Grid Learning Classifiers - A Web Based Interface

Manuel Filipe Santos, Wesley Mathew, Henrique Santos

Abstract—The toolkit for learning classifier system for grid data mining is a communication channel between remote users and gridclass system. Gridclass system is the system for grid data mining, grid computing approach in the distributed data mining. This toolkit is a web based system therefore end users can set the configuration of each node in the grid environment and execute the grid class system from the remote location. Mainly, configuration module of the toolkit is designed for the sUpervised Classifier System (UCS) as a data mining algorithm. Toolkit has three fundamental functions such as creating new project, updating the project, and executing the project. Initially, user has to define the project based on the complexity of the problem to the system. While creating a new project all the data and configuration information about all nods are stored in the file under a user defined project name. The updating phase user can makes changes in the configuration file or replace the training data for new experiments. There are two sub functions in the phase of execution: do the execution of gridclass system and do the comparison and evaluation of the performance of the different executions. Toolkit can store the global model and related local models and it testing accuracies in the server system. The main focus of this work is to improve the performance of learning classifier system: therefore an attempt is made to compare the performance of learning classifier system with different configurations, which has a significant role. The ROC graph is the best option to represent the performance of classifier system. Accuracy under the curve (ACU) is a numerical value to represent the ROC curve. Therefore, users can easy to measure the performance of global model with the help of AUC. Other objective of this work is to provide friendly environment to the end users and gives better facilities to evaluate the performance of the global model.

Keywords—Gridclass system, Grid data mining, Supervised classifier system, Toolkit

I. Introduction

THE software toolkit is the set of controls that manage the performance of a system. The toolkit for learning classifier system for grid data mining is a communication channel between remote users and gridclass system. Main propose of this toolkit is to provide a friendly environment for end users and grant the option to analysis the performance of gridclass system. Distributed Data Mining (DDM) approach has been adopted in this gridclass system therefore data mining algorithm has to be executed in each and every nodes in the grid environment, in parallel and distributed fashion. Each node in the grid requires to be set its training data and all configuration information separately. This toolkit is efficient to manage all the communication between user and different nodes in the grid environment, therefore the toolkit is an important component for the execution of gridclass system.

The key components of the toolkit are; (1) creating a new project; (2) update the project; and (3) execute the grid class system. Creating new project implies storing the data and the configuring of each node of the grid environment. Update the project means make some modification in the configuration information or replace the training data that are already saved in a project name. For the research work user need to test same data with different configuration or different data with same configuration. This module is efficient make experiments and helps to save the configuring time. The execution face, user can trigger the gridclass system and after the construction of global model, it can receive all results from the gridclass system. Those results will be stored in the server system for future reference. The execution phase is also providing the services for comparing and analysis the performance of different executions. The execution phase, system can generate the ROC cure of the global model that will assist users to understand the performance of global model.

Particularly, the toolkit is made for the supervised learning classifier system as the learning classifier system in the grid environment. Supervised learning classifier system (UCS) is the one of the successful implementation of learning classifier system [1, 2, 3]. Each parameters of the learning system has an important role for building the global model. Therefore proper configuration of each classifier system is necessary.

The remaining sections of this paper are organized as follows. Section II describes about gridclass system and its structure. This section gives details of the components and highlights its significances. Section III describes the distributed data mining and two different methods of distributed data mining: Centralized Data Mining (CDM) and Distributed Data Mining (DDM). Section IV describes the potential of gird computing. Section V explains about the

Manuscript received October 9, 2001. (Write the date on which you submitted your paper for review.) This work was supported by Foundation of Science and Technology (FCT), Portugal. The authors would like to express their gratitude's to FCT for the financial support through the contract GRID/GRI/81736/2006.

M. F. Santos is an auxiliary professor of Information system department, school of engineering, university of Minho, Portugal. (phone: «351253510306; fax: +351253510300; e-mail: mfs@ dsi.uminho.pt).

W. Mathew is the research fellow in department of information system, school of engineering University of Minho, Portugal (e-mail: wesley@dsi.uminho.pt).

H. Santos Author is professor of Information system department, school of engineering, university of Minho, Portugal. (phone: +351253510319; fax: +351253510300; e-mail: hsantos@ dsi.uminho.pt)

characteristics and construction methodologies of ROC graph. Section VI illustrates the components and modules of the toolkit. Section VII includes basic level of discussion and section VIII summarizes the main conclusions of this work.

II. GRIDCLASS SYSTEM

Gridclass system is a grid based learning classifier system [1, 2, 4]. Grid based leaning classifier system is the distributed learning classifier system for grid data mining. Supervised learning classifier system (UCS) is used as data mining algorithm in gridclass system. In Distributed data mining, UCS needs to be executed in every nodes of the grid environment. The UCS was modified as a function of grid data mining and therefore it is called UCSgrid [4].

The given system has three basic components namely, toolkit, central node, and grid environment. The Figure 1 displays the basic components of gridclass system.

