Bayesian Networks to Predict Data Mining Algorithm Behavior in Ubiquitous Environments

Aysegul Cayci, Santiago Eibe, Ernestina Menasalvas, and Yucel Saygin*

Sabanci University, Istanbul, Turkey,
Facultad de Informatica, Universidad Politecnica, Madrid, Spain
aysegulcayci@su.sabanciuniv.edu
{emenasalvas,seibe}@fi.upm.es
ysaygin@sabanciuniv.edu

Abstract. The growing demand of data mining services for ubiquitous environments motivates deployment of data mining algorithms that use context to adapt their behavior to present circumstances. Despite the efforts and results so far for efficient parameter tuning, there is a need to develop new mechanisms that integrate also the context information. Thus, in this paper, Bayesian networks are used to extract the effects of data mining algorithm parameters on the final model obtained, both in terms of efficiency and efficacy in a given situation. Based on this knowledge, we propose to infer future algorithm configurations appropriate for situations. Instantiation of the approach for association rules is also shown in the paper and the feasibility of the approach is validated by the experimentation.

Key words: automatic data mining, data mining configuration, ubiquitous data mining

1 Introduction

Ubiquitous computing, still an immature computing paradigm, brings new challenges to software designers and also to designers of data mining software. In ubiquitous computing, processing takes place on the restricted resource devices that are embedded or spread in the environment which is subject to change all the time. This means that, ubiquitous computing paradigm implies lack of expert involvement on tuning the software where it is most needed due to scarcity of resources and variability of the context. Consequently, automatic configuration is required under changing context and resource constrained environments.

There is an increasing demand for intelligent applications on ubiquitous devices while data mining methods has been the main way to provide such intelligence. As an example, knowledge discovery and data mining can be an enabling

* This work has been partially financed by Spanish Ministry of Innovation under project TIN2008-05924.

technology for more adaptive, dynamic, and autonomous social networking. Consider the scenario where mobile devices have the data mining ability for predicting the current activity and even the mood of the user. Combined with context information such as location and time enriched with more semantics, the mobile device can offer recommendations to the user taking into account the state of the other people in the social network of the user. For example if the user is at home but still not tired and in a jolly mood, the mobile device can communicate with the (mobile devices of the) friends of the user who are in the vicinity and in a similar mood to form an ad-hoc going-out group. Such an application will require mining of the data collected through various sensors to recognize the activity of the user and his/her mood taking into account the context as well.

Nevertheless, important challenges need to be addressed on the ubiquitous data mining design, which two of them will be focused on in this paper. First of all, the factors such as the context and the resource limitations of the device should be considered when deciding how to configure the data mining algorithm. Secondly, there is a need to develop methods for the autonomous and adaptable execution of data mining. Several context-aware and resource-aware data mining approaches have been proposed in the literature to deal with the issues posed due to ubiquity ([11] [12]). The proposed approaches consider the current context and/or resource availability to adjust parameters of data mining process or to determine the configuration setting of data mining in order to make autonomous decisions. However, in these approaches, knowledge from the past experiences is not incorporated into the decision mechanism of the parameter setting. As a result, settings are not adaptable in the sense that they do not improve over time based on past experience. A mechanism that adapts the data mining algorithm's configuration setting decisions according to the experiences learned, is lacking. In order to fulfill this deficiency, we propose to use machine learning techniques for deciding parameter settings according to past behavior of the algorithm under a given situation.

Automatic parameter tuning research area has gained much interest in the recent years. A number of studies have been published offering optimization and machine learning techniques to solve the problem. The main idea behind the optimization techniques is to determine the performance criteria to be optimized and the configuration that satisfies best this criteria. Optimization methods proposed for automatic parameter tuning are as follows: racing algorithm by Birattari et al ([4]); iterated local search approach by Hutter, Hoos and Stutzle ([15]); algorithm portfolios paradigm by Gagliolo and Schmidhuber ([10]); experimental design combined with local search by Diaz and Laguna ([9]). Other prevailing technique proposed for automatic parameter tuning is based on machine learning classifiers. In general terms, classifiers are used to learn the parameters to set the configuration. Srivastava and Mediratta ([20]) suggest usage of decision trees for automatic tuning of search algorithms. Through classification of previous runs of the algorithm by means of Bayesian network, Pavon, Diaz and Luzon ([18]) have automatized the parameter tuning process.

Interest on automatic parameter tuning originates in alleviating configuration setting of algorithms with a plethora of parameters most of the time but not to provide autonomy. In general, the argument of current work on automatic parameter setting is to find a configuration regardless of the circumstances. In the current proposed methods, neither the state of the device nor the requirements of the current situation are considered. However, in the ubiquitous environments, state of device and current situation are important factors to determine the appropriate parameter settings and to make the device behave autonomously under any circumstance. In our mechanism, we consider the relationship between situations and parameters. Specifically, context in which the device is in when data mining request is received and the availability of the resources are incorporated in the parameter setting decision.

