Proc. 13th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'09), Bangkok, Thailand, 2009.

Hot Item Detection in Uncertain Data

Thomas Bernecker, Hans-Peter Kriegel, Matthias Renz, Andreas Zuefle
{bernecker,kriegel,renz,zuefle}@dbs.ifi.lmu.de

Institute for Informatics, Ludwig-Maximilians-Universität München, Germany

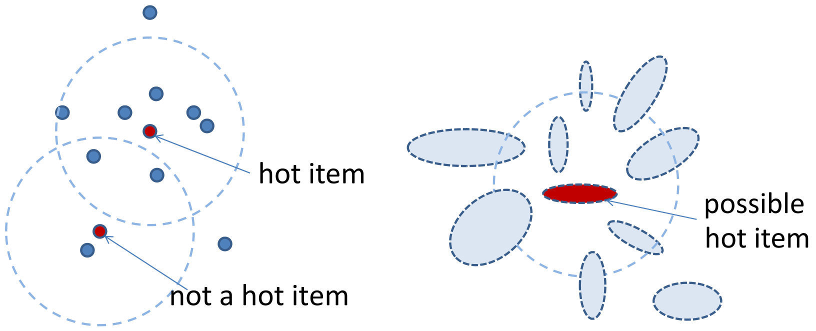
Abstract. An object o of a database $\mathcal{D}$ is called a *hot item*, if there is a sufficiently large population of other objects in $\mathcal{D}$ that are similar to o. In other words, *hot items* are objects within a dense region of other objects and provide a basis for many density-based data mining techniques. Intuitively, objects that share their attribute values with a lot of other objects could be potentially interesting as they show a typical occurrence of objects in the database. Also, there are a lot of application domains, e.g. sensor databases, traffic management or recognition systems, where objects have vague and uncertain attributes. We propose an approach for the detection of potentially interesting objects (*hot items*) of an uncertain database in a probabilistic way. An efficient algorithm is presented which detects *hot items*, where to each object o a confidence value is assigned that reflects the likelihood that o is a *hot item*. In an experimental evaluation we show that our method can compute the results very efficiently compared to its competitors.

1 Introduction

The detection of objects which build dense regions with other objects within a feature space is a foundation of several density-based data mining techniques, in particular density-based clustering [8], outlier detection and other density-based mining applications [11, 13]. We call an object o, for which exists a sufficiently large population of other objects in $\mathcal{D}$ that are similar to o, a *hot item*. Intuitively, an item that shares its attributes with a lot of other items could be potentially interesting as its shows a typical occurrence of items in the database. Application areas where the detection of *hot items* is potentially important exemplarily include scientific applications, e.g. astrophysics, biomedical, sociological and economic applications.

The applications mentioned above require special methods supporting the efficient search in modern databases that may contain not-standard data. Modern databases have to cope with uncertain or imprecise data. Example applications are location determination and proximity detection of moving objects, similarity search and pattern matching in sensor databases or personal identification and recognition systems based on video images or scanned image data. Several approaches that cope with uncertain objects have been proposed [6, 14, 15, 9]. The proposed methods mainly address efficient solutions for similarity search on uncertain data including probabilistic distance range, k-nearest neighbor and ranking. To the best of our knowledge there does not exist any approach addressing retrieval of *hot items* in uncertain domains.

A *hot item* o has the property that the number of other items (objects) which are in the proximity of o, i.e. are similar to o, exceed a given minimal population value. In this



(a) Hot items in certain data.

(b) Hot items in uncertain data.

Fig. 1. Examples of hot items.

paper, we give a more general definition of hot items by relaxing the distance/similarity predicate between the objects.

Definition 1 (Hot Item). *Given a database $\mathcal{D}$ with objects and a minimum population threshold min_items. Furthermore, we assume a score function $d_{score} : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}_0^+$ which is defined on pairs of objects in $\mathcal{D}$ and a predicate $\Phi_\varepsilon : \mathbb{R}_0^+ \rightarrow \{true, false\}$, where $\Phi_\varepsilon \in \{< \varepsilon, \leq \varepsilon, = \varepsilon, \geq \varepsilon, > \varepsilon\}$ and $\varepsilon \in \mathbb{R}_0^+$ is a given scalar. An object $o \in \mathcal{D}$ is called* hot item*, iff there exist at least min_items objects $o' \in \mathcal{D} \backslash \{o\}$ which fulfill the predicate Φ_ε, formally*

$$|\{o' \in \mathcal{D} \backslash \{o\} : \Phi_\varepsilon(d_{score}(o, o')) = true\}| \geq min\_items \Leftrightarrow o \textit{ is a } \textbf{\textit{hot item}}.$$

