2009 IEEE International Conference on Data Mining Workshops

Fast Visual Trajectory Analysis using Spatial Bayesian Networks

Thomas Liebig, Christine Körner, Michael May
Fraunhofer IAIS
Sankt Augustin, Germany
{*thomas.liebig, christine.koerner, michael.may*}*@iais.fraunhofer.de*

***Abstract*—During the past years the first tools for visual analysis of trajectory data appeared. Considering the growing sizes of trajectory collections, one important task is to ensure user interactivity during data analysis. In this paper we present a fast, model-based visualization approach for the analysis of location dependencies in large trajectory collections.**

Existing approaches are not suitable for visual dependency analysis as the size and complexity of trajectory data constrain ad hoc and advance computations. Also recent developments in the area of trajectory data warehouses cannot be applied because the spatial correlations are lost during trajectory aggregation. Our approach builds a compact model which represents the dependency structures of the data. The visualisation toolkit then interacts only with the model and is thus independent of the size of the underlying trajectory database. More precisely, we build a Bayesian Network model using the Scalable Sparse Bayesian Network Learning (SSBNL) algorithm [1], which we improve to represent also negative correlations. We implement our approach into the GIS MapInfo using MapBasic scripts for the user interface and an independent mediator script to retrieve patterns from the model. We demonstrate our approach using mobile phone data of the city of Milan, Italy.

***Keywords*-Spatial Bayesian Networks; SSBNL; trajectories; visualisation**

I. Introduction

Visualization is a natural approach to analyse spatial data. Due to the truthful representation of geographic shapes and relationships it allows, for example, an easy detection of correlation patterns. Also trajectory data, as one characteristic type of spatio-temporal data, has received notable attention from the area of visual analytics recently [2]. In this paper we consider the visualization of dependencies within trajectory data.

Clearly, the easy availability of GPS and other tracking technologies encourage a growing collection of trajectory data. This data bears numerous information that can be used in traffic management or location based services. However, the growing amount of data also poses challenges with respect to generalization and performance criteria. On one side, generalization techniques such as aggregation, smoothing or filtering are necessary to distil relevant information and to cancel out background noise. On the other side, large data collections easily outgrow the size of main memory and need sophisticated caching, sampling or compression techniques for ad hoc analysis. This is especially important for online analytical processing (OLAP) and visual analytics as both methods rely on user interaction.

In this paper we present a fast, model based visualization approach for the analysis of location dependencies. Location dependencies describe the co-occurrence of geographic locations within a trajectory. They occur naturally as personal movement is purpose-driven and not a random walk through a city. Location dependencies can be expressed as conditional probability to visit an arbitrary location given that another (set of) location(s) is visited within a trajectory as well. More formally, given a finite universal set $\mathcal{L}$ of discrete geographic locations, a set $L^+ \subseteq \mathcal{L}$ containing locations that are visited with certainty and a set $L^- \subseteq \mathcal{L} \setminus L^+$ containing locations that are not visited with certainty within a trajectory, we can specify the location dependency of an arbitrary location $l \in \mathcal{L}$ by the probability $P(l \mid L^+, \neg L^-)$. The sets L^+ and L^- are also called positive and negative evidence, respectively.

The degree of dependency between two or more locations can be calculated by simple counting statistics. However, as each statistic requires a complete database scan, this method is inefficient for ad hoc analysis of large datasets. What other options exist to speed up visual analysis? First, we could calculate all dependencies in advance and store the results. Second, we could try to reduce the size of the trajectory set. The first option is not practicable as exponentially many combinations of locations exist. The second option can be achieved by aggregation using trajectory data warehouses (TDW) [3], [4]. However, TDW are not able to reconstruct location dependencies because the identity of trajectories is lost during compression.

We therefore combine visualisation of dependencies with an approach introduced in previous work [1], which provides an algorithm for the compact representation of location dependencies using Bayesian Networks. We developed an interface to guide the selection of evidence in the geographic information system (GIS) MapInfo [5] and to control the interaction with the Bayesian Network model. We demonstrate the analysis process and visual interaction with Bayesian Networks using mobile phone data of the city of Milan.

In Section II we review related work on visual analysis techniques for large trajectory collections. Section III introduces Spatial Bayesian Networks and Section IV summarises the SSBNL algorithm of our previous work [1].