Cao, Gorodetsky and Mitkas ([6]) discuss the contribution of data mining to agent intelligence. They argue that a combination of autonomous agents with data mining supplied knowledge provides adaptability whereas knowledge acquisition with data mining for adaptability relies on past data (past decisions, actions, and so on). Our approach to provide adaptability is similar: we use machine learning approach in order to generate adaptable parameter setting decisions and enhance ubiquitous data mining with autonomy and adaptability. Our mechanism is based on Bayesian networks because Bayesian networks enable (1) the finding of the probabilistic relationships between the circumstances, parameters, and performance criteria, (2) considering several factors rather than a single criteria when determining the setting and (3) adaptability by learning from the past experiences. Pavon, Diaz, Laza and Luzon also proposed Bayesian networks for parameter tuning in [18]. The innovative feature of our mechanism is that we use not only information on parameters but also information on context and resources (what we call circumstances) to discover the appropriate configuration of a data mining algorithm for a given situation. When determining the best configuration, both efficiency and efficacy of the final model are taken into account.

The rest of the paper is organized as follows: Section 2, presents the proposed approach and instantiates it for a case. In Section 3, we explain the experiment that we performed in order to validate of the proposed approach. Finally, future work is described in Section 4.

2 Proposed Approach

We present a mechanism to determine the algorithm configuration in a resource-aware and context-aware manner with respect to a data mining request issued. The mechanism is based on learning from past experiences, that is, learning from the past executions of the algorithm in order to improve the future decisions.

2.1 Analysis of the Problem

The main objective can be stated as "to determine automatically the configuration of a data mining algorithm which will run on an ubiquitous device". This objective requires understanding the elements influencing the solution:

- the resources that the algorithm needs in order to accomplish its task,
- the algorithm parameters to determine their effect on the resource usage and the efficacy of the data model,
- the context features which may have an effect on the efficacy of the data mining model and the efficiency of data mining,
- the semantics of data,
- the features of mining data set,
- the quality indicators which show the efficacy of the data mining model and efficiency of data mining.

In addition, the most important issue is how to change or improve configuration setting decisions as the circumstances change. The decisions must be adaptable to changing conditions as a data miner expert adapts his decisions when the conditions change.

2.2 Bayesian Networks

Bayesian networks which represent the joint probability distributions for a set of domain variables are proved to be useful as a method of reasoning in several research areas. Medical diagnosis([3]), language understanding ([7]), network fault detection([14]) and ecology([2]) are just a few of the diverse number of application areas where Bayesian network modeling is exploited.

A Bayesian network is a structure that shows the conditional dependencies between domain variables and may also be used to illustrate graphically the probabilistic causal relationships among domain variables. A Bayesian network consists of a directed acyclic graph and probability tables. The nodes of the network represent the domain variables and an arc between two nodes (parent and child) indicates the existence of a causal relationship or dependency among these two nodes. Associated with each node there exist a probability table (PT). If the node has no parents, its probability table contains the prior probabilities else the conditional probabilities between the node and its parents. Although the domain variables can be continuous, they are discretized most of the time for simplicity and efficiency. Besides representing the dependencies between domain variables, a Bayesian network is used for inferencing the probability of a variable given the observations of other variables. In depth knowledge on Bayesian networks can be found in [19].

Learning the Bayesian network structure from data rather than drawing the structure by analyzing the dependencies of domain variables, is a field of research which was studied extensively. Algorithms that learn the structure are most useful when there is a need to construct a complex network structure or when domain knowledge does not exist as in an ubiquitous environment. A discussion of the literature can be found in [5].

2.3 Basis of the Approach

We propose machine learning as the main mechanism to learn from past executions of data mining algorithms. Without a loss of generality, we selected Apriori [1] as the prototype algorithm for automatic configuration but the proposed mechanism is also applicable to any other algorithm. In particular, we propose to construct a Bayesian Belief Network using the information collected during algorithm's previous executions in order to predict future behaviors. The content of the information is determined by considering the elements that influence the solution of the problem (given in subsection 2.1). This information which is recorded as history of execution records is divided into three groups:

- Circumstantial: Information about the conditions of the resources and the context states that is obtained prior to execution of the algorithm.
- Configuration Parameters: The values that the algorithm parameters receive.
- Quality Measurements: Efficiency and efficacy related information. They include resource consumption measurements such as duration of the data mining, average memory used or indicators of model quality such as minimum support or confidence of an association rule model.

