MOBILE MIND: A FULLY MOBILE PLATFORM BASED MACHINE LEARNING APPLICATION

Ahmet Selman BOZKIR
Hacettepe University Computer Engineering Department, Ankara, Turkey
selman@cs.hacettepe.edu.tr

Ebru AKCAPINAR SEZER
Hacettepe University Computer Engineering Department, Ankara, Turkey
ebru@hacettepe.edu.tr

ABSTRACT

In recent years, mobile devices have developed significantly in terms of technical capabilities, computing power, storage capacity and ability of sensing different activities via intelligent built-in sensors. In this perspective, capabilities of ultimate mobile phone technology has begun to be a candidate novel platform for machine learning and data mining activities by the help of its computing power. In this study, a fully mobile platform based machine learning application named Mobile Mind is designed and implemented. While, all other current mobile platform based machine learning and data mining applications are using central data mining servers to perform analysis, Mobile Mind does all tasks on cell phone's processor and memory. On the other hand, Mobile Mind currently supports support vector regression and kernel recursive least squares regression algorithms with polynomial and radial basis kernels to allow users performing predictive data mining operations on flat CSV (comma separated values) files. By this study, it is shown that mobile platforms are becoming native and ubiquitous platforms for machine learning purposes from now on. Therefore, the need of central data mining servers and web service usage for data transferring will started to be less and less in the future. Furthermore, a native fully mobile machine learning tool presents unlimited opportunities to the mobile application programmers especially dealing with sensor data driven applications has much potential in this point of view.

KEYWORDS

Mobile data mining, Machine learning, Support vector regression, Kernel recursive least squares

1. INTRODUCTION

Mobile data mining has started to be one the important fields of pervasive and ubiquitous mobile computing. In their study, Talia and Trunfio described the three possible usage scenarios of mobile devices in data mining oriented applications such as (1) terminal mode which users access to a central data mining server, select data source and invoke operations on data along with a returning result; (2) data generator mode that users or mobile devices generate data and send it to a remote server for further processing and (3) miner mode that all data mining related operations are done on device itself (Talia & Trunfio, 2010). However, it's reported that third mode is unrealistic because of limited capabilities of mobile devices such as computing power and storage area (Talia & Trunfio, 2010). In essence, if literature is reviewed, it can be seen that, many of the mobile data mining studies rely on utilizing first two modes. In one study, authors designed and implemented a distributed mobile data mining environment conforming to first usage scenario depicted above to smartly monitor and visualize stock market data on PDAs by utilizing Fourier transform of decision trees and employing Java technologies (Kargupta et al., 2002). In another study, a mobile data mining application based on communication with a remote data mining server via web service techniques is introduced (Talia & Trunfio, 2010). In that study, users allowed to select a data table on server and perform data mining methods by invoking remote built-in functions of Weka Toolkit (Weka, 2010). On the other hand, due to shortcomings of first and second modes in mobile data mining scenarios, an efficient mobile data mining model is developed which helps to preprocess of local mobile data and gain descriptive statistics before sending it to remote mining servers (Goh & Taniar, 2005). In another study, an intelligent system named VEDAS which tracks and monitors of vehicles by using on-board PDAs connected through wireless networks and performs data mining analysis over recorded paths is introduced (Kargupta et al., 2003).

As can be seen, mobile data mining centric applications have generally avoided using the mobile device itself for performing whole data mining task. Instead, they have used central data mining servers for analyzes and utilize web service or similar technologies for data transferring and gathering responses. This approach has the advantage of high speed and scalable data analysis but suffers from well-known low bandwidth problem of communication lines. However, as a result of innovations in mobile device industry, current high-end mobile devices have been equipped with adequate size of multi-touch screens, built-in magnetic field, proximity and accelerometer sensors, high resolution cameras, powerful processors up to 1 GHz, large volumes of storage capacities up to 32 GB which eliminate the limitations over scenario three and allows mobile devices to be a powerful candidate platform for data mining & machine learning activities. Furthermore, due to rapid development of mobile devices and built-in sensors, the way of mobile application development has evolved and mobile devices have became self data generators in this case. Therefore, local analysis of sensor generated data has much potential and valuable. One important instance of this case is motion based recognition. Liu et al., developed a single passes gesture recognition algorithm for three-axis accelerometers which allows capturing and revealing patterns of various gestures (Liu et al., 2009). As today's mobile devices have these kinds of accelerometers, it is possible to recognize motions of users and develop authentication mechanisms for mobile devices. However, to achieve this and similar goals, existence of mobile classifier or regressor is very essential.