978-0-7695-3902-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ICDMW.2009.44

In this section we also extend the algorithm to represent negative correlations. Section V shows the integration of the Bayesian model with MapInfo, and Section VI illustrates our work using real-world mobile phone data.

II. Related Work

A number of different research areas have contributed to the analysis of geographic data. Next to geostatistics and geographic information systems and science, database technology and data mining play a major role in the development of analysis methods for large spatial and spatio-temporal data sets. While spatial database technology and spatial data mining have become well-established parts in their respective research areas, methods and tools for the analysis of trajectory data are still in their infancy [6].

Recently, two approaches for aggregation and analysis of large sets of trajectory data have been published. Both approaches rely on a database-side aggregation of the data prior to data analysis and use the tool CommonGIS for visualization. CommonGIS [2] is a software system for interactive visual analysis of spatially and temporally referenced data. The first approach by Andrienko and Andrienko [7] relies on the user to perform aggregation using standard database functions. CommonGIS provides a direct database access, and the user can load tables with previously aggregated data (see Figure 1, Appr. 1). Only if the data set is small enough to fit into main memory, so-called dynamic aggregators can be applied directly within CommonGIS. The second approach by Leonardi et al. [8] performs aggregation using a TDW. TDW [3], [4] have recently been developed and are a first step into OLAP analysis of trajectory data. The TDW stores aggregated data at a given level of resolution. CommonGIS interacts with the TDW to allow for visually aided OLAP, e.g. roll-up and drill-down operations for graphically selected areas (see Figure 1, Appr. 2).

In contrast to the above approaches, our visual analysis process is not based on aggregated data but on a model of the data (see Figure 1, Appr. 3). The model is a compact representation of trajectory dependency structures and is extracted in a first data mining step. The visualisation toolkit interacts only with the model and is thus independent of the size of the underlying trajectory database.

The existing approaches are not suitable for visual dependency analysis. First, the exponential number of location subsets prohibits advance computation and ad hoc calculation of dependencies may take too long for large data sets. Second, TDW naturally do not keep the identity of trajectories during aggregation, which makes inference of location dependencies impossible. However, our approach is tailored to one specific analysis task as it extracts patterns early within the analysis process. In contrast, approaches 1 and 2 are flexible with respect to possible analysis questions, because the selection and control of analyses resides with the user in the upper most level.

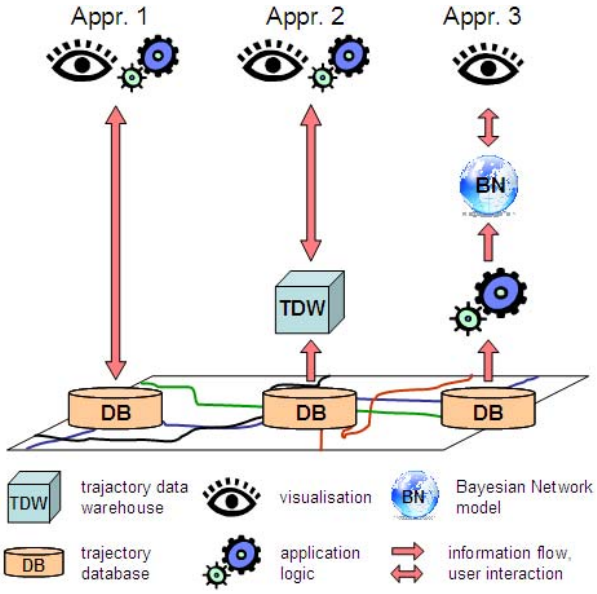

Figure 1. Aggregation and analysis approaches for trajectory data

III. Spatial Bayesian Networks

Given a set of trajectory data (for example in form of GPS logs), conditional dependencies between two or more locations can be determined by counting co-occurences within the data. The calculation of pairwise dependencies results in a square matrix, and the addition of further locations extends the dimension of the matrix respectively. This representation becomes soon unmanageable and is inappropriate in practice. Note, that pairwise dependencies do not suffice to represent trajectory data, as the choice of a person at a crossroad depends on his or her origin.

