Singapore Management University
Institutional Knowledge at Singapore Management University

Research Collection School Of Information Systems

School of Information Systems

5-2006

Discovering Causal Dependencies in Mobile Context-Aware Recommenders

Ghim-Eng YAP
Nanyang Technological University

Ah-Hwee TAN
Nanyang Technological University

Hwee Hwa PANG
Singapore Management University, hhpang@smu.edu.sg

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

Part of the Databases and Information Systems Commons, and the Numerical Analysis and Scientific Computing Commons

Citation

Discovering Causal Dependencies in Mobile Context-Aware Recommenders

Ghim-Eng Yap, Ah-Hwee Tan
School of Computer Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798
{yapg0001, asahtan}@ntu.edu.sg

Hwee-Hwa Pang
School of Information Systems
Singapore Management University
80 Stamford Rd, Singapore 178902
hhpang@smu.edu.sg

Abstract

Mobile context-aware recommender systems face unique challenges in acquiring context. Resource limitations make minimizing context acquisition a practical need, while the uncertainty inherent to the mobile environment makes missing context values a major concern. This paper introduces a scalable mechanism based on Bayesian network learning in a tiered context model to overcome both of these challenges. Extensive experiments on a restaurant recommender system showed that our mechanism can accurately discover causal dependencies among context, thereby enabling the effective identification of the minimal set of important contexts for a specific user and task, as well as providing highly accurate recommendations even when context values are missing.

1. Introduction

A *recommender* [20] is an application that ranks a set of available choices with respect to certain criteria. There are many well-studied recommenders, such as within the information retrieval field, where criteria are submitted as queries and the most relevant documents are recommended. Today, with the proliferation of e-services, recommenders are an actively researched area due to their obvious commercial values (e.g. [22], [10]), and have proven themselves as an important enabling technology behind the successes of major e-commerce sites like Amazon.com.

Two main recommender techniques in use today are the *content-based* and the *collaborative filtering* approach [17], although hybrid systems do exist [1]. These traditional approaches do not take into account situational information, and this seriously limits the relevance of their results. For instance, a user query to a restaurant recommender could be "*restaurants selling vegetarian food*". A traditional recommender would simply check if vegetarian food is available, oblivious to the fact that the user would have preferred a nearer eating place because of the rain outside. Clearly, the system's recommendations are impeded by its detachment from the situation of use. What is lacking in the traditional systems is an *awareness of the context*. According to [5],

[Context is defined as] any information that can be used to characterize the situation of an entity, where an entity can be a person, a place, or an object relevant to the interaction between the user and application, including the user and application themselves. A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.

It is recognized that the recommenders possess wide applications for mobile e-commerce, or *m-commerce* [22, 24], and recent years have seen a growing interest in *context-aware recommender systems* (e.g. [16], [1], [10], [9]). However, *mobile context-aware recommenders* poses certain unique challenges that remain to be addressed, particularly with respect to the acquisition of contextual data.

A key challenge is to accurately identify the *minimal* set of context important to a particular user for a particular task. Mobile devices suffer serious limitations in resources including bandwidth and battery life. As such, by identifying and acquiring just the minimal set of context that are truly important for recommendations to a particular user, mobile devices can save on unnecessary context acquisition costs and hence conserve their precious resources for other tasks.

A second key challenge to the acquisition of context is that of frequently missing context element values. Within the volatile mobile environment, lapses in context acquisitions occur due to various reasons such as failure to negotiate access control over protected information, intermittently faulty sensors, and also broken or unstable communication links. Explicit encoding of context dependencies is required to enable mobile recommenders to maintain optimal predictions even when crucial nuggets of information are missing.

For the first challenge, a common approach in the literature was to treat it as a *feature selection* problem and to use statistical techniques to identify the set of context elements to be retained [1]. Others like [24] suggested allowing users to explicate their preferences via rules like "*recommend me a cafe based on location, but not prices*". Our earlier work

in optimizing context within recommender systems made use of Support Vector Machines to identify the optimal context for a specific recommendation task and user [29].

For the second challenge, no existing work has considered modelling the *interdependencies among context* for handling missing context values. By learning these interdependencies, we can better understand and hence explain the behaviors of the system in terms of the interactions among context. This enables us to both readily identify the minimal set of context parameters applicable to a particular user and task, as well as to make use of the discovered context interdependencies for compensating missing context values.