In the case of uncertain objects, an exact score cannot be determined, particularly if the score relates to the object attributes which are assumed to be uncertain. Consequently, uncertain objects lead to uncertain scores which in turn lead to uncertain predicate results. Thus, the result of the predicate Φ_ε is no longer binary and instead yields a probability value. This probabilistic predicate result can be estimated. Based on this estimation we are able to compute for each object o of an uncertain database a probability value which reflects the likelihood that o is a hot item or not. A formal definition of probabilistic hot item detection is given later in Section 3. The solution for the efficient computation of hot item probabilities can be found in Section 4.

2 Related Work

In the context of this paper, hot items can be abstracted to objects that fulfill a given predicate together with a reasonably large set of other items. If we assume the *equality* predicate, i.e. $\Phi_\varepsilon(d_{score}) := d_{score} = 0$, then a hot item satisfies the *frequent item* property. The detection of frequent items or frequent itemsets as a preprocessing step for rule mining is one of the most important problems in data mining. Chui et al. study in [7] the problem of mining frequent itemsets from uncertain data. They assume transactions whose items are associated with existential probabilities and introduce the U-Apriori algorithm, which is a modified version of the Apriori algorithm. They present a framework which probabilistically computes frequent items in an efficient way.

The aspect to identify objects that are similar to a given amount of other objects is the basis of several density-based algorithms for discovering clusters and outliers. There exist approaches for density-based clustering of uncertain data, e.g. [10] which are quite related to our approach. However the proposed model used to determine the probabilistic density does not respect the mutual exclusiveness of alternative attribute values. The missing conditional probability in their approach leads to approximative results only which disqualifies this approach from the accurate detection of hot items.

A lot of work has been published for managing uncertain data [4, 5, 14], probabilistic similarity queries [9] and quite recently for probabilistic top-k queries [12, 15]. The detection of hot items can be efficiently supported by a similarity join query used in a preprocessing step, in particular the distance range self-join. Approaches for an efficient join on uncertain data are proposed in [9]. The main advantage of this approach is that sampled positions in space can efficiently be indexed using traditional spatial access methods thus allowing to reduce the computational complexity of complex query types. Our approach exploits the similarity join approach proposed in [9]. However, the cost of the probabilistic detection of hot items are originally highly CPU-bound which is demonstrated in our experimental evaluation (cf. Section 5). The advantage of an I/O cost efficient approach for the preprocessing step only becomes noticeable when applying the methods proposed in this paper such that the CPU cost less outbalance the overall query cost.

3 Problem Definition

In this section, we formally introduce the problem of probabilistic identification of hot items in uncertain databases.

3.1 Probabilistic Score

The identification whether an object is a hot item or not requires to know the neighborhood of the object according to a given (similarity) distance score function. Assuming that the object attributes the score function relates to are uncertain, then the score result is uncertain, too. Therefore, we require a probabilistic score function which is defined as follows: Let $P_{\Phi_\varepsilon} : \mathcal{D} \times \mathcal{D} \rightarrow [0, 1]$ be a probabilistic function defined on a pair of objects that returns the likelihood that a given score w.r.t. both objects fulfills a given predicate Φ_ε. For example, if we use the distance $d(o_i, o_j)$ between two uncertain vector objects o_i and o_j as score function and we use the predicate $\Phi_\varepsilon = d(o_i, o_j) \leq \varepsilon$, then $P_{\Phi_\varepsilon}(o_i, o_j)$ denotes the probability that o_j is within the ε-range of o_i and vice versa.

3.2 Probabilistic Hot Items

Based on the definitions given above, we can compute hot items in uncertain data in a probabilistic way. However, we have to solve the problem of dependencies of the uncertain attributes. Though we assume that the attributes of uncertain objects are independent of each other, we have to respect that the values of an uncertain object attribute are mutually exclusive. For this reason, first we have to define probabilistic hot items based on a conditional probability.

