGKDD Explorations 3, 37–46 (2002)

[5] Schlimmer, J., Granger, R.: Beyond incremental processing: Tracking concept drift.In: Proceedings of the Fifth National Conference on Artificial Intelligence, vol. 1, pp. 502–507 (1986)

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[8] Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B.: Moa: Massive online analysis. The Journal of Machine Learning Research 99, 1601–1604 (2010)

[9] Bifet, A., Kirkby, R.: Data stream mining: a practical approach. Tech. rep., Center for Open Source Innovation (2009)

[10] Blake, C.L., Merz, C.J.: UCI repository of machine learning databases. Tech. rep., Uni-versity of California, Irvine, Department of Information and Computer Sciences (1998)

[11] Chan, P., Stolfo, S.J.: Meta-Learning for multi strategy and parallel learning. Pro-ceedings of the Second International Workshop on Multistrategy Learning, pp. 150–165 (1993).

[12] Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. Com-mun. ACM 51, 107–113 (2008)

[13] Street, W., Kim, Y.: A streaming ensemble algorithm (SEA) for large-scale classifi-cation. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 377–382. ACM, New York (2001)

[14] Hotho, A., Pedersen, R.U., Wurst, M.: Ubiquitous Data. In: May, M., Saitta, L. (eds.) Ubiquitous Knowledge Discovery. LNCS, vol. 6202, pp. 61–74. Springer, Heidelberg (2010)

[15] Stahl, F., Gaber, M., Bramer, M., Yu, P.: Pocket data mining: Towards collaborative data mining in mobile computing vironments. In: 22nd IEEE International Con-ference on Tools with Artificial Intelligence (ICTAI), vol. 2, pp. 323–330 (2010), doi:10.1109/ICTAI.2010.118

[16] Wang, H., Fan, W., Yu, P., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 226–235. ACM, New York (2003)

[17] Way, J., Smith, E.A.: The evolution of synthetic aperture radar systems and their pro-gression to the eos sar. IEEE Transactions on Geoscience and Remote Sensing 29(6), 962–985 (1991)

[18] Pan, S., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 1345–1359 (2010)

[19] Park, B., Kargupta, H.: Distributed data mining: Algorithms, systems and applications.In: Data Mining Handbook, pp. 341–358. IEA (2002)

[20] Gomes, J., Krishnaswamy, S., Gaber, M.M., Sousa, P.A., Menasalvas, E.: Mars: a personalized mobile activity recognition system. In: 2012 IEEE 13th International Confer-ence on Mobile Data Management, MDM, pp. 316–319. IEEE (2012)