Groups are determined by taking into account the relationships that we want to examine by building a Bayesian network. Moreover, the content of the groups cover all the elements given in subsection 2.1 except the ones related to the data to be mined as we focus on the ubiquitous aspect in this work.

Once the Bayesian Network is built, the steps that lead to automatic parameter configuration are as follows (See Fig. 1 for details):

- Association rules are needed for a specific data set,
- Current circumstance and the quality requirements of the data model are determined to be used for the configuration setting,
- The algorithm which will discover the association rules is configured autonomously by inferencing from the Bayesian Belief Network that represents circumstances, parameters and quality measurements,

In the next subsections we describe in more detail the process of building the Bayesian network.

2.4 Definitions

Algorithm Configuration: Configuration of the algorithm is defined by a set of ordered pairs *(p,v)* where p stands for a parameter and v is the value it takes.

Circumstance: Circumstance is defined by a set of ordered pairs *(f,s)* where f is either a resource or context feature and s is the state of this feature.

Quality Criteria: Quality criteria is defined by a set of ordered pairs *(q,l)* where q is a quality measurement and l is the required level for this measurement. Quality measurements are metrics of efficiency or efficacy of the algorithm. Quality criteria define either the expected efficiency of data mining or efficacy of the model or both under the given circumstances.

2.5 Mechanism to Predict Ubiquitous Data Mining Configuration

We propose to build a Bayesian network using data collected from past executions of the data mining algorithm. The Bayesian network reflects the conditional dependency between the fields of execution records. Configuration is inferred from the Bayesian network. More specifically, the most likely configuration is determined from the Bayesian network by estimating the probabilities of possible configuration settings from previous Apriori runs in execution circumstances similar to current in order to obtain quality levels similar to the required.

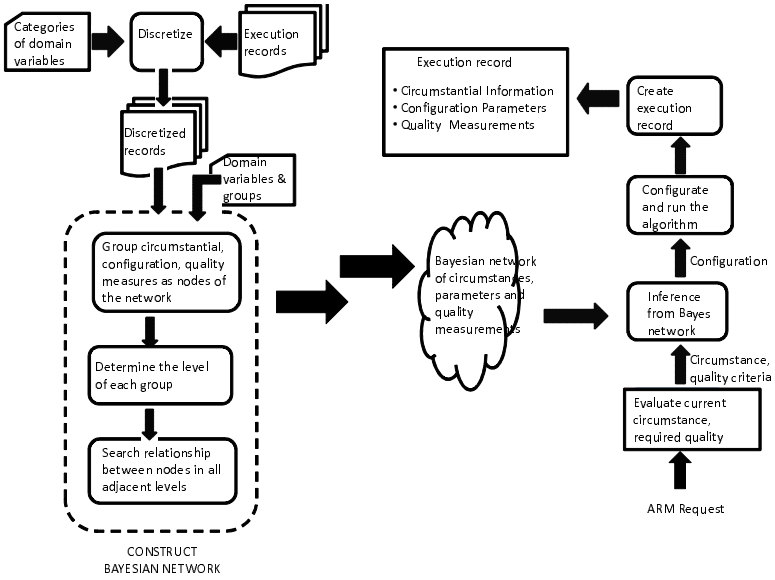

Fig. 1. Bayesian Network construction steps and data mining configuration mechanism.

We used the K2 algorithm proposed by Cooper and Herskovits([8]) as the basis for constructing Bayesian network. Before building the Bayesian network, we preprocessed the historical information and discretized the fields of execution records. We made use of the open source code of Weka software([13]) to construct the network and updated it to fit our needs. The original algorithm seeks relationships among all the variables. In our case, we have three groups of variables. The relationships among the variables within a group are not interesting. For example, we are not interested in the relationship among circumstantial variables such as *location* and *memory available*. For this reason, we modified the K2 algorithm accordingly and looked for relationships among nodes belonging to different groups of variables. Furthermore, we assigned a level to each group based on the possible cause-effect relationship between them and we used the levels to prevent the nodes in the lower level groups to be the parents of the

nodes in the upper level groups. Fig. 1 illustrates Bayesian network construction steps that we propose.

The Bayesian network that we construct from historical data represents the probabilistic relationship between circumstance states, discretized possible parameter settings, and measured as well as discretized quality measurements. Appropriate setting of an algorithm's parameter is extracted from the Bayesian network.

2.6 Instantiation of the Approach for Apriori

In this section, automatic configuration setting is instantiated for the well known association rule mining algorithm Apriori. A possible instantiation of the approach is shown by assignments made to the parameters of Apriori, circumstances and quality criteria.