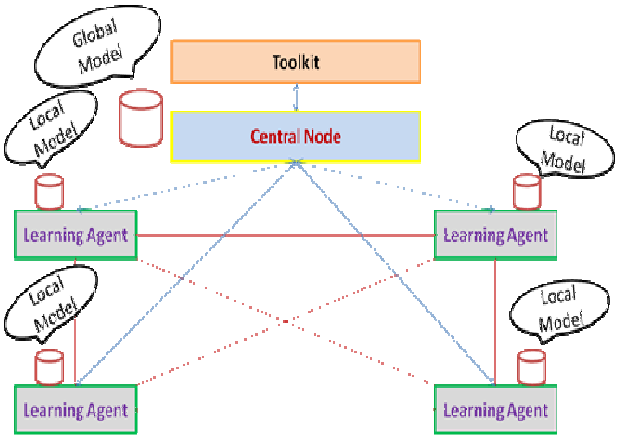

Figure 1. Basic components of GridClass system .

As mentioned, the toolkit is an interface between user and gridclass system. It is a web based system; hence the user can access the gridclass system from a remote location. The central node acts as a central managing node of gridclass system. The central node has direct access with each node in the grid by which it can send and receive data from each node. Initially, the central node receives all data and file configurations from the toolkit and then it distributes training and testing data sets to each node and triggers the data mining algorithm at each location. The data mining algorithm will process this training data and generate local model at each sites. After creates those local models, each node will return these local models to the central site. Then central site will generate global model from all local models using defined strategies. Grid environment is designed as parallel and distributed mode. Therefore each node in the grid structure will generate local models (learning model) at the same time. Each learning model will test separately at each node after the completion of the learning process.

The construction of global model is the main purpose of gridclass system. Local models only represent small portion of the whole problem, but the global model can represent all the problems together. So, global model is useful to analyze the all model together, instead of analyzing the each local data individually. The main novelity of the current work was to construct compact and effective global model from all the local models.

There are two levels of learning models in the Grid class system. The first level of learning model is developed from the data but the second level of model is developed from first level of learning models.

A. Level 1 learning model

Genetic algorithm based training algorithm is implemented in the gridclass system. During training process, learning system is developing a learning model based on the training data, and after the training process, accuracy of the learning model is tested with different sets of testing data. Gridclass system has an opportunity to introduce one learning model to training system. If user is introducing the learning model to the training system then it will act as increment learning. The increment learning can help to produce more generalized learning model at each execution. If user is not introducing the learning model then learning process will start with empty population. Genetic algorithm and covering are two vital operations for developing the learning model [1, 2, 4, 7]. Each training instance, one example is selecting from the environment (training data) to training process. The selected example is matched with all classifies in the population and makes the match set. If there are no classifiers that are matched with training data then covering process will execute. During covering operation, unmatched training data will be introduced to the population as a new classifier. Otherwise, if the average experience of the classifiers that are in match with the set is eligible to come to the genetic algorithm then genetic algorithm will be executed [1, 2, 7]. Two classifiers are selected as parent classifiers from the match set and processed to form two new classifiers. Crossover and mutation are the two processes for generating new classifier. The parent classifiers are divided into two sections based on the crossover probability. Through crossover and mutation operation parent classifiers are transformed to form new classifies. New classifiers will inherit the properties of the parent classifiers. If the child classifiers are more general than the parent classifier then child classifier will introduce into the population otherwise parent classifier will again add to the population [2, 4, 7]. Supervised learning classifier system has user defined population size. When the number of classifiers in the population is exceeding the maximum population size then the deletion operation will execute [1, 7].

There are two types of testing methods: one is online testing and second one is offline testing. Online testing implies testing process will execute after a particular period of training iteration. Offline learning implies testing process will execute only after the completion of training process. Level one learning process is executing at each and every node of the distributed environment.

B. Level 2 learning model

The participations of the second level of training process are the first level of learning models. Construction of second level of learning model (Global model) is the combined first level of learning models. Concatenated learning model should be the representation of all the learning models in the distributed site therefore constructing the global model is very decisive task. There are many things need to be

considered during the construction of optimized global model. Instead of conditions and actions, the parameters of each classifier have significant impact in monitoring its behavior. Individual local models have some influences from their training data size. Hence, before constructing the global model, each local model should be normalized. Optimized global model don't need to add all the classifier from all the local models instead global model must keep all generalized and more weighted classifiers. There are six different strategies that are used for constructing the global model: specific classifier method, generalized classifier method, majority voting method, weighted classifier method, model sampling and global training approach.

In specific classifier method only isolated classifiers are kept in the global model [4]. Isolated classifier implies there are no two similar classifiers in the global model. Local models contain several classifiers with similar condition and action part with different parameter values. However the presence of one repeated classifier in a global model, doesn't allow repeated classifier enter into the global model instead, updating the parameters of the classifier, which is already in the global model with parameters of repeated classifier. The global model size of specific classifier is not defined by user [4].