Our solution to this problem lies in *Bayesian networks*. A Bayesian network is a directed acyclic graph that encodes the complete causal dependencies among context variables. From this graph, we can interpret the dependencies between a context like the "*availability*" of an item and the suitability or "*score*" of that item. Thus, the minimal set of context that is important to a particular user and task would simply be those that are directly connected to the "*score*" node. In addition, encoded dependencies among context enable the predictions on the outcome by using the available information to estimate missing context. In contrary, most other standard supervised learning techniques such as *Decision Trees* and *Support Vector Machines* yield inaccurate predictions when crucial inputs are missing as they do not encode these important context dependencies.

As state-of-the-art Bayesian network learning schemes are slow when the number of context is large, we propose a learning mechanism that makes use of task-specific *markers* within a tiered context model to make learning *faster* (learning from fewer observations effectively) and more *scalable* (handling more context parameters efficiently). With these enhancements, Bayesian network learning and prediction provides a natural and integrated solution to the above challenges faced by mobile context-aware recommenders.

In section 2, we provide an overview to the Bayesian networks, the *CaMML* program [27] that we adopted for learning Bayesian networks from observations, and also the prior work that are related to this paper. In section 3, we present our two-tier context learning solution for resolving the key challenges faced by mobile context-aware recommenders. We present in section 4 a restaurant recommender application that implements our context learning model. Lastly, we present our experimental protocols and results in section 5, and conclude with a discussion of future work in section 6.

2. Background

We chose Bayesian networks for modelling causal dependencies within a context-aware recommender system due to their natural ability to address the key challenges as highlighted previously. In particular, we employed the *CaMML* Bayesian network learning tool [27] to discover causal de-

pendencies among context parameters for a particular user and task. We now describe briefly Bayesian networks and *CaMML*, and also some of the related prior work.

2.1. Bayesian Networks

Pearl [19] concisely defined Bayesian networks as follows.

Formally, Bayesian networks are directed acyclic graphs in which each node represents a random variable, or uncertain quantity, which can take on two or more possible values. The arcs signify the existence of direct causal influences between the linked variables, and the strengths of these influences are quantified by conditional probabilities.

Bayesian networks are also known as *Belief Networks*, since they are actually probabilistic models that encode the extent to which each variable is believed *to affect and be affected by* others. Knowledge about the dependencies among the key variables of any domain can be effectively modelled as a system of connected nodes within the directed acyclic graphical model of *Bayesian Network*. These causal dependencies are captured qualitatively by the network structure, and quantitatively by the *beliefs* associated with each node.

Heckerman [8] highlighted that because the local distribution functions in Bayesian networks were in fact classification/regression models, Bayesian networks would be identical to classification/regression approaches given complete data for predictions. Heckerman cited work [6, 21] that argued, either based on bias-variance analysis or via empirical means, that neither decision trees nor Bayesian networks would dramatically outperform the other. However, he also agreed that Bayesian networks provided a natural model for learning about and encoding the dependencies among input variables. Indeed, our experiments showed that this unique feature made Bayesian networks suitable for handling missing context values in mobile context-aware recommenders.

Most prior work relied on Bayesian networks that were either manually-crafted via expert elicitation or otherwise translated from an existing model of the domain [7]. However, because causal dependencies among context and also relative importance of context for a particular recommendation task is user-specific, we need to adopt existing techniques for automatically learning the network structure and parameterizing the learned network based on observations. To this end, we identified *CaMML* program [27] as a suitable Bayesian network learning tool.

2.2. Learning Bayesian Networks with *CaMML*

In context-aware recommender systems, context values are observed in an atemporal fashion from many sources. In addition, the models are likely to be more easily understood to be, not linear as in $Y = a_1 X_1 + \ldots a_k X_k + U$, but discrete or multinomial, where variables take on one out of a fixed, finite number of states e.g. the current *weather* can be either *Hot*, *Rainy*, or *Fine*. Like past work that includes [12], we found that the state-of-the-art automated causal discov-

ery learning method for addressing such problems is implemented in the tool *CaMML* [27], largely developed by Wallace, who invented *Minimum Message Length* (MML) [26].

The authors of *CaMML* explained in [14] that, rather than using orthodox statistical significance tests such as the χ^2-test to examine variables for conditional independence, *CaMML* learns causal structure by stochastically searching over the entire space $\{h\}$ of causal models or hypotheses, aiming at finding the model h that maximizes a MML posterior metric. The MML posterior metric used in *CaMML* is defined as $P_{MML}(h) = e^{-I_{MML}(h)}$, for $I_{MML}(h) = logN! - \sum_i logp_i - \sum_j log(1 - p_j)$, where N is the number of variables in h, p_i reflects the prior probability for directed arc i, i indexes the arcs present in h, and j indexes the possible arcs absent from h. Chapter 8 of [14] has more details on the MML metrics for learning discrete causal structures.