In this paper, we introduce a completely mobile based machine learning tool named Mobile Mind. To allow users performing predictions in test data through employing pre-installed algorithms over train data constitutes the main purpose of the present study. Mobile Mind currently supports support vector regression and kernel recursive least squares (an online algorithm) algorithms with polynomial and radial basis kernel choices. The algorithms used in Mobile Mind are migrated from Dlib Machine Learning Toolkit (King, 2009).

2. MOBILE MIND WITH ALL ASPECTS

As described in previous section, many data mining and machine learning related applications on mobile platforms are designed and implemented in client-server architecture considering the limited capabilities of mobile devices, such as processor power, low memory and battery life. However, as the aim of this study is to run mining algorithms on device itself, current high-end cell phones and platforms attributes are investigated at startup. Google's Android (Android, 2011), Apple's IOS 4 (IOS 4, 2011) and Samsung's Bada (Bada, 2011) were the choices stand out with their pervasiveness, robustness and easy programmability features. However, as our core mining library Dlib Machine Learning Toolkit was written in standard C++ language and the platform which has C++ support was only the Bada platform, Samsung's Bada platform has been chosen. The reason behind the selection of Dlib library rather than others is being an open-source project, having many kernel based methods including support vector machine, relevance vector machine and kernel recursive least squares, being written in standard C++ language for easy migration between different platforms and regular updates for new features and bug fixing. On the other hand, Dlib library presents many other utility libraries as well as image processing, networking and optimization that can be used in this study in the future for further aims.

Support vector regression (SVR) and kernel recursive least squares (KRLS) algorithms are selected and adopted in Mobile Mind due to some reasons. First of all, SVR is a state-of-algorithm which proved its reliability and pervasiveness by being used in many study fields such as medicine and finance (Farquad et al., 2010) On the other hand, KRLS is an on-line learning algorithm developed by Engel, Mannor and Meir at 2003 which has advantage of avoiding re-training of whole training cases from startup for new upcoming cases. On the other hand, KRLS is highly suitable algorithm for mining of streaming data and time series analysis (Engel, Mannor & Meir, 2003). Therefore it is selected to be a part of Mobile Mind for streaming data analysis and sensor activity mining purposes that are high probable in cell phone and sensor based future applications. Readers can found deeper information about Bada platform, algorithms used in this study and details of Mobile Mind in the sections below.

2.1 Bada smartphone platform

Bada platform is an operating system named BadaOS designed and implemented for cell phones developed by Samsung. BadaOS is currently used various cell phones of Samsung including Wave 8500, Wave II 8530 and Samsung S7233. Samsung provides software development kits (SDK) regularly and version of ultimate SDK is 1.2.1 which was released as of December 17 of 2010. SDK presents GUI, I/O, security, networking, media, location based services and social networking libraries & namespaces for developers together with easy-to-use integrated development environment (IDE) which is shipped with SDK. Although, Bada is a very new platform, due to its robust structure and C++ basis, it has found worldwide usage throughout the mobile application programmers. However, it has some limitations due to its immaturity such as lack of full support for standard C++ libraries and multitasking issues that avoids an application to start another application without blocking running status of its own. To overcome these problems Samsung is currently developing BadaOS 2.0 platform which will have support for complete standard C++ and fully multitasking along with new unique improvements such as built-in text to speech (TTS) and speech recognition.

2.2 Algorithms used in Mobile Mind

Mobile Mind is a developing tool and currently supports only SVR and KRLS algorithms which are chosen for different purposes. In next versions new other algorithms which have already been implemented in Dlib library will be adopted in Mobile Mind.

SVR which was developed by Vapnik at 1995 is based on statistical learning theory and aims to create a hyperplane which lies near to or close to as many instances as possible (Farquad et al., 2010; Ancona, 1999). On the other hand, as SVR is a kernel-based method, it does not have some shortcomings which some other methods have such as decision trees in terms of over-fitting and competition of input variables on node splitting. However, it needs more time than the other non-kernel based algorithms for computing support vectors in case of increment in number of training examples. By considering the pros and cons of SVR, it is decided to be included in Mobile Mind. SVR implementation in Dlib ML toolkit is epsilon insensitive version of SVR that requires three parameters such as kernel type, C (regularization parameter) and epsilon-insensitive parameter that sets the accuracy level of regression.