Weighted classifier method is the second method for constructing the global model. Criteria to be considered in a generalized classifier method are focused on the generality of the classifier. The generalized classifier method only protects more general classifiers in the global model [4]. More general classifiers can be replaced with less general classifiers; moreover repeated classifiers can also be avoided in the generalized classifier method. System doesn't allow the entrance of less general classifier into the global model. Alternatively, it will update the parameters of the more general classifiers with parameters of less general classifiers. In addition, the generalized classifier method could form more optimized global model. The global model size of generalized classifier method is lively.

Majority voting is the third method for constructing the global model. In majority voting approach, one cutoff threshold value to bench mark the classifiers in the global model [4]. This cutoff threshold value is calculated from the accuracies of the classifiers in the global model. If the accuracy of the classifier is less than this cutoff threshold value, then that classifier is removed from the global model. This method helps to sustain more valuable classifier in the global model. The global model size of majority voting approach is also dynamic.

Weighted classifier method is the forth approach for constructing the global model. Weighted method is the simple and effective method for constructing the global model. Here, system first calculates the weights of each classifier. In the weighted classifier method, user can define the size of the global model; hence highest weighted classifiers are stored in the global model based on user defined population size. The weight of a classifier is calculated based on the accuracy of the classifier. Accuracy of each classifier is multiplied with ratio of the size of local training data and size of global training data.

Model sampling approach is the fifth method for constructing the global model. Global model of model sampling approach contain samples of each classifiers in the local models. Sampling of a classifier is based on the experience of the classifier (*number of match*). Experience of a classifier is calculated from how many times that classifier matched with different training instances. Suppose the experience of a classifier is five then five different sample classifiers are added to the global model. Each sampling operation, the *don't care* symbols that are replaces with numeric value, which is feasible for the required position. The action part and all other parameters of the classifier will not make any changes. After sampling, system will filter the classifiers based on the user defined cut points. Filtering operation assists in removing the futile classifiers from the global model. The global population size of model sampling approach is not defined by users.

Global training approach is the sixth approach for constructing the global model. Main concept of global training approach is to generate a training data from all local models. Initially all classifiers convert to training data by sampling process. Similar to model sampling approach, sampling the classifiers are based on the experience of individual classifiers. All the parameters of the classifiers are ignoring in the training data more over all the *don't care* symbols in the condition part of the training data is replaced with other possible numeric values. Then, data mining algorithm is applied to the training data in the central system to form the global model. Global model size of global training approach is equal to the population size of the central training algorithm.

III. Distributed Data Mining

Distributed data mining is the field of data mining that offers a structure to mine distributed data and gives specific attention to the distributed data and computing resources. Data mining in isolated environment is no more sufficient to solve the problems of analyze worldwide distributed data [4]. Recently, all over the world, the size of digital data is growing rapidly [1, 4]. Distributed environment occupies huge size of various resources of data. Analyzing those distributed data requires data mining techniques specially designed for distributed applications.

The Data mining in the distributed environment can be implemented in two different ways: Centralized Data Mining (CDM) and Distributed Data Mining (DDM) [1, 8]. The conventional method of data mining in the distributed environment was CDM, warehousing method [13]. Data in different repositories has to be collected and stored in a central location later data mining algorithm is applied on the data, which is stored in the central repository. CDM is an effortless process for establishing knowledge discovery from the distributed environment than DDM. Figure 2 shows the functional structure of centralized method. In figure 2, Local data1, local data 2, local data 3 are three different repositories in the distributed environment. Those local data repositories are suppose to be a part of one organization or multinational company. For the knowledge discovery, the data from those three local environments has to be collected and stored in the central repository. The central repository has a high volume of storage device for storing large size of digital data from different locations. The knowledge discovery algorithm will generate centralized learning model (Global model) from the data that is stored in the central system.