In brief, as described in the accompanying manual [23], the *CaMML* employs a *Metropolis algorithm* to sample the space of all possible models subject to specified constraints. For each model that *CaMML* visits on its sampling walk through the real model space, it computes a representative simplification and counts only on these representative models to overlook trivial model variations of no statistical significance. The MML posterior probability of each of these representative models is taken to be the sum of MML posteriors of its members. This total posterior value, that is estimated via the above Metropolis sampling process, approximates the probability that the true model actually lies within the MML equivalence class of that model. The best model is therefore chosen as the one representative model with the highest MML posterior. For details of the sampling process as implemented in *CaMML*, please refer to [27].

2.3. Related Work Utilizing Bayesian Networks

Among prior works that applied Bayesian networks in the area of context-aware computing, a recent paper by Gu et al. [7] explicitly proposed the application of Bayesian network for dealing with uncertain context, by adding dependency and probability markups to the W3C Web Ontology Language (*OWL*) specification and translating their manually defined context ontology into the form of a Bayesian network. This and other prior researches were motivated by the highly-efficient probabilistic reasoning capability of the Bayesian networks in addition to their graphical superiority in representing causal relations among context. Indeed, the use of Bayesian networks in context-aware systems shows great promises in many meaningful applications such as the robotic aid for the elderly blind by Lacey et al. [15].

Similarly, the application of Bayesian networks in recommender system is not new. In [2], a model-based probabilistic approach for collaborative filtering recommendation was proposed that explored the learning of Bayesian network from transaction data, with each node corresponding to an item of interest. More recent works include the papers by Ji et al. [11, 10] that learns Bayesian network as a customer model from customer shopping history data, such that each item or network node represents one particular kind of commodity. Real-time recommendation of items could then be based on probabilistic inference in combination with the last known shopping action of a customer.

Past efforts that suggested the automatic learning of Bayesian network from data had simply taken each potential item of interest as a network node, and then applied the learned model directly for recommendations. They had used the Bayesian network to learn the associations among the items, which could be goods that were sold by a certain store. We note that such an approach (e.g. [16, 11, 10]) suffers from a problem known as *sparsity*, where the number of observations is far too few compared to the size of items to be considered. More importantly, all these previous work did not consider learning and exploiting the important underlying user-specific dependencies among context. As such, they were neither able to identify the minimal context set nor handle missing context values.

3. A Context Model for Effective Learning

The context elements eligible for consideration within the recommender systems are often too numerous for effective learning to be feasible. In addition, within the mobile environment, individual context elements could be acquired in a dynamic process that involves rapidly-changing avenues due to the constant flux in the availability of context sources, an essential characteristic of the popular service-oriented context frameworks described in e.g. [13] and [30]. To resolve this need for a small, practically-manageable set of learning parameters, we adopt the idea of defining *domain-specific markers*, a practice popular within clinical research.

A *marker* is any benchmark that is considered by the system designer to be suitable for evaluating a set of items. For instance, a marker for a restaurant recommender could be "*Is restaurant open during the visit?*", as it is a relevant concern of that system's typical users. In clinical research especially, it is common to define a fixed set of markers for tasks such as evaluation of new drugs. We have built upon this useful concept and focused our learning on a small, defined set of possibly-important markers, instead of learning directly on the potentially overwhelming pool of context.

With reference to our context model in Figure 1, an *item* refers to one of the many choices that an application is designed to rank and recommend, e.g. an available restaurant. In this context model, an upper tier that is concerned with recommendation considerations including *item marker* values and *item score*, is clearly separated from a lower tier that deals with both *user context* and *item attributes*. User context include more abstract context such as a certain user's preference for cleanliness, as well as relatively more direct
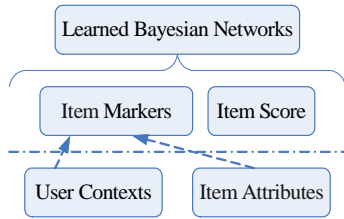
Figure 1. A Tiered Context Model

Figure 2. Context Model of *RR_II*

context like weather. The *item marker* values are derived by comparing the observed user context values with each item's attribute values. The *item score* is then a measure of the suitability of individual item of interest as computed based on these derived marker values.