Parameters. We use the parameters for the implementation of Apriori available by Weka software ([13]). The list of parameters that we selected for tuning are as follows:

- upperBoundMinSupport: upper bound for minimum support.
- delta: the factor which minimum support is decreased each time Apriori is iterated.
- lowerBoundMinSupport: lower bound for minimum support.
- numRules: number of strong association rules.
- minMetric: minimum confidence.

A possible configuration that is obtained as a result of inferences made from the Bayesian network, is as follows:
Configuration = {(upperBoundMinSupport, 0.9), (lowerBoundMinSupport, 0.5), (delta, 0.05),(numRules, 15),(minMetric, 0.9)}

Circumstance. We restrict the instantiation of circumstance to a small number of resources and context. Resources that we use are *memory* and *CPU*. Context features considered in the instantiation are *location* and *time*. We discretized the actual figures that we obtained for the availability measures and converted to states. We define five states for the resource availability measures numbered from one to five where small numbered states indicate scarcity of the resource; high numbered ones, the opposite. *Location* refers to physical location and time slots of the day are used when discretizing *time*. Some possible instantiations of circumstances are as follows:
Circumstance 1= {(memory available, 4), (location, home)}
Circumstance 2= {(CPU available, 1),(memory available, 2),(location, office)}

Quality Criteria. Quality measurements selected for the instantiation are divided into two in relation to the kind of criteria they measure (Table 1). The ones for measuring efficiency can be obtained by calling system primitives. On

the other hand, it is possible to extract the efficacy related ones from the data mining model generated. Some possible instantiations of quality criteria are as follows:
Quality Criteria 1 ={(average memory usage, 1)}
Quality Criteria 2 ={(total CPU time, 1), (number of rules discovered, 3)}

Table 1. Quality measurements for efficiency and efficacy.

Efficiency	Efficacy
Average memory usage	Minimum support of the resulting model
Maximum memory usage	Number of iterations
Total CPU time	Number of rules discovered
Total number of CPU cycles	Minimum confidence obtained
Duration	

3 Experimental Evaluation

We proposed an automatic parameter setting mechanism in which the decisions are based on Bayesian network of data mining algorithm's past executions. Our aim in conducting experiments is to validate that it is possible to establish the configuration of the algorithm by making use of the proposed mechanism. In order to do that, we created Bayesian network of Apriori executions to be used for understanding the behavior of Apriori. Besides, we employed another approach, full factorial experiment design to compare and verify the results obtained from the Bayesian network. In full factorial experiment design, we examined the effects of the parameter settings of the algorithm against a quality metric. Same data used for the Bayesian network construction is supplied to the full factorial experiment design. Finally, we compared the inferences made from the Bayesian network against the results of the full factorial experiment design. There should

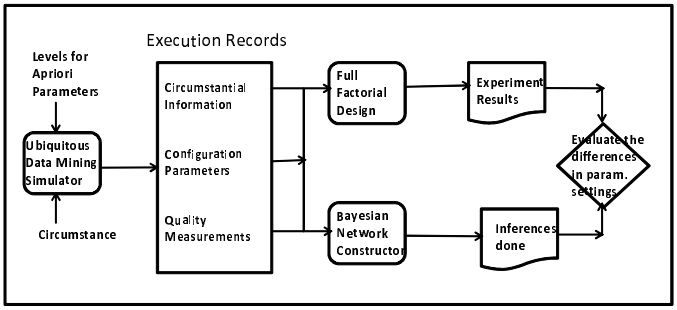

Fig. 2. Experiment phases.

be a sufficient number of Apriori executions in order to obtain a sound and reliable Bayesian network. Considering this requirement, we preferred to use full factorial design where all combinations of parameters for the determined levels are utilized during the tests. We divide the experiment into three phases:

1. Multi-level full factorial design is used to reveal the factors (parameters) that are effective on the quality indicators under a given circumstance.
2. Bayesian network is constructed in order to infer the probabilistic relationships among parameters, circumstances and quality criteria.
3. The accuracy of the proposed mechanism is assessed by comparing the results of two phases.

In order to generate historical data, we used an ubiquitous data mining simulator. Fig. 2 shows the interaction of the experiment phases and the ubiquitous data mining simulator.

3.1 Ubiquitous Data Mining Simulator and Experiment Data

We have developed a simulator in order to incorporate the features of ubiquitous data mining that we are interested in while creating records of Apriori executions for the experiment. While running Apriori, a possible circumstance consisting of resources states and context is given as input to simulator for each Apriori execution. An execution environment according to the circumstance provided is simulated. For example, if the stated resource state is the scarcity of memory, the simulator starts dummy processes to use up the memory in order to run Apriori in a memory constrained situation. In the same way, information given about context is stored in the execution record as part of circumstantial information. Briefly, the data mining simulator reads circumstance and Apriori parameters from a file, generates the resource scarcity conditions if the given circumstance requires and runs Apriori by calling Weka with the given parameters. Upon completion, an execution record is written consisting of input context fields, fields showing resources availability, Apriori parameters, resource usage fields and the data mining model.