Definition 2 (Conditional Probabilistic Hot Item). *Given a database $\mathcal{D}$ with uncertain objects and a minimum population threshold min_items. Furthermore, we assume a predicate $\Phi_\varepsilon : \mathbb{R}_0^+ \rightarrow \{true, false\}$ which is defined on a probabilistic score function, where $\Phi_\varepsilon \in \{< \varepsilon, \leq \varepsilon, = \varepsilon, \geq \varepsilon, > \varepsilon\}$ and $\varepsilon \in \mathbb{R}_0^+$ is a given scalar. Under the condition that an uncertain object $o \in \mathcal{D}$ is equal to a certain vector $x \in \mathbb{R}^d$, the probability that o is a* hot item *can be computed by*

$$P(o \textit{ is a } \textbf{\textit{hot item}} | o = x) =$$

$$P(|\{o' \in \mathcal{D} \backslash \{o\} : \Phi_\varepsilon(d_{score}(o, o')) = \textit{true}\}| \geq min\_items) =$$

$$\sum_{\substack{S_{min\_items} \subseteq \mathcal{D} \backslash \{o\} \\ |S_{min\_items}| \geq min\_items}} \Big(\prod_{o' \in S_{min\_items}} P_{\Phi_\varepsilon}(o, o') \cdot \prod_{o' \in \mathcal{D} \backslash (S_m in\_items \cup \{o\})} (1 - P_{\Phi_\varepsilon}(o, o')) \Big).$$

The above definition gives rise to the following general definition of probabilistic hot items which depends on the used uncertainty model. The probability P(o is a **hot item**) of an object o being an (unconditionally) probabilistic hot item can be computed by aggregating the conditional hot item probabilities over all possible instances x of o multiplied with the probability that object o corresponds to x, i.e.

$$\sum_{x \in \mathbb{R}^d} P(o = x) \cdot P(|\{o' \in \mathcal{D} \backslash \{o\} : \Phi_\varepsilon(d_{score}((x, 1), o')) = \text{true}\}| \geq min\_items).$$

4 Hot Item Detection Algorithm

Let $\mathcal{D}$ be a database with uncertain objects. Each object $o \in \mathcal{D}$ is probed w.r.t. the hot item property. This computation can be split into the preprocessing step which finds candidates that match the predicate Φ_ε and the query step which detects the *hot items*.

4.1 Preprocessing Step

First, for each object $o' \in (\mathcal{D} \backslash \{o\})$ we have to compute the probability that o' fulfills a given predicate $\Phi_\varepsilon \in \{< \varepsilon, \leq \varepsilon, = \varepsilon, \geq \varepsilon, > \varepsilon\}$ w.r.t. object o, i.e. we have to compute $P_{\Phi_\varepsilon}(o, o')$. Obviously, only those objects $o' \in \mathcal{D}' \subset \mathcal{D}$ for which the predicate Φ_ε is fulfilled with a probability greater than zero, i.e. $P_{\Phi_\varepsilon}(o, o') > 0$, have to be taken into account in order to compute the probability $P(o$ is a hot item). Note that, depending on the used predicate Φ_ε, usually only a small portion $\mathcal{D}' \subset \mathcal{D}$ of the database fulfills the predicate $\Phi_\varepsilon(o, o' \in \mathcal{D}')$ with a probability greater than zero. A quick search of those objects which have to be taken into account can be efficiently supported by means of an index structure, e.g. the R*-tree. In particular for the predicate $\Phi_\varepsilon =$ " $\leq \varepsilon$", the index supported ε-range join [2] can be used to speed-up the search as proposed in [3]. Here, approximative representations like the minimal bounding rectangle (mbr) of an uncertain object are very appropriate to be used as index key for a filter step following the multi-step query processing paradigm. A solution for the ε-range join on uncertain data is proposed in [9] which can be used as a preprocessing step for our proposed algorithm for the detection of hot items.

4.2 Query Step

In the following, we introduce our new approach which is able to efficiently compute the probability that an object $o \in \mathcal{D}$ is a hot item. As mentioned above, our algorithm has quadratic runtime or even needs linear time if min_items is assumed to be constant. The key idea of our approach is based on the following property. Given a set of j predicates $\mathcal{S} = \{p_1, p_2, \ldots, p_j\}$ for which the probability $P(p_i)$ that the predicate $p_i \in \mathcal{S}$ is "true" is known, respectively. Now, we want to compute the probability $P_{k,\mathcal{S}}$ that at least k predicates of $\mathcal{S}$ are "true".