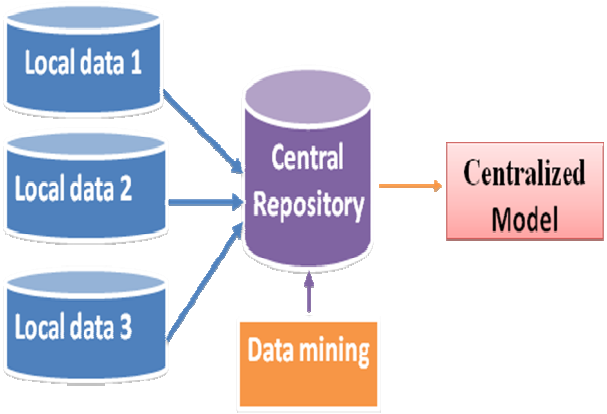

Figure 2. Functional structure of Centralized Data Mining method

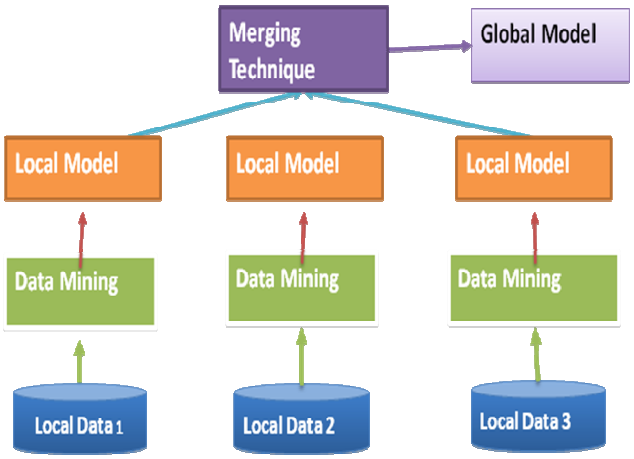

Figure 3. Functional structure of Distributed Data Mining Method

Some of the drawbacks of CDM are higher computational cost, higher communicational cost, higher implementation cost, and privacy issues. Size of the data at each repository should be very large therefore communicational cost of transmitting the data from local repository to central repository is higher. Correspondingly, computational cost of knowledge discovery process should be higher since large size of data has to be processed at central system. Another main issue in CDM is the privacy of data [8]. Data in each location may have some confidential information that is based on their strategy or peculiar ideas. When each node transfers their original data to central location then that will cause to lose the privacy of that data. The implementation of CDM needs high bandwidth of the communication channels and large volume of storage device is required in the central repository because of large size of digital data.

DDM is an advanced method for data mining in the distributed environment. Even though DDM and CDM are compatible to manage similar kind of problems; however, DDM is more efficient to overcome the drawbacks of CDM. In DDM, knowledge discovery algorithm is applied to each distributed locations separately [9]. Instead of sending data to central location, in DDM each node sends discovered knowledge (local model) to central location. Each location of the distributed sites have different sets of discovered knowledge therefore in the central system will collect all discovered knowledge and combine it form centralized knowledge (Global model). The main challenge in DDM is in proper concatenation of locally discovered knowledge. Figure 3 shows the basic functional structure of DDM. Three different data stores are local data1, local data2, and local data 3 in the distributed environments. Here, data mining algorithms applies to each location autonomously; as a result those three locations will have individual local models. The central system uses some construction strategies to combine those three local models to form global model.

The advantages of the DDM is the lesser communication cost, lesser computational cost, no privacy issue, and lesser implementation cost. In DDM, each node discovers knowledge from the available data on that location. The size of the discovered knowledge (local model) is smaller than the size of the local data therefore the communication cost of transferring the local models from distributed site to the central location is lower. Data mining algorithm applies to the local data in a parallel fashion; due to this the mining algorithm can finish with a shorter time and less number of training iteration. Since, each data set is processed separately; learning models will be more general than the centralized learning model. In DDM, local data is secured and has higher privacy because data is processing only in the local sites and discovered knowledge is sent to the central node. High bandwidth of communication channels aren't required because of small size of models are sent to central site. Likewise huge volume of storage devices are also not necessary in the central system hence the implementation cost of the DDM is less.

IV. GRID DATA MINING

Grid computing is the next generation of distributed and parallel computing technologies. Grid integrates the technologies of both distributed and parallel computing and it can manage more complex distributed data mining tasks. Grid computing delivers higher throughput computing by taking the benefits of other computers, which are connected by a network and thus, grid computing is viewed as virtual computing [10]. Under distributed computing, one or more resources is shared by other resources (computers) in the same network, hence every resource in the network is shared in grid computing [1, 2]. Grid computing is a distributed heterogeneous computer network with storage and network resources, which gives secure and feasible access to their combined capabilities. Grid environment makes it possible to share, transfer, explore, select and merge distributed heterogeneous resources. In grid, all computer resources in the network are connected together and share their computing capability to elaborate the computing power like a supercomputer so that the users can access and leverage the collected power of all the computers in the system [1]. Grid computing can increase the efficiency, decrease the cost of

computing by reducing the processing time, optimize resources, and distribute workloads. Therefore, users can achieve much faster results on massive operations at lower costs.

The grid platform has the facility to apply parallel computing and dynamic allocation of resources. Decentralized method (distributed) for data mining is suitable for grid based DM. Grid platform can offer data management services and computations for distributed data mining process of parallel data analysis and decentralization. The goal of grid computing is to create distributed computing environment for organizations and provide application developers the ability to utilize computing resources on demand.

V. ROC CURVE

Receiver operating characteristics (ROC) curve is the graphical representing of the performance of classifiers [11]. ROC graph is useful for classifying, characterizing, selecting the classifiers depending on their performance. The ROC curve has significance for evaluating the performance of classifier system instead of only considering the accuracy [11].

An instance of a classifier has four possible conclusions such as; true positive, false positive, true negative and false negative [11, 12]. If the real class is positive and predicted class is positive then it is known as a true positive. If the real class is negative and predicted class is positive then it is known as a false positive. If the real class is positive and predicted class is negative then it is known as false Negative. If the real class is negative and predicted class is negative then it is known as true negative. Figure 4 is represents four possible outcomes of classifiers, confusion matrix.