3.1. Learning Bayesian Network from Observations

We developed a two-tier strategy for learning the proposed context model. Firstly, the minimal set of markers that are important to a user can be identified via Bayesian network learning on the upper tier. Then, we retain just this minimal set of markers and their corresponding subset of relevant user context and item attributes for a final learning. This would yield a Bayesian network that is truly personalized to that user, since it encodes the interdependencies among context that are important specifically to that user.

Our context model provides us with a great flexibility to analyze the dependencies among system variables at multiple levels - not only both at the *lower level* process of marker abstraction and at the *higher level* process of scoring, but also at a *cross-level* process where integrated learning can involve any relevant subset of user context, item attributes, markers, as well as the score variable. In this way, our context model helps to improve scalability as there is no need to learn on the entire large set of potentially important context and attributes. In addition, since we do not model individual items as separate network nodes, we manage to trade a complex, inflexible model of the associations among items for a compact, explanatory model of the user-specific dependencies among context, allowing our learned network to be used to score new items without having to relearn.

3.2. Predictions Using Learned Bayesian Network

For predictions, we feed in the observed values of the necessary user context and item attributes, and allow the beliefs of the marker values and in turn those of the item score to be updated. We can then read off the most probable score value of each item and recommend the highest-scoring ones.

We employed the *Netica-J* API [18] to make predictions on *score* with the networks learned from data. Applying the belief updating mechanism within this API to make predictions is straightforward once CaMML was instructed to store the learned network in the Netica format. Predicting
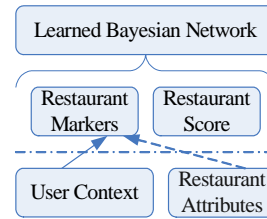
the score value for an item, based on the current observations or finding, involved performing the following steps.

Step 1 Present finding to the network, without score value.

Step 2 Allow network to update beliefs of values for score.

Step 3 Identify the score value with highest updated belief.

For details of the algorithm used within Netica for exact general probabilistic inference in a compiled Bayesian network, known as *message passing in a join tree of cliques*, please refer to the software's documentations [18].

3.3. Online Learning

We can improve the prediction accuracies by making the network learn from the last correct answer as soon as the user's choice was known after each test prediction. In other words, each time after the learned network makes a prediction, which is a best-effort one since the network has no prior knowledge of the correct answer, we feed back into the network that same example now with the correct answer as well. We can then instruct the network to revise its learned conditional probabilities based on this new "training" case.

We observed the prediction accuracy rates of our learned networks both with and without this improvisation applied, and our results confirmed that the accuracies for each test data-set had either stayed equal (if the accuracy was already close to 100%), or had increased noticeably after we introduced case-by-case learning. We note that this case-by-case learning strategy implements our adaptive learning approach where, given each set of context values, we make a best-effort prediction and then learn from the user's choice.

4. A Restaurant Recommender Application

We demonstrate the advantages of our context model via a restaurant recommender named "*RR_II*" (as it is our second restaurant recommender after our first one in [29]). This application is capable of learning automatically from observation data the underlying causal dependencies among parameters as a Bayesian network, and thereafter predicting on a score for each restaurant using the learned network.

Figure 2 shows the context model of *RR_II*. We identified twenty-six user context, thirty restaurant attributes,

4

and twenty-one markers. In a typical mobile recommender, the number of context are much greater than the number of markers, so our advantage of not learning on all the context becomes even more significant in practice. Examples of user context defined were "*weather*", which described an aspect of the user's situation, and "*cleanliness*", an aspect of the user's preferences. Restaurant attributes included "*category*" and "*isClean*". The markers were defined to reflect task-relevant concerns like whether a restaurant is "*opened during visit*". A total of fifteen restaurants were modelled.

Each marker was defined as a boolean variable. For example, a restaurant was either "opened" or "not opened" at the stated time of visit. We developed a software simulator that generated cases spanning across the possible values of context while adhering to a user-specific causal model. In each recommendation cycle, user context values and restaurant attribute values were compared via internal heuristics to determine if a certain marker should take on a value of 1 ("*satisfied*") or 0 ("*not satisfied*") for each of the restaurants.

Examples of the internal heuristics for deriving a partial set of the defined markers are given in Heuristics M3, M11 and M12, where the prefixes "*US.*" and "*UP.*" refer to a *user situation* and a *user preference* context respectively, whilst "*RA.*" references a *restaurant attribute*. Figure 3 shows the defined causal dependencies corresponding to each marker. Based on the derived marker values, we then computed scores for ranking the restaurants. Supervised learning to discover the underlying Bayesian network from the observations was then performed using *CaMML*.