Lemma 1. *If we assume that predicate p_j is "true", then $P_{k,\mathcal{S}}$ is equal to the probability that at least $k-1$ predicates of $\mathcal{S}\backslash\{p_j\}$ are "true". Otherwise, $P_{k,\mathcal{S}}$ is equal to the probability that at least k predicates of $\mathcal{S}\backslash\{p_j\}$ are "true".*

The above lemma leads to the following recursion that allows to compute $P_{k,\mathcal{S}}$ by means of the paradigm of dynamic programming:

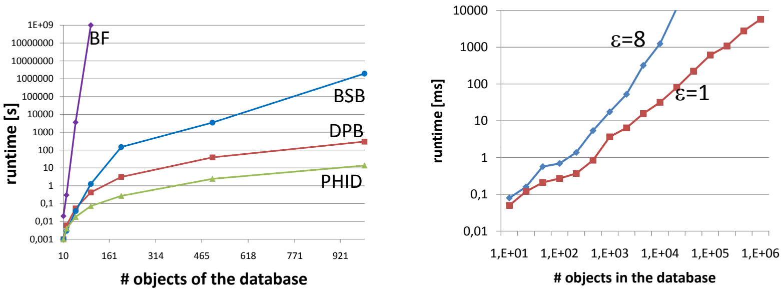
$$P_{k,\mathcal{S}} = P_{k-1,\mathcal{S}\backslash\{p_j\}} \cdot p_j + P_{k,\mathcal{S}\backslash\{p_j\}} \cdot (1 - p_j),$$

where

$$P_{0,\mathcal{S}} = 1.$$

The above dynamic programming scheme is an adaption of a technique previously used in the context of probabilistic top-k queries [15]. Here, we generalize this technique for arbitrary probabilistic predicates. We apply this method to compute the probability that an uncertain object $o \in \mathcal{D}$ is a hot item. Given an uncertain object $o \in \mathcal{D}$, the value for min_items and the set $\mathcal{D}' \subseteq \mathcal{D}$ of objects for which the probability that the predicate $P_{\Phi_\varepsilon}(o', o)$ $(o' \in \mathcal{D}')$ is "true" is greater than zero, i.e. $\forall o' \in \mathcal{D}' : P_{\Phi_\varepsilon}(o, o') > 0$. The probability P that object o is a hot item is equal to the probability $P_{min\_items,\mathcal{D}'}(o)$ that for at least min_items objects $o' \in \mathcal{D}'$ the predicates $\Phi_\varepsilon(o, o')$ are "true". With Lemma 1 and the dynamic programming technique described above we can compute $P = P_{min\_items,\mathcal{D}'}(o)$ efficiently by:

$$P_{min\_items,\mathcal{D}'}(o) = \begin{cases} \begin{array}{l} P_{min\_items-1,\mathcal{D}'\backslash\{o'\}}(o) \cdot P_{\Phi_\varepsilon}(o, o') + \\ P_{min\_items,\mathcal{D}'\backslash\{o'\}}(o) \cdot (1 - P_{\Phi_\varepsilon}(o, o')) \end{array} & \text{, if } min\_items > 0 \\ \qquad 1 & \text{, if } min\_items = 0. \end{cases}$$



(a) Evaluation of competing techniques.

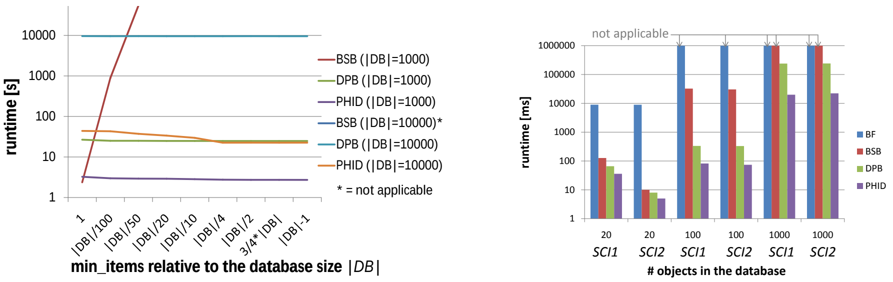
(b) Scalability experiments.

Fig. 2. Performance w.r.t database size.

5 Experimental Evaluation

In this section, we present the results of an experimental evaluation of the proposed methods w.r.t. efficiency. First we specify the used datasets and experimental setup. In the artificial *ART* dataset, each object is represented by a set of positions sampled from an individual five-dimensional hyper-rectangle R with a given size. The samples are uniformly distributed within the rectangles. The rectangles are arbitrarily distributed within the object space. Each of the 1500 objects of the two real-world datasets *SCI1* and *SCI2* consists of 10 samples, where each sample corresponds to a set of environmental sensor measurements of one single day that consist of several dimensions (attributes). The attribute set of *SCI1* describes temperature, humidity and CO concentration, whereas *SCI2* has a larger set of attributes (temperature, humidity, speed and direction of wind as well as concentrations of CO, SO_2, NO, NO_2 and O_3).