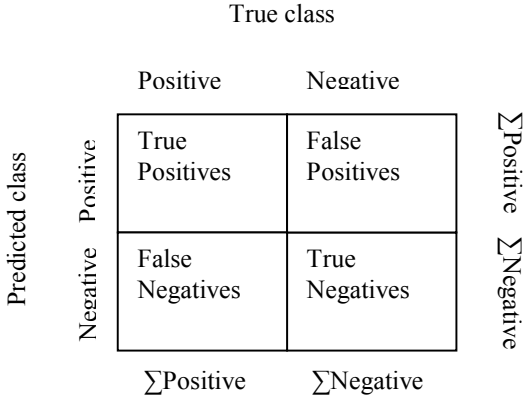

Figure 4. Confusion matrix.

For the construction of ROC curve, only true positive and false positive rates are required. The true positive rate (TPR) is based on the number of positive predicted classes are correctly matched with real positive classes, during the testing [11, 12]. False Positive Rate (FPR) is based on the number of positive predicted classes are matched with real negative classes, during the testing [11, 12]. Expression1 defines the TPR, expression 2 defines the FPR and expression 3 and 4 defines the specificity.

$$TPR = \frac{Positives\ correctly\ classified}{Total\ Positives} \quad (1)$$

$$FPR = \frac{Negatives\ incorrectly\ classified}{Total\ Negatives} \quad (2)$$

$$Specificity = \frac{True\ negative}{false\ positive + true\ negative} \quad (3)$$

$$= 1 - false\ positive\ rate \quad (4)$$

A. ROC Space

ROC graph is a two dimensional graph, FPR is defined in the' X' axis and TPR is defined in the 'Y'axis. Sensitivity is equivalent to TPR and 1- specificity is equivalent to FPR. Every predicted result makes different points in the ROC graph. Point (0, 1) on the ROC graph is knows as perfect classification since there is no false positive [11]. ROC graph also known as sensitivity Vs 1-specificity. The area under the curve represents the accuracy of the execution of the problem. The learning classifier system has specific algorithm for developing the ROC curve.

The classifiers rule sets are designed to produce only class decision (true/false), on each instance. If we implement those binary classifiers to the test set it will generate a single point on ROC curve [11]. Discrete classifiers can produce a scores (rank), a numeric value that shows the degree to which an instance is a member of a class. These ranking classifiers can be used with a threshold to produce binary classifiers. Binary classifiers can be produced based on the threshold value. If the score value is above the threshold then that classifier produces True (1) else the score value is below the threshold then it will be False (0). Each threshold value can generate separate points in the ROC curve. Therefore the threshold value can be varying from least ranking rate to maximum ranking rate ($-\infty$ to $+\infty$). Each test instance is sorted in descending order based on the rank values. The maximum ranking value may come to the point (0, 0) on the graph and least ranking value will come the point to (1, 1) on the graph.

B. Accuracy under the curve (AUC)

The performance of the classifier can be drawn using the ROC curve but for the comparison or evaluation of the ROC curve a single value is necessary [11]. The AUC is the area below the ROC curve, is part of the graph. The value of the AUC is always between zero and one [11]. ACU of the diagonal line between (0, 0) and (1, 1) of the ROC space is 0.50, sensible classifiers should have a curve above this diagonal line.

Table II shows the example of the ROC scores and class values of a curve which is derived from the 11 multiplexer problems. Twenty test instances are considered and the score

value is varying from 0.629 to 0.8987. Figure 5 shows the ROC curve that is based on the scores that are shows in table 2. Tom Fawcett was developed an algorithm for the construction of the ROC curve [11]. This curve was developed based on his algorithm. When the value of class is one then graph will move to vertical direction similarly when the class is zero then graph will move to horizontal direction. In the table12, classes values of first three scores are 1 hence in the graph first three points are plotted in the vertical direction. The fourth class is zero therefore the fourth point is plotted in the horizontal direction. The distance of each point is derived based on the number of test instances.

TABLE I. SCORES AND CLASS VALUSE OF THE ROC CURVE

Index	Score	Class
1	0.8987584	1
2	0.8961339	1
3	0.8776377	1
4	0.8637962	0
5	0.8574706	1
6	0.8573441	1
7	0.8572847	1
8	0.8462338	0
9	0.8321522	0
10	0.8251410	0
11	0.8217107	0
12	0.8108142	1
13	0.8107230	0
14	0.8103770	1
15	0.8057387	0
16	0.7786200	0
17	0.7649651	1
18	0.7125923	1
19	0.6850417	0
20	0.6295575	1

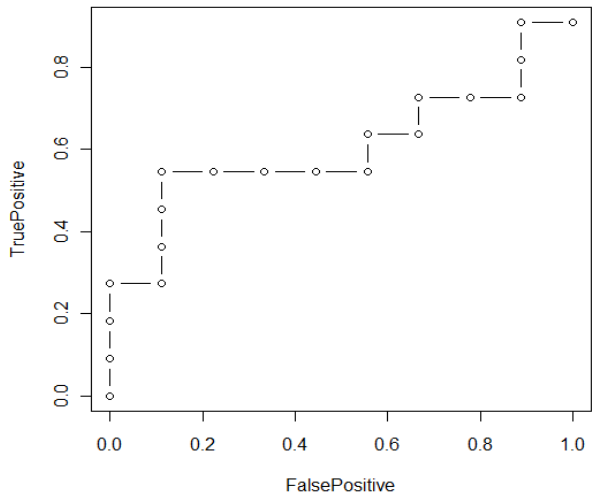

Figure 5. ROC graph is generated based on thesorted score value.