Heuristic M3   *userAttireIsAppropriate* (*M3*)

if *RA.attireRequirement*=="*formal*" **then**
    if US.attire=="*formal*" **then** valueof(*M3*)="*yes*";
    else if US.attire=="*casual*" **then** valueof(*M3*)="*no*";
else if *RA.attireRequirement*=="*none*" **then**
    valueof(*M3*)="*yes*";

Heuristic M11   *matchesIsAirConditionedPref* (*M11*)

if *UP.cleanliness*=="*yes*" **then**
    if *RA.isClean*=="*yes*" **then** valueof(*M11*)="*yes*";
    else if *RA.isClean*=="*no*" **then** valueof(*M11*)="*no*";
else if *UP.cleanliness*=="*dunCare*" **then**
    valueof(*M11*)="*yes*";

Heuristic M12   *isOfDesiredCategory* (*M12*)

if *UP.category*==*RA.category* **then**
    valueof(*M12*)="*yes*";
    else valueof(*M12*)="*no*";

By comparing Figure 2 to our context model in Figure 1, it is clear that our *RR_II* application fully implemented our proposed model. We now describe a series of experiments on *RR_II* to show that the two context acquisition challenges in mobile recommenders, namely identifying the minimal
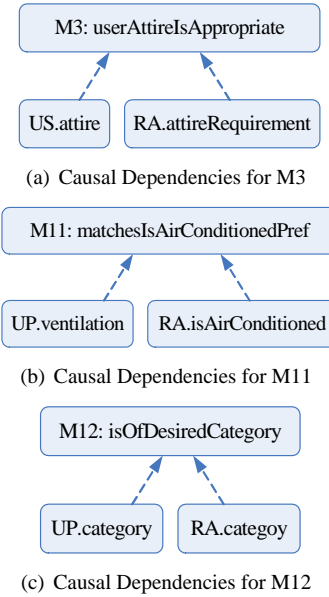

(a) Causal Dependencies for M3

(b) Causal Dependencies for M11

(c) Causal Dependencies for M12

Figure 3. Dependencies for M3, M11 and M12

Table 1. User Preference Rules and Models

User	Preference Rule	Causal Model
1	M3, M11, M12	Model 1
2	M5, M10, M12, M15	Model 2
3	M10, M12, M14	Model 3
4	M5, M10, M12, M21	Model 4
5	M3, M5, M12	Model 5

set of context to minimize acquisition costs, and maintaining a high degree of predictive accuracy in scenarios with missing context values, can indeed both be effectively resolved through the learning of context interdependencies.

5. Experimental Validation

By *causal dependency*, we mean that a variable *X causes* or *affects Y* (denoted as "$X \rightarrow Y$", or "*Y is dependent on X*"), if changes in the observed value of *X* bring about changes to our beliefs for values of *Y*. We now describe two sets of experiments that illustrate the feasibility of our Bayesian network approach towards modelling causal dependencies in a typical context-aware recommender system like *RR_II*.

5.1. Learning Minimal Set of Important Markers

The first set of experiment verified that the minimal set of context truly important to a certain user could be effective recovered in the Bayesian network learned from observation data alone. To generate observation data, we defined a set of *user preference rules* to represent different user logics in considering markers when scoring restaurants, and a set of *user-specific causal models* stating the causal interdependencies among the corresponding important contexts.

The user-specific preference rules and their corresponding causal models are listed in Table 1. For our *User 1*, the three markers of "*userAttireIsAppropriate*", "*matchesIsAir-ConditionedPref*" and "*isOfDesiredCategory*", labelled as "*M3*", "*M11*" and "*M12*" respectively, were equally important. The score for each restaurant was thus computed as

$$score = \frac{No.\ of\ Important\ Markers\ with\ Value\ "1"}{No.\ of\ Important\ Markers}$$

The causal models were defined on user context and not on system markers so as to aptly represent a certain logical user's consistent preferences. Model 1 is illustrated below.

Model 1   *Causal Dependencies Among Context for User 1*

---
  if *UP.category == "cafe"* **then**
    US.attire = "casual";
    UP.ventilation = "aircon";
  else if *UP.category == "club"* **then**
    US.attire = "formal";
    UP.ventilation = "aircon";
  else if *UP.category == "canteen"* **then**
    US.attire = "casual";
    UP.ventilation = "nonAircon";
  else if *UP.category == "dunCare"* **then**
    US.attire = "dunCare";
    UP.ventilation = "dunCare";

For each rule, we learned using *CaMML* from a set of a thousand observations, and retained only markers that were directly linked to the *score* node in the resulting network. We then learned again on the same set of examples but with all other markers removed. This process was repeated until a learned model had all the markers directly linked to score.