While generating possible parameter values of Apriori, all combinations of determined levels of process factors are considered as in full factorial design. Full factorial design is one of the Design of Experiment (DoE) methods [17] which is statistically determining the effects of factors of a process to its response by systematically varying the levels of factors during testing of the process. In DoE terminology, *response* is the output variable of the process, *factors* are its input variables and *level* is a possible setting for a factor. In our case, Apriori parameters correspond to full factorial experiment design *factors* and possible settings of Apriori parameters correspond to *levels* in factorial design terminology. Hence, we determined possible settings for Apriori parameters (Table 2) and run for all combinations of levels.

We simulate an ubiquitous device by composing different availability of resource and context. Therefore we determined two groups of levels to be used

Table 2. Levels used for parameters.

Circumstance	Mnemonic	Parameter	Levels
Group 1	U	upper bound minimum support	0.7, 0.8, 0.9
	M	lower bound minimum support	0.1, 0.2, 0.3, 0.4, 0.5, 0.6
	D	delta	0.01, 0.05, 0.1, 0.15, 0.2
	N	number of association rules	1, 5, 10, 15, 20
	C	minimum confidence	0.5, 0.6, 0.7, 0.8, 0.9
Group 2	U	upper bound minimum support	0.7, 0.8, 0.9
	M	lower bound minimum support	0.4, 0.5, 0.6
	D	delta	0.01, 0.05, 0.1, 0.15, 0.2
	N	number of association rules	15, 20
	C	minimum confidence	0.8, 0.9

on different circumstances (Table 2). We run Apriori for every combination of levels so the number of distinct settings in the groups are 2250(3x6x5x5x5) and 180(3x3x5x2x2) respectively. The states of the context and resource features that we used in forming the circumstances are {*home*, *office*} and {*short on memory*, *cpu bottleneck*, *none*} respectively. All the resource states are simulated for each context state, resulting in six circumstances. Each group is used for one context state as in the first group for *home* and the second group for *office*.

3.2 Multi-level Full-Factorial Experiment Design

In this phase, we apply multi-level full factorial design using Apriori execution records generated in the manner explained in the previous section. Applying multi-level full factorial design to the history of Apriori execution data determines which Apriori parameters (factors) effect which quality measurement (response). Since we run Apriori by simulating specific circumstances, we are able to analyze the effects for each circumstance. We obtained an "effects table" of quality measurements by parameters for each circumstance.

We used experiment software Minitab([16]) to calculate the estimated effects and to plot the analysis results. In this paper, we limited our discussion of results to six quality measurements: *average memory use*, *total CPU time*, *duration*, *maximum memory use*, *total CPU cycles*, *minimum support of the model*. We determined the effects of five Apriori parameters to each of the quality measurement under a single, different circumstance. For example, when *home-memory low*, we examined the effects of five Apriori parameters to *average memory use* or when *office-CPU bottleneck*, the effects of parameters to *total CPU cycles* and so on.

Fig. 3 illustrates the full factorial design results obtained for *home-memory low*. We analyze the results for this circumstance in detail in order to explain the method. In the figure, the means of quality measurements for the utilized levels of parameters are plotted. In quadrants of Fig. 3, plots for *average memory use*, *total CPU time*, *duration* and *minimum support of the model* are given respectively. Each plot (U, M, D, N, C) within a quadrant is for a parameter. The mean of the measured value is plotted for every level we tested for that parameter in the experiment. If the plot is not flat which indicates the means of measured values vary with different value assignments of this parameter, then