Bayesian Networks are intended to store multivariate probability distributions of a set of random variables $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$. They are graphical models representing each random variable by a vertex and dependency relations among them by arrows, which results in a directed acyclic graph $G = (\boldsymbol{X}, E \subseteq \boldsymbol{X} \times \boldsymbol{X})$. Additionally, each vertex X_i stores a probability table describing its own state depending on its parents. Thus, the joint probability distribution $p(\boldsymbol{X} = \boldsymbol{x})$ among the random variables $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ is given by

$$p(\boldsymbol{X} = \boldsymbol{x}) = p(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \prod_{i=1}^{n} p(X_i = x_i | parents(X_i))$$

where $parents(X_i)$ is the set of all ancestors X_j having a directed edge in G connecting X_j with X_i, i.e.

$$parents(X_i) := \{X_j | (X_j, X_i) \in E\}.$$

The random variables $\boldsymbol{X}$ correspond to the discrete location set $\mathcal{L}$. For each positive evidence $l_i \in L^+$ the corre-

sponding random variable x_i is $TRUE$ and for each negative evidence $l_j \in L^-$ variable x_j is $FALSE$. Locations are generally obtained from trajectories using a discretisation operator. For example, GPS data can be matched to the street network, and each street segment represents a location. In our application, the target data are reconstructed GSM trajectories, i.e. the discretisation is already provided by the data collection technique. Each GSM-cell represents a location which, in contrast to an arbitrary grid, can vary in shape and size, overlap and also depend on the weather [9].

The application of Bayesian Networks for spatial data mining and knowledge discovery was introduced by Huang and Yuan [10]. They give a brief overview on the promising possibilities of such a probabilistic model and its construction. Nevertheless, Bayesian Networks are seldom applied in spatial data mining because the structure learning of Bayesian Networks is proven to be NP-complete [11] and spatial data sets are usually of high complexity. Therefore, approximation algorithms which reduce the Bayesian Network search space according to a heuristic are necessary. In previous work [1], we developed an algorithm which is able to handle the demands of spatial data sets and can be applied to large trajectory collections. We will briefly review this algorithm in the next section.

IV. Scalable Sparse Bayesian Network Learning

The Scalable Sparse Bayesian Network Learning (SSBNL) algorithm [1] combines the advantages of the Sparse Candidate [12] and the Screen Based Network Search [13]. It bounds the number of possible ancestors similar to [12] by pre-sampling a given sparseness in the database, and bounds the edgeset to most significant dependencies by only processing frequent itemsets similar to [13]. This is done in a two-step algorithm: First, we pre-sample within each route a set of maximal k distinct locations uniformly distributed among the trajectory. Afterwards, we enumerate frequent variable sets on this pre-sampled data with threshold t and maximal length ml. The result is a bounded number of location-subsets adjustable in their size. For each of these sets a local Bayesian Network is determined in a second step that fits the original data best and the involved edges become collected on a stack. Next, this stack is sorted according to the score of the local networks. In a third step, edges are drawn from the ordered stack to construct a global Bayesian Network. Constraints for this selection are that every chosen edge must not create any cycle in the network but increase the score of the final network. Afterwards, a final database scan of the original trajectory dataset is required to recompute the common probability tables for each vertex in the global Bayesian Network.

The whole Scalable Sparse Bayesian Network Learning (SSBNL) algorithm uses pre-sampling to transform an arbitrary dataset to a processable one with adjustable size and density. Although being an approximation algorithm, the guaranteed output is one of its main advantages. It gives a reasonable approximation for positive correlations [1], because the most significant dependencies persist the pre-selection of variables.

However, in order to answer queries correctly in our visual trajectory analysis, the model needs also the ability to represent negative correlations. Otherwise we are unable to express exclusive or (XOR) relations among locations in a trajectory, e.g. “If a car passes location A it is unlikely to pass location B within the same trajectory”. Including edges to a Bayesian Network is always possible, if it does not create directed cycles in the network structure. Thus we sample multiple pairs of variables. In case both variables of a pair correlate negative and an edge would be valid and increases the network score, we insert an edge into the network (see lines 18 to 27 in Algorithm 1). This pairwise approach is reasonable as shown in [14]. The complete network learning Algorithm is summarized in algorithm 1.