From our first round of learning using CaMML on the one-thousand-sample training set corresponding to User 1, we obtained the Bayesian network shown in Figure 4(a). On the right-hand side of this figure, we see that only the four markers of M3, M11, M12 and M18 were connected to score. To verify if all these four markers were really important, we retained just these four markers and score and performed a second round of learning. From the resulting network in Figure 4(b), only M3, M11 and M12 (the three on the left) were directly connected to the score node. We therefore retained only score and these three nodes for our third round of confirmation learning. The resulting network of Figure 4(c) shows that *all* the three important markers specific to the first user were directly connected to score.

For performance metrics, we adopted the *F-measure* under the following definitions.
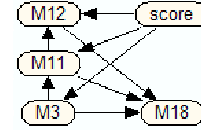
$$Recall = \frac{No.\ of\ Important\ Markers\ Connected\ to\ Score}{Number\ of\ Important\ Markers}$$

$$Precision = \frac{No.\ of\ Important\ Markers\ Connected\ to\ Score}{No.\ of\ Markers\ Connected\ to\ Score}$$
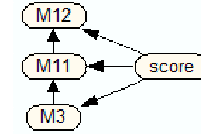
$$F\text{-}measure = \frac{2(Recall)(Precision)}{(Recall)+(Precision)}$$
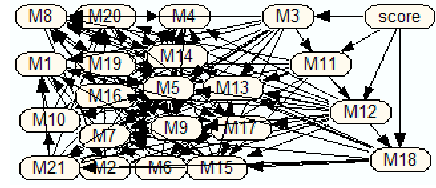
(a) Bayesian Network Discovered in Round 1

(b) Bayesian Network Discovered in Round 2

(c) Bayesian Network Discovered in Round 3

Figure 4. Discovered Bayesian Networks

Table 2. Minimal Set Learning Performance

User	Round	Recall	Precision	F-measure
1	1	1.00	0.75	0.86
	2	1.00	1.00	1.00
2	1	1.00	0.67	0.80
	2	1.00	1.00	1.00
3	1	1.00	0.50	0.67
	2	1.00	1.00	1.00
4	1	1.00	1.00	1.00
5	1	1.00	0.60	0.75
	2	1.00	1.00	1.00

The performance of learning for the modelled user rules are given in Table 2. We observed that the learned Bayesian networks consistently yielded a F-measure of 100% by the second round for users 1, 2, 3 and 5, while a 100% measure was obtained after just one round of learning for our user 4. However, logically, we expect the causal arcs to point *from* each of the important markers *to* score, but in all the learned networks, the arcs pointed *outward* from score instead. This was because the learned network and the logically-expected network were *statistically equivalent* [4], i.e. they had the same *skeleton* (undirected graph) and *v-structures* (nodes that are the children of two non-adjacent parents) [25].

Dai et al. [4] pointed out that according to Chickering in [3], statistically equivalent models cannot be distinguished on the basis of observational data alone. Therefore, we can say that the discovery of a statistically equivalent model is as good as the discovery of the actual explanatory causal
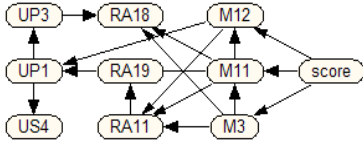
Figure 5. Full Bayesian Network for User 1

model in our experiments, since we are learning Bayesian networks from just the observation data. Our results show that our approach is indeed effective in identifying the minimal set of context that were important to a particular user.

5.2. Predictions with Missing Context Values

In our second set of experiments, we employed the minimal sets of important markers identified in our earlier experiments to verify that the prediction accuracies on *score* remained high under scenarios with missing context values. For each user, we prepared the same earlier set of a thousand observations to perform a 5×2-fold cross-validation. In each round of the validation, five-hundred observations were used for learning of the causal dependencies, while the remaining observations were used for testing of prediction accuracy on score. Within each fold, we learned a Bayesian network on the training set consisting of just the score variable and the minimal set of markers together with their corresponding user context and restaurant attributes. Predictions on the test set were then performed using the Netica-J API. In all our predictions using the learned Bayesian networks, we did not use case-by-case revision of beliefs when predicting on the test samples so as to reflect the true accuracies after learning on just the five-hundred training samples.