VI. UTILITIES OF GRIDCLASS[TK]

The main modules of the toolkit include facilities to create new projects; update the projects and to execute the gridclass system. This section explains the utilities of the toolkit in more detail.

A. Creating New Project

Creating a new project is the beginning step of the execution of the gridclass system. First, user can generate a name of the project and then decides the number nodes in the grid environment. After this, user has to provide different set of data and configuration file according to the number of nodes. Three set of data sets are required, they are training data, testing data, predefined classifier file. Each learning classifier system has to be configured with different parameters, which are provided by the configuration file. For the execution of the learning classifier system the training and testing data files are necessary and the predefined classifier file is not necessary. The predefined classifier file is one learning model which is generated previously when the system processed the same type of data. If the user provides the predefined classifier file then the learning classifier system can generate a more generalized local model (learning model) at each site of grid environment. The training and testing file must be in CSV format and predefined classifier file is in XML format.
The basic configuration parameters include:

- *Population size,*
- *Online learning,*
- *Noise,*
- *Probability of class Zero,*
- *V (fitness value),*
- *GA Threshold,*
- *Mutation Probability,*
- *Cross over Probability,*
- *Inexperience penalty,*
- *Inexperience threshold,*
- *Covering probability,*
- *ThetaSub,*
- *ThetaSubAccuracyMinimum,*
- *Theta deletion fraction,*
- *Theta deletion,*
- *Number of Iteration.*

The *population size* means the size of the learning model. In supervised classifier system the size of the learning model is set by the user. *Online learning* is a Boolean value and it decides whether it is online learning or offline learning. Noise is another variable, probability of class noise being added to every example of the training data. *Probability of class zero* is used to balance the class distribution in the training data. Supervised classifier system support only two values in the action class either zero or one (Yes/ No). If the probability of class zero is 0.50 then it gives equal distribution of class zero and one to training process. *V* is another parameter used for

controlling fitness evaluation in supervised learning classifier system. The parameter value *GA_ Threshold* is used to decide when genetic algorithm needs to be triggered. When the average experience of a classifiers in the correct set is above the user defined *GA_ Threshold,* then the genetic algorithm will be invoked [4, 7, 13, 14]. The parameter *mutation probability* is the probability of mutating a single point in a rule condition. The parameter *crossover probability* value decides the crossover point on a rule condition. The parameter inexperience penalty is used for fitness function. When the experience of the classifier is too low, then multiply by inexperience penalty value with fitness value to discount. The parameter, *inexperience threshold* it is also part of fitness function. Inexperience threshold value is used to compare the experience of classifier during the fitness function. Covering probability is the probability value for covering operation. The parameters *ThetaSub ThetaSubAccuracyMinimum* are the part of subsumption function. The parameters Theta deletion fraction, Theta deletion are employed in deletion function. Those parameters theta deletion and theta deletion fraction can recommend which classifier has to be deleted. The last parameter *number of iteration* decides the number of training iteration. Table 2 shows the parameters and its default values. The Figure 6 shows the node entry form of creating new project.

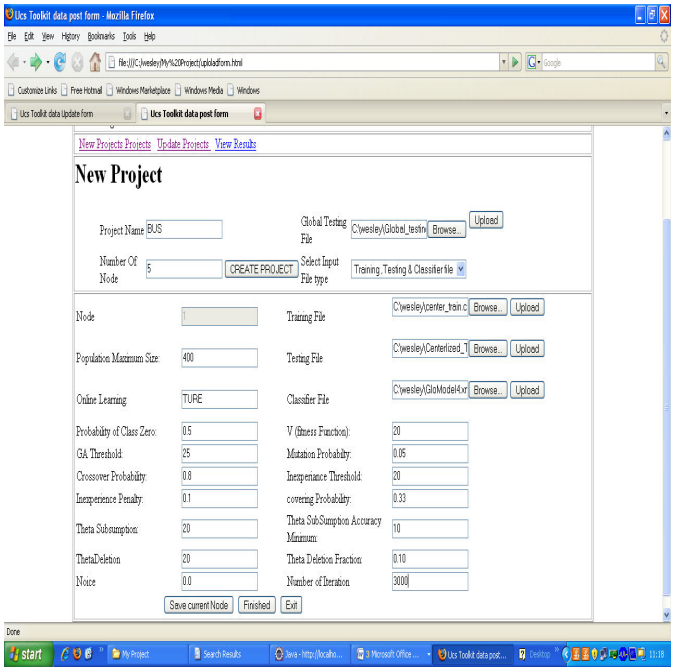

Figure 6. Node entry form.