V. Integration in MapInfo

After the dependencies among the locations are extracted utilizing Spatial Bayesian Networks, a query tool is required to inspect the correlations of different locations within the underlying trajectory set. Our approach is to embed this user interface directly into the geographic information system (GIS) MapInfo [5]. It is then possible to set positive and negative location evidence in a user-friendly way by simply selecting the discrete spatial object (points, shapes or lines) from a map.

Our extension for MapInfo consists of two parts: a user interface and a mediator script. The user interface offers tool buttons, dialogs, error messages and user guidance. It also visualises the result of a query as thematic layer within the current map window. This part of our query tool is implemented in MapBasic [15], a scripting language shipped with MapInfo. Each user query needs to execute Bayesian inference on the Spatial Bayesian Network according to the given evidence. In order to keep this part independent of the currently used GIS, we create a separate mediator script written in the language R [16]. Thus, we may easily use other geographic information systems or access the learned Spatial Bayesian Network from different applications written in R script as well. An advantage of the scripting language R is its large collection of statistical analysis packages and references. In our case we use the Bayesian Network data structure defined in the `deal` package [17].

The data exchange between the user interface and mediator script is implemented using files. The control flow and synchronization of the execution is solved calling a shell execution command at each computation request sent by MapInfo. This means, any computation cycle starts a single R process that reads the evidence from a file and stores the inference results in a different file. The MapBasic program

Algorithm 1 SCALABLE SPARSE BAYESIAN NETWORK LEARNING

Require: D , complete dataset
k , maximal frequent set size
ml , frequent set length
t , support threshold
n , number of random edges
$g(\cdot)$, Bayesian Network score
Ensure: BN , a Bayesian Network

for all observations $\omega \in D$ **do**
  ω' := sample k locations from ω
  add ω' to D'
end for
FS := enumerate frequent sets (D',t,ml)
for all $fs \in FS$ **do**
  $BN^* = \arg\max_{BNonfs}\, g(BN, D)$
  add edges of BN^* to $edgedump$ or if already in $edgedump$ increase their score
end for
sort $edgedump$ decreasing
for all $edge \in edgedump$ **do**
  if $BN \cup edge$ contains no cycle **then**
    if $g(BN \cup edge) > g(BN)$ **then**
      add $edge$ to BN
    end if
  end if
end for
for $i = 1$ to n **do**
  sample 2 different locations X_1, X_2
  if X_1, X_2 correlate negative **then**
    if $BN \cup edge(X_1, X_2)$ contains no cycle **then**
      if $g(BN \cup edge) > g(BN)$ **then**
        add $edge$ to BN
      end if
    end if
  end if
end for
return BN

waits for the creation of this file and continues execution afterwards. In order to prevent long import times of the Spatial Bayesian Network every time a new R process is created, we store the complete R workspace with all objects (including the Bayesian Network) as the default workspace. The workspace is read very fast at startup and written after execution automatically.

Combining all parts, our fast query tool based on Bayesian Networks consists of a layered structure as depicted in picture 2. The architecture offers several possibilities for the independent exchange of components, which is important for future development and reusability. At the bottom in figure 2 is the Bayesian Network. The spatial dependency model may be accessed by other tools and the Spatial Bayesian Network may also be replaced by a more accurate one or even a complete different dependency model.