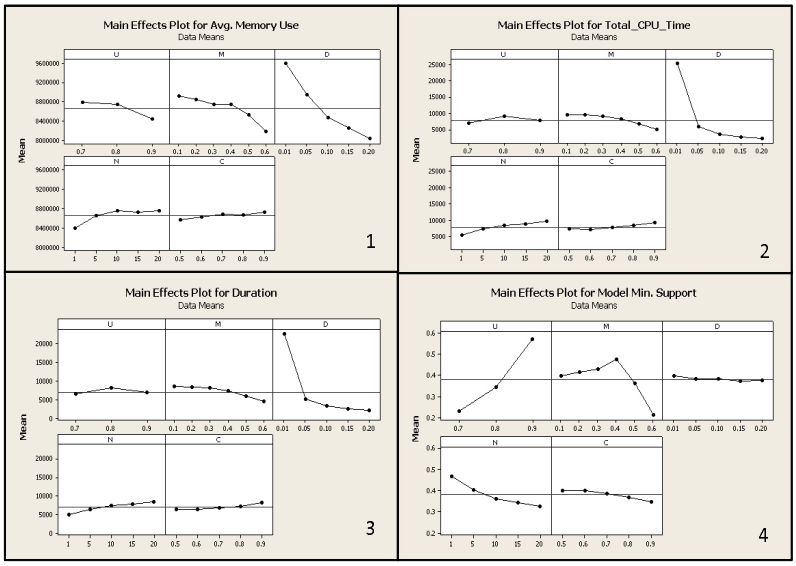

Fig. 3. Main effects plot of 4 quality measurements for *home-short on memory*.

this parameter is effective on the measured value. For example, we analyze the main effects plot for *average memory use* in quadrant 1 and we observe that D is most effective on *average memory use* since varying its value causes big differences on the mean *average memory use*. The plots of means give an insight on understanding the effect of a certain parameter to a quality measure. On the other hand, we considered the F test values that are supplied by Minitab([16]) to determine the significance of the effect.

The next step is to determine the appropriate value of the parameter which is designated as effective on the measured criteria. We choose the value that has the smallest mean of response for its factor level combinations as the appropriate value. We present the results of full factorial experiment design in Table 3 where we compare against the results of the Bayesian network.

3.3 Parameter Setting by Bayesian Network Inferences

In this phase, we apply our mechanism and obtain the results, that is, the parameter settings of Apriori from the Bayesian network. First, we explain in detail our considerations while constructing the Bayesian network before presenting the results of this phase.

Execution records generated by ubiquitous data mining simulator are first discretized, then they are used to construct the Bayesian network (Fig. 4). We made use of the K2 algorithm ([8]) but we grouped the nodes while constructing the network and searched for causal relationship among these groups of nodes.

The nodes in the upper level represent the circumstance, middle level nodes represent Apriori parameters, and finally the lowest level nodes are quality measures.

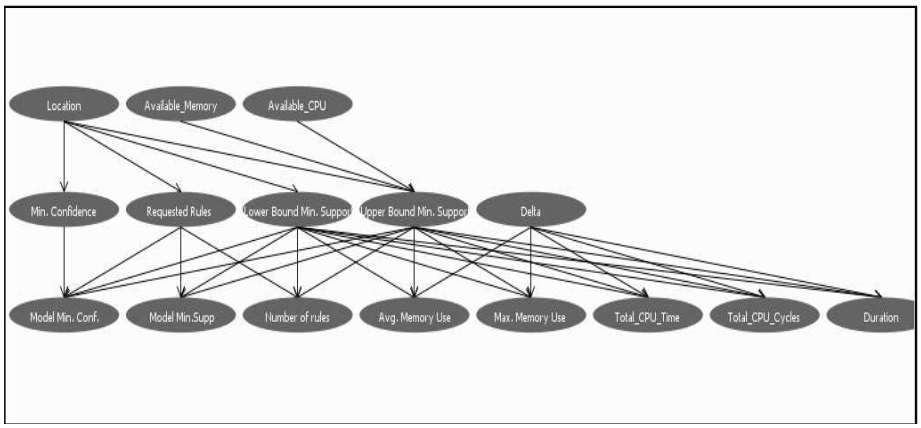

Fig. 4. Bayesian network of Apriori runs.

The cause and effect relationships between circumstances and parameters present which parameter settings are appropriate under which circumstances, whereas the cause and effect relationships between parameters and quality metrics show which parameters are effective on which quality measures.

While producing the experiment data for this Bayesian network, we did not determine appropriate parameter settings for circumstances but we ran Apriori for every combination of parameters in each circumstance because our purpose is to find the effect of parameters to quality measurements in the first place. Therefore, at this stage the relationships between circumstances and parameters is not meaningful. We assumed each circumstance variable relates to each parameter node in order to include circumstances in the inference mechanism. The relationships between the parameter nodes and quality measure nodes represent the effectiveness of parameters against quality measurements. The Bayesian network in Fig. 4 shows that *minimum confidence* and *requested rules* are related only to efficacy; *delta* to all efficiency measurements as well as *lower and upper bound minimum support*, are related to all.

We determined parameter settings of parameters by inferencing from the Bayesian network given in Fig. 4. In particular, we evaluated p(x|E) which is the conditional probability of x (parameter variable) given E (circumstantial and quality measure variables). A pseudo code of this calculation is given below. We estimate from previous Apriori runs, the support for assigning a certain value to a parameter when execution circumstances similar to current and quality levels similar to the required were observed. Given the circumstances and quality

requirements, we repeat the estimation of the most likely assignment for every parameter of the algorithm.