TABLE II. SUPERVISED CLASSIFIER PARAMETERS AND DEFAULT VALUES

Parameters	Default value
Covering Probability (γ)	0.33
Crossover Prob (χ)	0.8
GaThreshold (Θ_{GA})	25
inexperiencePenalty	0.01
mutationProb (μ)	0.05
Noise	0.0
Onlinelearning	TRUE
PopMaxSize (N)	400
Probabilityofclasszero	0.5
ThetaDel (Θ_{del})	20
ThetaDelFrac	0.10
ThetaSub(Θ_{Sub})	20
V	20

B. Update the project

Update the project is one useful part of the toolkit, because it helps the user to save the time and the memory of the system. When user wants to test the same data with different configurations or different data sets with same configuration, then he can use same project file. Otherwise, the user needs to enter all required data and configuration file each time. All these project information are stored in the XML file. If the user wants to store all the information separately, then the size of the file will elaborate. Figure 7 shows the updating form.

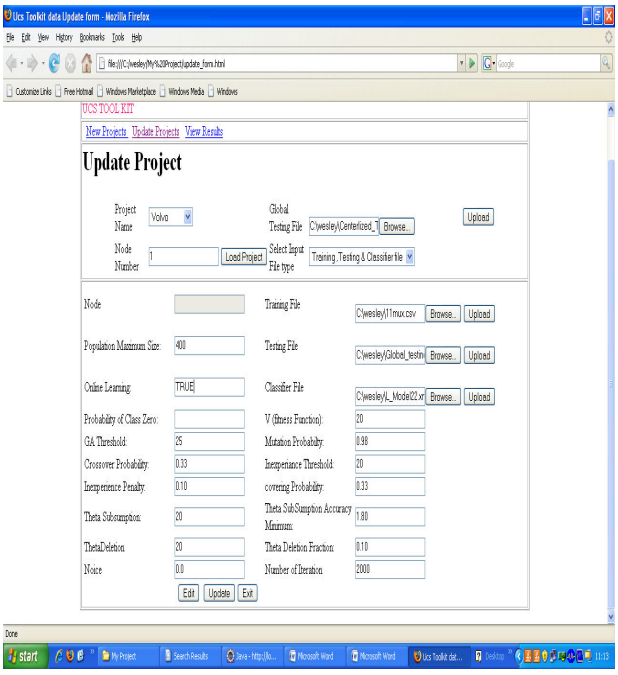

Figure 7. Updating form.

C. *Executing of grids class system*

Remote execution of the gridclass system and evaluate the performance are the main intention of the toolkit. The execution stage of the toolkit has two options one is execution of the project and second one the evaluation of the results and compare the performance of different executions.

For the execution of the project, user has to specify the name of project and calls the gridclass system. Toolkit will send all the data and configure the file that belongs to the project name to the central node of the gridclass system. The central node of the gridclass system will distribute the data and information of the configuration to the entire grid environment, based on the number of nodes specified in the configuration file. When the central node receives local models from each site of the grid environment, and it will generate global model. The central node will test this global model using separate testing file, which is provided by the user. Finally, global model and all the local model and their testing accuracies are send to the toolkit from the central node. All the local models and global models are stored in the XML format. Toolkit can generate ROC graph from the collected results. So user can evaluate the performance of each global model based on the AUC. This facility is more convenient to evaluate the performance of global model and each local model separately. Local models are representing the portion of whole problem, but the global model is the representation of complete problem. Therefore local models and global models have significant for evaluating the problem. Stored local models can be used as the predefined classifier file for future execution which is another advantage of local models. Figure 8 shows the execution form.

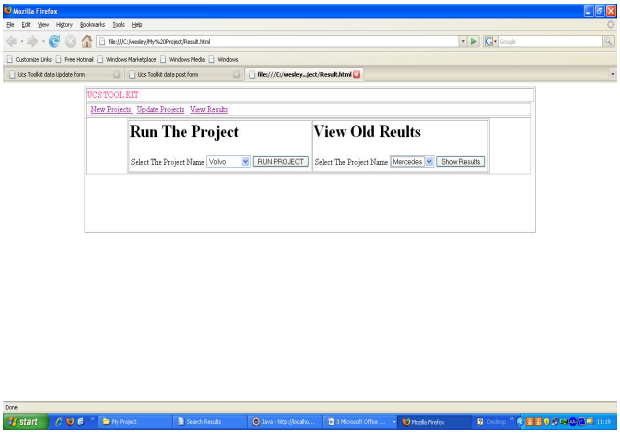

Figure 8. Execution form.

The performance of the global model and the local models are depending on the configurations of each node in the grid environment. This work gives more important of the performance of learning classifier system; therefore evaluate the performance of learning classifier system with different parameter has significant role. Here, user can compare the results of different executions based on the project name. So the end user can evaluate the global model and global model testing accuracy from the toolkit.

VII. Discussion

The toolkit for learning classifier system for grid data mining is very useful component in Grid class system. The user requires easy access to the gridclass system and further he can access this system from the remote location. Gridclass system was developed by R environment; hence it is not easy to set the configuration parameters and data for all the nodes without toolkit. Toolkit has graphical user interface that makes more flexibility to access the system. Another main advantage of this toolkit is the facility to modify the existing project. That assists user to save time and makes easier to do the experimental work. Toolkit gives the opportunity to test the global model and all the local models with the accuracies of each model. User can compare the results and configuration of one project with results and configuration of another project. System does not provide any specific algorithm for matching two results of two executions that is one inconvenience of this system. Therefore manual comparison is required.