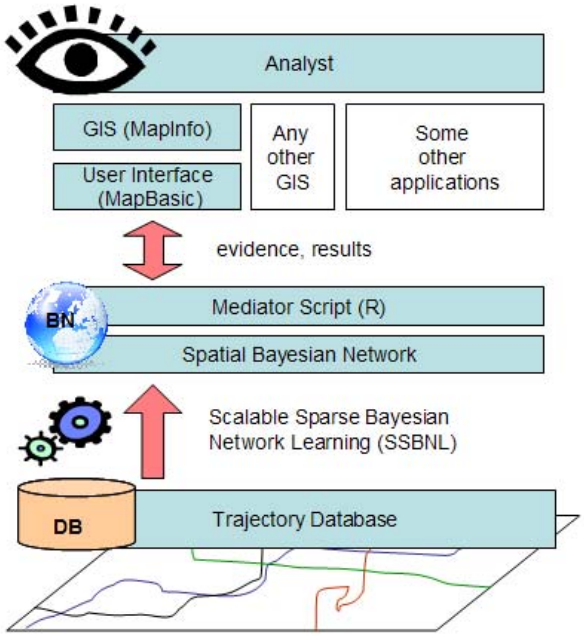

Figure 2. Layered integration of Spatial Bayesian Network model into GIS

VI. APPLICATION

Our data set consists of trajectories collected through mobile communication technology. Every day mobile phone service providers store great amounts of technical communication data that are used for billing purposes and resource allocation. This data also contains valuable information about the movements of mobile phone users, and their usage is under discussion in many places as they may be provided in large numbers at low cost. Typically, the data consists of tag data and handover data. The tag data is used for billing and states for every call an user id, the radio cells in which the call started and ended as well as the start and end time. A handover is a transfer of an ongoing call from one cell to another. It occurs when people move between areas that are covered by two different radio cells or when a reassignment of users due to lack of capacity is required. In our case the handover data are one-hour aggregates that contain the number of people that pass from one cell to an adjacent cell while making or receiving a call. Handover data is used by network providers to optimise resource allocation. Clearly, tag and handover data contain only partial information about customer movement and do not have the format of trajectories themselves. However, trajectories may be reconstructed using current information about transition behaviour. The aggregated handover information can be used to derive the most likely route between start and end cell of

a call. In addition, the call duration can be used to verify the plausibility of a reconstructed route through the radio network.

Our data set consists of reconstructed trajectories for the city of Milan, that are spaced over a radio network with 69 cells. The data have been collected over one week in autumn 2008. In total, the data set consists of 98,994 records of 17,948 mobile phone users.

We applied the improved SSBNL algorithm (see Algorithm 1) to the data set using the following parameterization. As the data set is comparably small in its number of variables for this algorithm, a first pre-sampling step within the trajectories was not necessary. We computed frequent location sets with maximal parity of 4 and a frequency threshold of 5. The Bayesian Network scoring metric we applied was BDeu [18]. In the end we drew 1000 edge candidates and add negative correlations to the network. The whole Bayesian Network learning took about 10 minutes on a standard desktop computer (CPU 3GHz, RAM 3GB).

Figure 3 depicts the results of the Milan Bayesian Network for five different queries. Red colors indicate a low visit probability, green colors indicate a high probability. The blue cells are not part of model because the input data does not contain trajectories for these cells. The top middle picture shows the unconditioned probability to be in some cell during a phone call. The probabilities in this case are in general low. The top right figure shows the probability distribution after setting a positive evidence for one cell. The cells included in a query are marked by dashes. If the cells are green, a positive evidence has been set (i.e. certain passage). If the cells are red, a negative evidence (i.e. no passage with certainty) has been set.

The passage of the marked cell increases the passage probabilities in the surrounding area, which clearly shows the positive correlation between neighbouring cells. If further positive evidence is added, the probabilities increase in the direction of the new cells but show also a tendency towards the city center. In the bottom right picture, two cells with negative evidence are added in the northwest of Milan. They prohibit the spread of mobility in the northern part of Milan. Note that one cell in the very west of Milan shows a high positive correlation although all neighbouring cells contain low probabilities. This behavior is not unreasonable, as connecting paths may lie outside of the network area.

Beware that the model represents not a general mobility model of Milan. As phone calls are usually of short duration, predominantly local dependencies are expressed in the Bayesian Network.

In future work we plan to add further variables that contain semantic information to the trajectories and thus to the learning process. Examples are type of contract, used services (e.g. SMS, phone call) day of week or time of day of a call. It is then possible to condition the network not only on location information but also on qualitative variables. The Bayesian network could then help the mobile phone service provider to perform customer segmentation or to improve resource allocation.

VII. Conclusion

In this paper we present a model-based approach for fast visualization of dependency patterns in large collections of trajectory data. Our approach applies the SSBNL algorithm of [1] to generate a compact model of trajectory dependency structures. This algorithm can be applied to arbitrarily large collections of trajectory data and produces a model that is independent of the size of the data set. Using only the model for visualization has the advantage that neither large amounts of aggregated data need to be stored nor that visualization is restrained by long execution times of ad hoc queries.