KRLS on the other hand, is another kernel based method which is an improved version of recursive least squares algorithm, that is popular and practical in many fields covering signal processing, communications and control kernel (Engel, Mannor & Meir, 2003). KRLS performs linear regression in the feature space induced by Mercer kernel. For this reason, KRLS algorithm can be used to build a minimum mean squared error regressor recursively. On the other hand, KRLS is an online algorithm which means that algorithm does not need to be re-trained from scratch when new train cases are available. According Engel et al., "on-line algorithms are useful in learning scenarios where input samples are observed sequentially, on at a time (e.g. data mining, time series prediction, reinforcement learning)" (Engel, Mannor & Meir, 2003). Moreover, on-line algorithms are useful and practical in real-time decision giving operations. For these reasons, KRLS is included in Mobile Mind algorithms library. KRLS implementation in Dlib ML toolkit requires three parameters such as kernel type, accuracy level and dictionary size determining how many of the dictionary vectors will be memorized during training phase.

As stated before, Mobile Mind uses kernel based methods. Therefore currently polynomial and radial basis (Gaussian) kernels are presented to users. As a result of this, different numbers of kernel parameters for each of these kernel types are presented in GUI such as gamma value (for both of them), coefficient and degree values (for only polynomial kernel)

2.3 Design, implementation and usage of Mobile Mind

Mobile Mind is written in C++ on Bada SDK 1.2.1 and includes Dlib machine learning toolkit library as a background algorithm engine. Due to be compact and shrink the size of program, only machine learning and essential components of Dlib are integrated at compiling stage. As a visual design requirement, mostly used functions are located on main form as button elements instead of menu items. Thus, a perspicuous and easy to use environment is tried to be obtained. As depicted in Fig. 1, usage of Mobile Miner consists of four essential and one optional stage. First of all, users must pick training and testing data files which must be conformed to CSV file standards. Predictions can be performed with existence of these two files. On the other hand, to measure the generalization capability of prediction, R^2 (determination of coefficient) measure is selected. To obtain R^2 value, user must provide also one another file named as “observed values” file which contains only the observed values as a column matrix. After picking data files, users either perform training by pushing on selected algorithm button or can continue to configure operation’s kernel type or parameters as depicted in Fig. 2.b. After training stage, algorithms predict the cases given in test files and outputs R^2 result with output file (if filename is defined) as depicted in Fig. 2.d.