Pseudo code of parameter setting estimation from the Bayesian network

```
Definitions:
Let C be the set of circumstance sets Ck
     where (fkj,skj) is a feature, state pair in Ck,
            ck is the number of pairs in Ck.
    Qk is the corresponding quality criteria set of Ck
     where (qkj,vkj) is a quality measure, state pair in Qk,
            nk is the number of pairs in Qk.
    P is the set of parameters sets Px
     where pxy is a parameter setting in Px,
           rx is the number of possible settings for Px.
Pseudo Code:
     for every Ck in C
      let E be all fkj=skj (forall j <= ck) and qkl=vil (forall l <= nk)
          for every Px in P
              for every pxy in Px (forall y <= rx)
                 calculate Prxy = Probability (Px = pxy | E )
          pxy with highest Pr is the appropriate setting of Px for Ck.
```

The parameter settings that we obtained by applying the pseudo code above to the Bayesian network given in Figure 4 are presented in Table 3 and are compared against the parameter settings of full factorial design.

3.4 Comparison of Results

We propose to use Bayesian network for automatizing the parameter tuning of data mining algorithms. We discovered the relationships among circumstantial variables, parameters and quality measures to use this information for probabilistically estimating the appropriate parameter settings. In order to validate the results that are obtained from the Bayesian network, we used another approach, full factorial design to achieve the same goal. In full factorial design, regression is used to determine the effect of a parameter to a quality measure. Both of the approaches can be used to determine the parameter settings of an algorithm where the outcomes of Bayesian network and full factorial design are summarized as follows:

- Full factorial design provides
  - The list of parameters which are not effective on a quality measure
  - The parameter setting which has the highest/lowest least square mean for a quality measure
- Inference from Bayesian network provides
  - The list of parameters which are not related to a quality measure
  - Most likely parameter setting given the circumstance(s) and the quality measure(s) as evidence

In factorial design, the effects of factors can be analyzed against a single response at a time. Although, this restriction does not apply to Bayesian inferences, we used a single quality measure in order to analyze and compare the results of the Bayesian inference with the full factorial design results. The flexibility of using multiple variables on forming the evidences of inferences is one of the strengths of Bayesian network over full factorial design. We simulated a specific circumstance each time we analyzed the effects of parameters onto a single quality measure. In both approaches, we aimed to estimate the appropriate values for five parameters considering six circumstance-quality measures. Table 3 shows the results we obtained from both approaches. *Minimum confidence* is not included in Table 3, since it is observed that *minimum confidence* is not related to any of the considered quality measures in both of the approaches. The first parameter value given on each column is obtained by inferring from the Bayesian network whereas the second one is from the full factorial design (B/F).

Table 3. Parameter settings by circumstance.

Circumstance	Quality Measure		U	M	D	N
home-short on memory	average memory usage	B/F	0.9/0.9	0.6/0.6	0.2/0.2	
home-CPU bottleneck	total CPU time	B/F	0.7/-	0.6/0.6	0.2/0.2	-/1
home-no constraints	model's minimum support	B/F	0.7/0.7	0.6/0.6		20/20
office-short on memory	maximum memory usage	B/F	0.9/0.7	0.6/0.6	0.2/0.2	
office-CPU bottleneck	total CPU cycles	B/F	0.7/0.7	0.6/0.6	0.2/0.2	
office-no constraints	duration	B/F	0.7/0.7	0.6/0.6	0.2/0.2	

It is possible to say based on the results in Table 3, that in majority of the cases, parameters that are found to have effect on a quality measure under a circumstance in full factorial design, are represented as related to that quality measure under the same circumstance in the Bayesian network. The appropriate parameter settings decided in order to optimize a quality measure in full factorial design is identical in most of the cases to the parameter settings inferred from the Bayesian network given the same quality measure.

4 Conclusion

Anticipating the importance of autonomous and adaptable behavior incorporation in ubiquitous data mining enabled us to propose Bayesian network for understanding the algorithm behavior and determining the appropriate parameters. Considering the characteristics of ubiquitous environments, we stressed the usage of circumstance on determining appropriate parameter values. We also aimed at recommending the parameter settings that most likely satisfies the required quality. Thus, we analyzed the effects of parameter settings to quality measures which are related to both efficiency of data mining process and efficacy of the data mining model.

As the result of our simulation experiment, satisfactory parameter and quality measure relationships to recommend parameter settings, are formed in the Bayesian network. We also validated our proposal by comparing the parameter settings obtained from the Bayesian network against another approach, full factorial experiment design. Experiment on association rule mining shows that proposed method gives parameter settings almost identical to the optimal setting obtained from full factor analysis which is a completely different approach.