In each of the validation fold, prediction accuracies were recorded for the baseline scenario where all the context data were available, as well as under imperfect scenarios where a particular context value (and hence its dependent marker value) were missing in all of the five-hundred test samples. We compared the results to the corresponding prediction accuracies achieved using the *J4.8 decision tree* classifier as implemented in the *Weka* knowledge analysis tool [28], using the default options within the *Explorer* GUI of *Weka*.

We first investigated if the learned networks comprising just the minimal set of markers and the score (i.e. the *upper tier* in our context model) were already sufficient to handle missing context values. Through a series of 5×2-fold cross-validations, we observed the predictions on score when all context were available, and when the important contexts (and hence their corresponding dependent markers) went missing one at a time. The results are shown in Table 3(a). Both Bayesian network and decision tree suffered more than 20% drop in accuracy with missing context values, suggesting that the important context and their user-specific dependencies had to be captured directly in our learning process.

Table 3. Prediction Accuracies *(%)*: average accuracy of 5×2-fold cross-validation; *Complete*: when data is complete; *Missing*: with one context missing

(a) Learning and Predicting Using Only the Upper Tier

User	Bayesian Network (%)		Decision Tree (%)	
	Complete	Missing	Complete	Missing
1	100	76.8	100	76.8
2	100	80.5	100	77.5
3	100	70.5	100	66.3
4	100	86.2	100	68.6
5	100	79.8	100	75.9
Avg.	100	78.8	100	73.0

(b) Learning and Predicting Using a Cross-Tier Approach

User	Bayesian Network (%)		Decision Tree (%)	
	Complete	Missing	Complete	Missing
1	100	100	100	78.4
2	100	100	100	81.7
3	100	100	100	64.5
4	100	100	100	77.8
5	100	100	100	77.5
Avg.	100	100	100	76.0

Table 3(b) summarizes the accuracies when the important context and restaurant attribute values from the same examples were incorporated for learning. A sample network learned for User 1 is shown in Figure 5. Clearly, Bayesian networks and decision trees performed equally well when data was complete, i.e. when no context value was missing, but we observe that Bayesian network significantly outperformed decision tree to maintain a strong 100% prediction performance under the scenarios of missing context values.

6. Conclusion and Future Work

Mobile context-aware recommenders face operational constraints in context sensing and acquisition. As such, they present unique key challenges, among which is the need to minimize the context to be acquired, and to handle missing context appropriately so as to maintain accurate predictions.

We proposed applying Bayesian network learning to discover the underlying context dependencies that are specific to a particular user and recommendation task. To enable a fast and scalable learning of the network from data, we presented a context model involving the abstraction of user context and item attributes into a more manageable set of markers. By first learning on these small set of markers to identify the user-specific considerations, we can then pinpoint the optimal set of context and attributes from which we can learn a predictive model that effectively captures the interdependencies among these important parameters.

Through a series of experiments conducted on a restaurant recommender based upon our proposed context model, we validated that our system can indeed accurately recover the dependencies among context to yield a clear graphical model of the problem, so that we were able to readily identify the minimal set of context important to a particular user and task. In addition, we showed that because Bayesian networks intrinsically encode the correlations among context, our approach maintained high predictive accuracies even in scenarios when the necessary context values were missing.

Although Bayesian networks demonstrated superior performance in handling missing context values, the interdependencies is not so interpretable compared with the logic learned by a decision tree. Due to its interpretability, *decision trees* can provide explanation for *interactions* between variables. For instance, suppose an induced tree splits on "*weather*", which can either be "*good*" or "*bad.*". We may see that for "*bad weather*", "*location*" yields the highest entropy reduction, but not so for "*good weather*". Although decision trees are useful generally for explaining up to just three-way interactions, such an ability is useful for answering important questions like "*Is location important to that user when weather is good/bad?*". It is therefore our intention to explore in future, among other issues, the automatic extraction of similar explanations from a Bayesian network.

References

[1] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. *ACM Trans. On Info. Systems (TOIS)*, 23(1):103–145, 2005.

[2] J. S. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Procs of UAI 1998*, pages 43–52, Madison, WI, 1998.

[3] D. M. Chickering. A Transformational Characterization of Equivalent Bayesian Network Structures. In *Procs of UAI 1995*, pages 87–98. Morgan Kaufman, 1995.

[4] H. Dai, K. Korb, C. Wallace, and X. Wu. A Study of Causal Discovery With Weak Links and Small Samples. In *Procs of IJCAI 1997*, pages 1304–1309, San Francisco, August 1997.