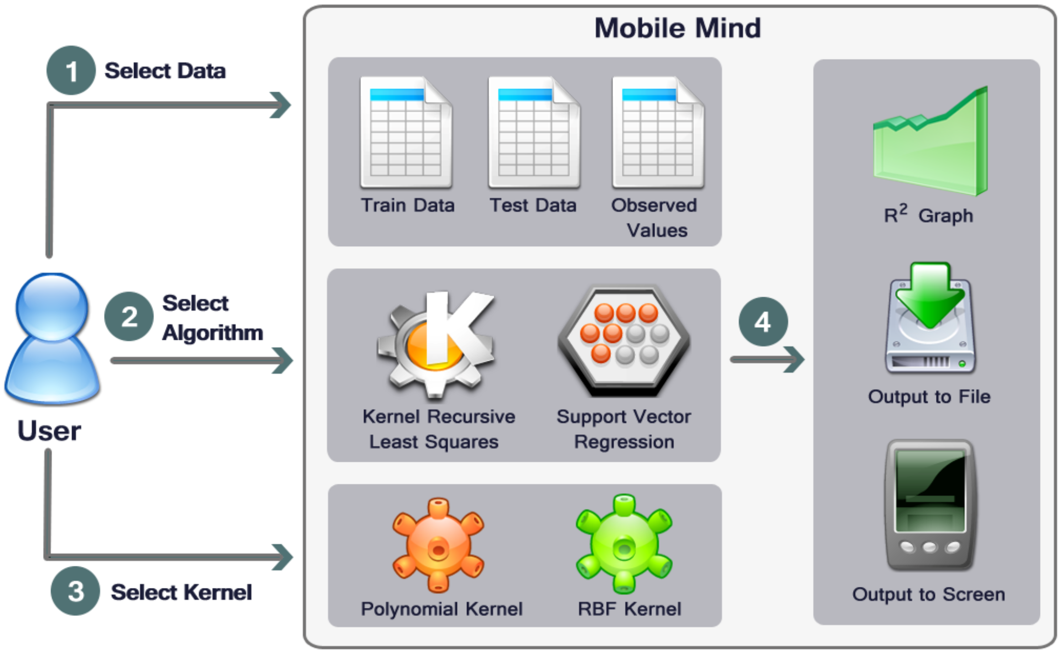

Figure 1. System chart of Mobile Mind

As an obligation, the output variable must be the in last order in training file and the test file should not contain observed values (output variable) as reside in training file. Instead, observed values must be stored in another file named “observed values file”. During the development stage, model saving and re-loading capabilities are tried to be implemented with the co-operation of Dlib. However, as current Bada platform does not have full support of standard C++ and <iostream> classes, model saving and re-loading methods could not be implemented. Likewise, these features require of serialization and storing of created models and need the <iostream> and some other standard C++ functions.

To measure the performance of the methods and check whether or not methods are suitable for mobile machine learning purposes, Mobile Mind is benchmarked with three datasets by employing each algorithm with two types of kernels. All these datasets have only continuous variables as well as output variable and datasets are divided into train and test groups with 80% - 20% partitions. First dataset includes data gathered via a tunnel boring machine and the aim of the data is to predict “field rate penetration” variable. This data is also subjected in a study (Akcapinar Sezer et al., 2010). The remaining datasets are downloaded from UCI Machine Learning Repository located at http://archive.ics.uci.edu/ml/datasets/. Subject of the second dataset is predicting concrete compressive strength by using age and other ingredient variables. As a last and larger one, third dataset contains “parkinsons telemonitoring” data in terms of voice recording measurements of 42 patients, aiming to predict “total UPDRS” variable by using other input variables. The properties of datasets are given in Table 1 below.

Figure 2. Screenshot of Mobile Mind: [a] Main form, [b] Parameter settings form, [c] Main menu and [d] Result form

In benchmark stage, gamma, accuracy, dictionary size (specific to KRLS method), C (regularization parameter specific to support vector machine methodology), coefficient and degree parameters (used by only polynomial kernel) are kept as 0.0005, 0.0001, 10000, 10, 2 and 2 respectively. All tests are executed at simulator. As the goal of the benchmark is to measure the capabilities of algorithms on mobile device and determine the concordance of algorithms for mobile machine learning purposes, count of seconds elapsed during operation and maximum memory usage records are determined. In time measurements, only training and testing phases are considered while R^2 calculation and file outputting are excluded .On the other hand, 600 seconds (10 minutes) is accepted as a maximum threshold value of operation. Thus, operations exceeding 600 seconds are cut.

Table 1. Different datasets with features and corresponding computational cost in time

Algorithm - Kernel	Dataset	# of Input Variables	# of Training Cases	# of Test Cases	# of Seconds To Finish Job	Maximum Memory Usage (MB)
SVR – Polynomial	TBM	4	119	32	8	79
SVR – Radial Basis	TBM	4	119	32	1,5	81
KRLS – Polynomial	TBM	4	119	32	3	82
KRLS – Radial Basis	TBM	4	119	32	3.5	82
SVR – Polynomial	Concrete	8	822	208	>600	87
SVR – Radial Basis	Concrete	8	822	208	9	86
KRLS – Polynomial	Concrete	8	822	208	19	79
KRLS – Radial Basis	Concrete	8	822	208	103	98
SVR – Polynomial	Parkinsons	18	4703	1172	>600	275
SVR – Radial Basis	Parkinsons	18	4703	1172	235	260
KRLS – Polynomial	Parkinsons	18	4703	1172	98	81
KRLS – Radial Basis	Parkinsons	18	4703	1172	131	79

3. DISCUSSION

If the results given on Table 1 are investigated, it can be seen that KRLS algorithm performs fast and scalable analyses rather than SVR. As it uses a dictionary vector and it only remembers last important vectors (the memorization level can be set by dictionary size) required memory size remains constant and therefore it is not affected by the numbers of cases in training data. On the other hand, considering time needs, KRLS performs faster mining with polynomial kernel than radial basis kernel. From this point of view, analysis with polynomial kernel stands more plausible in KRLS usage. However, as KRLS uses a dictionary size, it increments required memory space linearly; KRLS is suitable for both kernel types. Likewise it is indispensible to use radial basis kernel for non-linear analysis.