In the future, we will assess the adaptability of the proposed approach by conducting experiments using recommended parameter values obtained in this work. We aim to extend this work by constructing Bayesian network structures for alternative variety of circumstantial variables and quality measures. Assessing the accuracy and the cost of the estimation and analyzing when to update the network are still the open issues in which we also plan to work in the near future.

Acknowledgments. Authors would like to thank Can Tunca and Engin Dogusay from Sabanci University who contributed to this study by developing the supporting software.

References

1. Agrawal, R. and Srikant R.: Fast Algorithms for Mining Association Rules. In : Proceedings of the Int. Conf. on Very Large Data Bases (VLDB'94), pp. 487–499. Morgan Kaufmann, San Francisco (1994)
2. Amstrup, S. C., Marcot, B. G., and Douglas, D. C.: A Bayesian Network Modeling Approach to Forecasting the 21st Century Worldwide Status of Polar Bears. In : Arctic Sea Ice Decline: Observations, Projections, Mechanisms, and Implications. Geophysical Monograph 180, 487–499. American Geophysical Union, Washington, DC (2008)
3. Beinlich, I.A., Suermondt, H.J., Chavez, R.M., and Cooper, G.E.: The ALARM Monitoring System: A Case Study with two Probabilistic Inference Techniques for Belief Networks. In : Proceedings of the Second European Conference on Artificial Intelligence in Medicine, pp. 247–256. London (1989)
4. Birattari, M., Stutzle, T., Paquete,L. and Varrentrapp, K.: A Racing Algorithm for Configuring Metaheuristics. In : GECCO '02 Proceedings of the Genetic and Evolutionary Computation Conf., pp. 11–18. Morgan Kaufmann, San Francisco (2002)
5. Buntine, W.: A Guide to the Literature on Learning Probabilistic Networks from Data. IEEE Trans. on Knowl. and Data Eng. 8, 195–210 (1996)
6. Cao, L., Gorodetsky, V. and Mitkas, P.A.: Agent Mining: The Synergy of Agents and Data Mining. IEEE Intelligent Systems, 24, 64–72, (2009)
7. Charniak, E., and Goldman, R.: A Semantics for Probabilistic Quantifier–Free First–Order Languages with Particular Application to Story Understanding. In: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, pp. 1074-1079. Menlo Park, California (1989)
8. Cooper, G. F. and Herskovits, E.: A Bayesian Method for Constructing Bayesian Belief Networks from Databases. In: Seventh Conference on Uncertainty in Artificial Intelligence, pp. 86–94. Morgan Kaufmann, San Francisco (1991)

9. Adenso-Diaz, B. and Laguna, M.: Fine-Tuning of Algorithms Using Fractional Experimental Designs and Local Search. Oper. Res. 54, 99-114 (2006)
10. Gagliolo, M. and Schmidhuber, J.: Learning Dynamic Algorithm Portfolios. Annals of Mathematics and Artificial Intelligence 47, 295-328 (2006)
11. Gaber, M.M. and Yu, P. S.: A Framework for Resource-Aware Knowledge Discovery in Data Streams: a Holistic Approach with its Application to Clustering. In:ACM Symposium on Applied Computing, pp. 649–656. ACM, NY (2006)
12. Haghighi, P.D., Zaslavsky, A., Krishnaswamy, S., and Gaber, M.M.: Mobile Data Mining for Intelligent Healthcare Support. In: 42nd Hawaii international Conference on System Sciences, pp. 1–10. IEEE Computer Society, Washington,DC (2009)
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations, 11, (2009)
14. Hood, and C., Ji, C.: Proactive Network Fault Detection. In: Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution, INFOCOM, pp. 1147. IEEE Computer Society, Washington, DC (1997)
15. Hutter, F., Hoos, H. H., and Stutzle, T.: Automatic Algorithm Configuration Based on Local Search. In: 22nd National Conference on Artificial Intelligence, pp. 1152–1157. AAAI Press, (2007)
16. Minitab Inc., `http://www.minitab.com/en-US/`
17. Montgomery, D.C.: Design and Analysis of Experiments. John Wiley and Sons, (2006)
18. Pavon, R., Diaz, F., Laza, R., and Luzon, V.: Automatic Parameter Tuning with a Bayesian Case-Based Reasoning System. A Case of Study. Expert Syst. Appl. 36, 3407–3420 (2009)
19. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, (1988)
20. Srivastava, B. and Mediratta, A.: Domain-Dependent Parameter Selection of Search-Based Algorithms Compatible with User Performance Criteria. In: 20th National Conference on Artificial Intelligence, pp. 1386–1391. AAAI Press (2005)