[5] A. K. Dey and G. D. Abowd. Towards a Better Understanding of Context and Context-Awareness. Technical Report GIT-GVU-99-22, June 1999. Panel paper at HUC 1999.

[6] J. H. Friedman. On Bias, Variance, 0/1-Loss, and the Curse of Dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, March 1997.

[7] T. Gu, H. K. Pung, and D. Q. Zhang. A Bayesian Approach for Dealing with Uncertain Contexts. In *Adjunct Procs of Pervasive 2004*, Vienna, Austria, April 2004. OCG.

[8] D. Heckerman. A Tutorial on Learning with Bayesian Networks. In M. Jordan, editor, *Learning in Graphical Models*, Cambridge MA, 1999. MIT Press.

[9] J. L. Herlocker and J. A. Konstan. Content-Independent Task-Focused Recommendation. *IEEE Internet Computing*, 5(6):40–47, 2001.

[10] J. Ji, C. Liu, J. Yan, and N. Zhong. Bayesian Networks Structure Learning and Its Application to Personalized Recommendation in a B2C Portal. In *Procs of WI 2004*, pages 179–184. IEEE Computer Society, September 2004.

[11] J. Ji, Z. Sha, C. Liu, and N. Zhong. Online Recommendation Based on Customer Shopping Model in E-Commerce. In *Procs of WI 2003*, pages 68–74, Canada, October 2003.

[12] K. Karimi and H. J. Hamilton. Discovering Temporal/Causal Rules: A Comparison of Methods. In *Procs of AI 2003*, pages 175–189, 2003.

[13] J. W. Koolwaaij and P. Strating. Service Frameworks for Mobile Context-Aware Applications. In *Procs of eChallenges 2003 Workshop*, Bologna, Italy, October 2003.

[14] K. B. Korb and A. E. Nicholson. *Bayesian Artificial Intelligence*. CRC Press, 2003.

[15] G. Lacey and S. MacNamara. Context-Aware Shared Control of a Robot Mobility Aid for the Elderly Blind. *Robotics Research*, 19(11):1054–1065, November 2000.

[16] O. Madani and D. DeCoste. Contextual Recommender Problems. In *Procs of UBDM 2005*, pages 86–89, August 2005.

[17] P. Massa and B. Bhattacharjee. Using Trust in Recommender Systems: An Experimental Analysis. In *Procs of iTrust 2004*, pages 221–235, 2004.

[18] Norsys Software Corporation. Netica-J: Java Netica API. Retrieved June 2005, http://www.norsys.com/netica-j.html.

[19] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1989.

[20] P. Resnick and H. R. Varian. Recommender Systems. *Communications of the ACM*, 40(3):56–58, 1997.

[21] M. Singh and G. Provan. Efficient Learning of Selective Bayesian Network Classifiers. Technical Report MS-CIS-95-36, Computer and Information Science Department, University of Pennsylvania, Philadelphia, PA., November 1995.

[22] H. W. Tung and V. W. Soo. A Personalized Restaurant Recommender Agent for Mobile E-Service. In *Procs of EEE 2004*, pages 259–262, 2004.

[23] C. R. Twardy and R. O'Donnell. *CaMML Manual*. Monash Data Mining Center (MDMC), July 2004.

[24] M. van Setten, S. Pokraev, and J. Koolwaaij. Context-Aware Recommendations in the Mobile Tourist Application COMPASS. In *Procs of AH 2004*, pages 235–244, 2004.

[25] T. Verma and J. Pearl. Equivalence and Synthesis of Causal Models. In *Procs of UAI 1990*, pages 220–227, Boston, MA, 1990. Morgan Kaufmann.

[26] C. S. Wallace and D. M. Boulton. An Information Measure for Classification. *Computer Journal*, 11(2):185–194, 1968.

[27] C. S. Wallace and K. B. Korb. Learning linear causal models by MML sampling. In A. Gammerman, editor, *Causal Models and Intelligent Data Management*. Springer, 1999.

[28] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[29] G.-E. Yap, A.-H. Tan, and H.-H. Pang. Dynamically-Optimized Context in Recommender Systems. In *Procs of MDM 2005*, pages 265–272, May 2005.

[30] D. Zhang, X. H. Wang, and K. Hackbarth. OSGi Based Service Infrastructure for Context Aware Automotive Telematics. In *Procs of the IEEE Vehicular Technology Conf. (VTC Spring 2004)*, pages 2957–2961, Milan, Italy, May 2004.