In SVR, memory usage increments exponentially when number of cases and especially the number of variables increases. Performing SVR with radial basis kernel requires acceptable amount time to perform. However, when polynomial kernel is to be discussed, due to the nature and structure of algorithm, SVR requires excessive amounts of time which is unacceptable. Therefore, it can be stated that SVR with polynomial kernel is useless for current mobile cell-phone features. The main reason of this finding is SVR generates exponential growing number of support vectors by the increment of training data size. Furthermore, working with polynomial kernel on SVR requires degree value to bet as an integer and gamma value is small enough so that the output of the kernel is not a huge number. However, SVR & polynomial kernel couple runs slower than KRLS.

In the tests, the biggest training file contains 4703 cases. If the fact that ultimate mobile devices which BadaOS runs on, have at most 256 MB usable memory, it can be concluded that KRLS is a very suitable algorithm for mobile machine learning as it is not affected from the number of training cases. In contrast, SVR is a reasonable algorithm unless polynomial kernel is used. However, as it is not designed as an online learner, it can be clearly stated that SVR has limitations in mobile device platforms due to memory issues.

4. CONCLUSION AND FUTURE WORK

In this study, a fully mobile platform based machine learning tool named Mobile Mind was developed. It is designed and implemented to let the users to perform whole machine learning processes on mobile device instead of using remote servers. As can be seen in the study, ultimate mobile device technology has become capable of doing such kind of tasks. Likewise, currently used processors in mobile devices can reach up 1GHz. If benchmark tests are considered together with this improvement, processor capability is not barrier for many cases from now on. However, as the machine learning algorithms require memory, available memory area could still be a bottleneck in mobile machine learning. Due to fact that, mobile devices do not have virtual memory features as existed in PCs, algorithms are being limited with the amount of memory at device. On the other hand, with this study, it is shown that mobile devices have enough power to be a mobile classifier or regressor for artificial intelligence, machine learning and data mining centric applications.

As a further research, adapting new classifier algorithms which are independent from the count of training cases is planned. Additionally, model load and save utilities are planning to be implemented in Mobile Mind whenever standard C++ support is available in Bada.

REFERENCES

Akcapinar Sezer, E., Bozkir, A.S., Yagiz, S, Gokceoglu, S.,2010, Karar Ağacı Derinliğinin CART Algoritmasında Kestirim Kapasitesine Etkisi: Bir Tünel Açma Makinesinin İlerleme Hızı Üzerinde Uygulama, Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu, Kayseri, Turkey.

Ancona, N., 1999, Classification properties of support vector machines for regression. Technical report, R.I.-IESCI/CNR-Nr., 02/99

Android, 2011, Available: http://www.android.com/

Bada, 2011, Available: http://www.bada.com/

Goh, J. and Taniar, D., 2005, An Efficient Mobile Data Mining Model, Springer-Verlag LNCS, pp 54-58.

IOS 4, 2011, Available : http://www.apple.com/iphone/ios4/

Engel, Y., Manor, S., Meir, R., 2003. The Kernel Recursive Least Squares Algorithm, *IEEE Transactions on Signal Processing*, Vol. 52, pp 2275-2285.

Farkuad, M.A.H., Ravi, V., Bapi Raju, S., 2010. Support vector regression based hybrid rule extraction methods for forecasting, *Expert Systems with Applications*, Vol. 37, No. 8, pp 5577-5589.

Kargupta, H. et al, 2002. Mobimine: monitoring the stock market from a PDA, *ACM SIGKDD Explorations*. Vol. 3, No. 2, pp 37-46.

Kargupta, H. et al, 2003. VEDAS: A Mobile and Distributed Data Stream Mining System for Real-Time Vehicle Monitoring. Proc. SIAM Data Mining Conference.

King, D., 2009, Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research,* Vol. 10, pp 1755-1758.

Liu, J. Et al, 2009, uWave: Accelerometer-based personalized gesture recognition and its applications , *Pervasive and Mobile Computing*, Vol. 5, No. 6, pp 657-675.

Talia, D. and Trunfio, P., 2010. *Mobile Data Mining on Small Devices Through Web Services*, Wiley, USA.

Weka, 2010, Available: http://www.cs.waikato.ac.nz/ml/weka/