We implemented our approach into the GIS MapInfo using MapBasic scripts for the user interface and an independent mediator script to retrieve patterns from the model. We applied our approach to mobile phone data of the city of Milan.

Acknowledgement

This research has partially been supported by the EU FP6-14915 project GeoPKDD. We thank our project partner *elis*[1] for data preparation and provision of the application data.

References

[1] T. Liebig, C. Körner, and M. May, "Scalable sparse bayesian network learning for spatial applications," in *ICDM Workshops*. IEEE Computer Society, 2008, pp. 420–425.

[2] G. Andrienko, N. Andrienko, and S. Wrobel, "Visual analytics tools for analysis of movement data," *ACM SIGKDD Explorations*, vol. 9, no. 2, pp. 38–46, 2007.

[3] S. Orlando, R. Orsini, A. Raffaetà, A. Roncato, and C. Silvestri, "Trajectory data warehouses: Design and implementation issues," *Journal of Computing Science and Engineering*, vol. 1, no. 2, pp. 240–261, 2007.

[4] G. Marketos, E. Frentzos, I. Ntoutsi, N. Pelekis, A. Raffaetà, and Y. Theodoridis, "Building real world trajectory warehouses," in *Proc. 7th International ACM SIGMOD Workshop on Data Engineering for Wireless and Mobile Access (MobiDE'08)*, 2008.

[5] L. Daniel, P. Loree, and A. Whitener, *Inside MapInfo professional*, 3rd ed. OnWord Press, 2002.

[6] M. Nanni, B. Kuijpers, C. Körner, M. May, and D. Pedreschi, "Spatiotemporal data mining," in *Mobility, Data Mining and Privacy*, F. Giannotti and D. Pedreschi, Eds. Berlin Heidelberg: Springer, 2008, ch. 10.

[7] G. Andrienko and N. Andrienko, "Spatio-temporal aggregation for visual analysis of movements," in *Symposium on Visual Analytics Science and Technology (VAST '08)*. IEEE Computer Society, 2008, pp. 51–58.