VIII. Conclusion and Future Work

The toolkit of learning classifier system for grid data mining is the first version. This interface is more convenient and user friendly. This system has only supported one learning algorithm, but the future system will be implemented as a common interface for many learning algorithms. Also the future system will be developed good methods for comparing the performance of different experiments and results.

Acknowledgment

The authors would like to express their gratitude's to FCT (Foundation of Science and Technology, Portugal), for the financial support through the contract GRID/GRI/81736/2006.

References

[1] M. F. Santos, W. Mathew, T. Kovacs, H. Santos: Supervised Learning Classifier system for grid data mining. Proceeding of the International conference on computer and Informational Science, Huston, USA, page 416- 424, 2009.

[2] M. F. Santos, W. Mathew, T. Kovacs, H. Santos, A grid data mining architecture for learning classifier system. WSEAS TRANSACTIONS on COMPUTERS Volume 8, 2009 ISSN: 1109-2750.

[3] M.Cannataro, A. Congiusta, A. Pugliese, D.Talia, P. Trunfio, Distributed Data Mining on Grid: Services, Tools, and Applications. IEEE TRANSACTIONS ON SYSTEM, MAN, AND CYBERNETICS- PART B: CYBETNETICS, VOL. 34 NO6, DECEMBER 2004.

[4] M. F. Santos, W. Mathew, and H. Santos: GridClass: Strategies for Global Vs Centralized Model Construction in Grid Data Mining, Proceeding of the workshop on ECAI, Lisbon 2010.

[5] J. Luo, M. Wang, J. Hu, Z. Shi, Distributed data mining on Agent Grid: Issues, Platform and development toolkit. Future Generation computer system 23 (2007) 61-68.

[6] J. Aguilar, A Web Mining System. WSEASTRANSACTION on INFORMATION SCIENCE AND APPLICATIONS. ISSN: 1790-0832, Issue 9, Volume 6, September 2009.

[7] H. H. Dam, A scalable Evolutionary Learning Classifier System for Knowledge Discovery in Stream Data Mining, M.Sci. University of Western Australia, Australia, B.Sci. (Hons) Curtin University of Technology, Australia. *Thesis work 2008.*

[8] C. Clifton, M. Kantarcioglu, J. Vaidya, X.Lin, M. Y. Zhu, Tools for Privacy Preserving Distributed Data Mining. Sigkdd Explorations volume 2, 2002.

[9] N. Zhang, H. Bao, Researchon Distributed Data Technology Based on Grid. First International Workshop on Database Technology and Applications 2009.

[10] V. Stankovski, M. Swain, V. Kravtsov, T. Niessen, D. Wegener, J. Kindermann, W. Dubitzky, Grid-enabling data mining applications with DataMiningGrid: An Architectural perspective. Future Generation Computer System24, 256- 279, 2008.

[11] T. Fawcett, An introduction to ROC analysis. Pattern recognition letters 27 861- 874, 2006.

[12] http://en.wikipedia.org/wiki/Receiver_operating_characteristic, consulted on 15- 12 – 2010.

[13] A. Orriols-Puig, A Further Look at UCS Classifier System. *GECCO'06,* July 8–12, 2006, Seattle, Washington, USA.

[14] K. Shafi, T. Kovacs, H. A. Abbass, W. Zhu, Intrusion detection with evolutionary learning classifier system. Springer Science+ Business Media B. V. 2007.

[15] A. Orriols, EsterBernado-Mansilla, Class Imbalance Problem in UCS Classifier System: Fitness Adaptation..,Evolutionary Computation, 2005. The 2005 IEEE congress on, 604-611 Vol.1 2005.

[16] T. Kovacs, Strength or Accuracy: Credit Assignment in Learning Classifier Systems. Distinguished Dissertation,. Springer-Verlag London limited 2004.

[17] M. F. Santos, Learning Classifier System in distributed environments, University of Minho School of Engineering Department of Information system. PhD Thesis work 1999.

[18] M. Liu, K. Gao, High Efficient Scheduling Mechanism for Distributed Knowledge Discovery Platform, WSEAS TRANSACTIONS on INFORMATION SCIENCE AND APPLICATIONS. ISSN:1790-0832, Issue1 Volume6 January 2009.

[19] A. J. S. Santiago, A. J. Yuste, J. E. M. Exposito, S. G. Galan, J. M. M. Marin, S. Bruque, A dynamic- balanced scheduler for Genetic Algorithms for Grid Computing. WSEAS TRANSACTON on COMPUTERS. ISSN: 1109- 2750, Issue 1 Volume 8 January 2009.

[20] C. Shen, H. Chuang, A Study on the Applications of Data Mining Techniques to Enhance Customer Lifetime Value. WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS. ISSN: 1790- 0832, Issue 2, Volume 6, February 2009.