[1] http://www.elis.org/

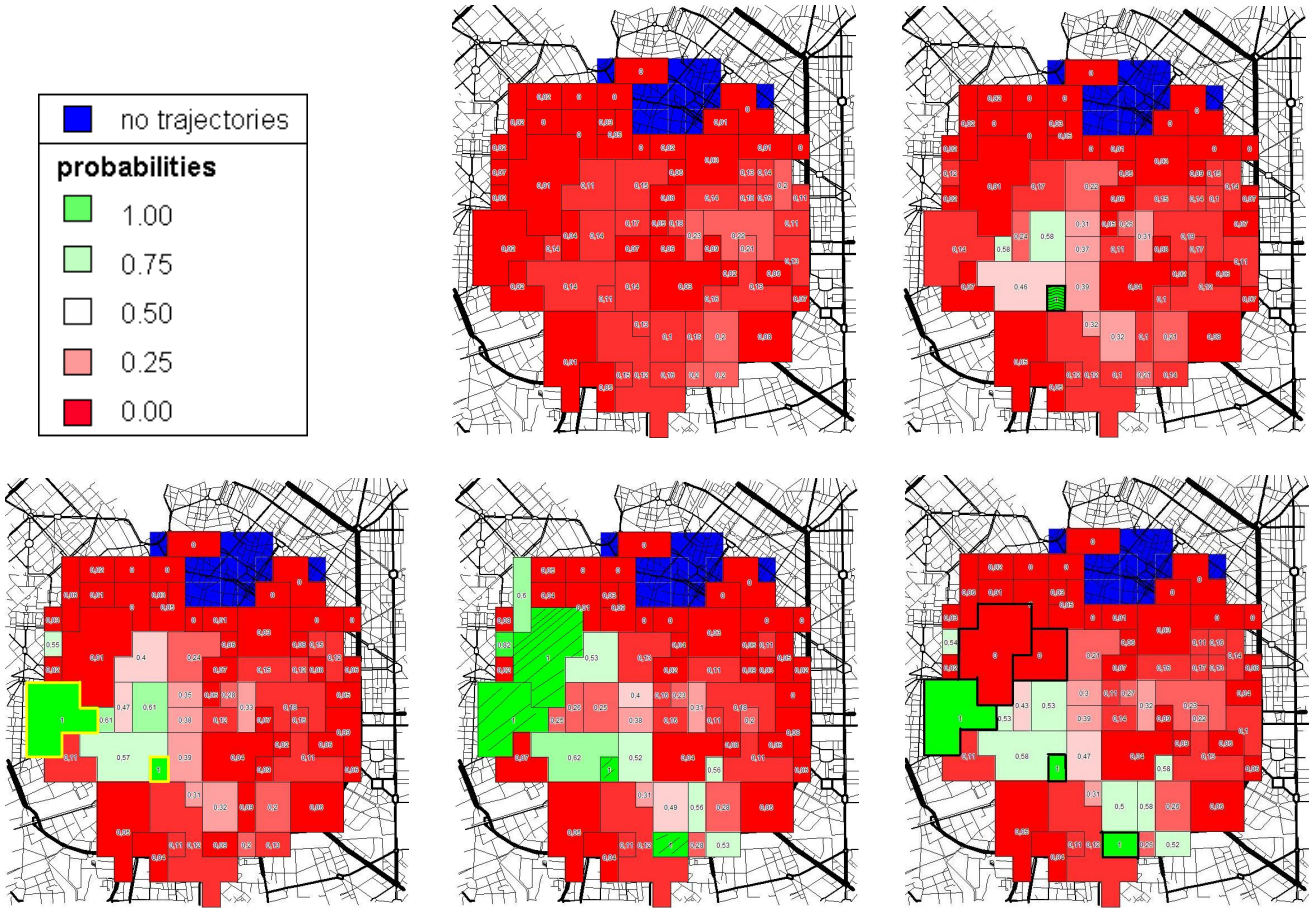

Figure 3. Visualization of Bayesian Network model based on GSM data for the city of Milan

[8] L. Leonardi, G. Marketos, E. Frentzos, N. Giatrakos, S. Orlando, N. Pelekis, A. Raffaetà, A. Roncato, C. Silvestri, and Y. Theodoridis, "T-Warehouse: Visual OLAP Analysis on Trajectory Data," Università Ca' Foscari di Venezia, Tech. Rep. CS-2009-7, July 2009.

[9] K. Laasonen, "Mining Cell Transition Data," Department of Computer Sciences, Tech. Rep. A-2009-3, March 2009.

[10] J. Huang and Y. Yuan, "Construction and application of bayesian network model for spatial data mining," in *Control and Automation, 2007. ICCA 2007. IEEE International Conference on*. Institute of Electrical and Electronics Engineers (IEEE), 2007, pp. 2802–2805.

[11] D. M. Chickering, "Learning Bayesian networks is NP-Complete," in *Learning from Data: Artificial Intelligence and Statistics V*, D. Fisher and H. J. Lenz, Eds. Springer-Verlag, 1996, pp. 121–130.

[12] N. Friedman, I. Nachman, and D. Peér, "Learning Bayesian Network Structure from Massive Datasets: The "Sparse Candidate" Algorithm." in *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI'99)*. Morgan Kaufmann, 1999, pp. 206–215.

[13] A. Goldenberg and A. W. Moore, "Tractable Learning of Large Bayes Net Structures from Sparse Data," in *Proceedings of the twenty-first International Conference on Machine learning (ICML'04)*. ACM Press, 2004, pp. 44–52.

[14] M. Meilă, "An Accelerated Chow and Liu Algorithm: Fitting Tree Distributions to High-Dimensional Sparse Data," in *Proceedings of the Sixteenth International Conference on Machine Learning (ICML'99)*. Morgan Kaufmann Publishers Inc., 1999, pp. 249–257.

[15] A. Whitener and B. Ryker, *MapBasic Developerś Guide*, 1st ed. OnWord Press, 1997.

[16] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2009, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org

[17] S. G. Bottcher and C. Dethlefsen, *deal: Learning Bayesian Networks with Mixed Variables*, 2009, r package version 1.2-33. [Online]. Available: http://www.math.aau.dk/~dethlef/novo/deal

[18] D. Heckerman, "A Tutorial on Learning With Bayesian Networks," Microsoft Research, Tech. Rep., March 1995.