In this section, we compare two variants of our approach denoted by *DPB* and *PHID*. In contrast to *PHID*, *DPB* applies dynamic programming on the complete database, i.e. $\mathcal{D}' = \mathcal{D}$ and, thus, does not require the pre-processing step. The performance of *PHID* and *DPB* is compared to that of the brute-force solution (*BF*) by simply applying the formulas given in Section 3.2. Furthermore, we compare them to the bisection-based method (*BSB*) which is adapted to the method proposed in [1]. This method is able to significantly speed-up computation compared to the brute-force method, but is still exponential. Note that in our algorithm, we concentrate on the evaluation of the CPU-cost only. The reason is that the *PHID*-algorithm is clearly CPU-bound. The only I/O bottleneck is the initial computation of the likelihood that o is in the ϵ-range of $s_{o'}$, for each object $o \in DB$ and each sample $s_{o'}$, where $o' \in DB$ and $o \neq o'$. This requires a distance-range-self-join of the database which can be performed by a nested-block-loop join that requires $O(|DB|^2)$ page-faults in the worst case. In contrast, the CPU time for the *PHID*-algorithm is cubic: Each call of the dynamic programming algorithm requires $O(|DB|^2)$ time and has to be performed once for each sample in the database.



(a) Influence of min_items.

(b) Experiments on real-world data.

Fig. 3. Performance experiments.

The first experiments relate to the scalability of the proposed approaches. The results depicted in Figure 2 demonstrate how the runtime of the competing techniques is influenced by the database size. Figure 2(a) shows that, though the bisection-based approach has exponential runtime, it outperforms the brute-force approach by several orders of magnitude. However, the dynamic-programming-based approaches scale significantly better than their competitors which in contrast to *DPB* and *PHID* have exponential runtime. Furthermore, the pre-processing step of *PHID* obviously pays off. The performance can be further improved by an order of magnitude when applying the dynamic-programming technique only on objects o' where the probabilistic predicate $P_{\Phi_\varepsilon}(o, o')$ is not zero. The next experiment shows the scalability of *PHID* for different ε-range values. Here, the average time required to compute the hot item probability for an object was measured. The results shown in Figure 2(b) demonstrate that *PHID* scales well, even for very large databases. Figure 3(a) demonstrates the performance w.r.t. the min_items value for different database sizes. Contrary to *DPB* and *PHID*, the *BSB* method is very affected by the min_items value due to the expensive probability computation. The slight increase of the *DPB* and *PHID* performances can be explained by the reduced number of hot items with increasing min_items value.

Finally, we evaluate the performance based on real-world data (cf. Figure 3(b)). Unlike the exponential algorithms, *DPB* and *PHID* are able to perform a full hot item scan of the database in reasonable time, even for a relatively large database size.

6 Conclusions

In this paper, we propose an efficient approach for probabilistic queries for hot items, i.e. objects for which at least min_items other objects exist which are similar to o. In particular our approach computes for each object o in an uncertain database the probability that o is a hot item. We proposed methods that are able to break down the high computational complexity required to compute for an object o the probability, that o is a hot item. We theoretically and experimentally show that our approach can efficiently solve the problem (in worst-case $O(n^3)$) while the competing techniques have exponential runtime. Thereby, we achieve a speed-up of several orders of magnitude.

References

1. T. Bernecker, H.-P. Kriegel, and M. Renz. Proud: Probabilistic ranking in uncertain databases. In *In Proc. 20th Int. Conf. on Scientific and Statistical Database Management (SSDBM'08), Hong Kong, China, July 9-11*, pages 558–565, 2008.
2. T. Brinkhoff, H.-P. Kriegel, and B. Seeger. Efficient processing of spatial joins using r-trees. *SIGMOD Rec.*, 22(2):237–246, 1993.
3. C. Böhm, B. Braunmüller, M. M. Breunig, and H.-P. Kriegel. "Fast Clustering Based on High-Dimensional Similarity Joins". In *Int. Conf. on Information Knowledge Management (CIKM)*, 2000.
4. C. Böhm, A. Pryakhin, and M. Schubert. "The Gauss-Tree: Efficient Object Identification of Probabilistic Feature Vectors". In *Proc. 22nd Int. Conf. on Data Engineering (ICDE'06)),Atlanta,GA,US*, page 9, 2006.
5. R. Cheng, D. Kalashnikov, and S. Prabhakar. "Evaluating Probabilistic Queries over Imprecise Data". In *Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'03), San Diego, CA)*, pages 551–562, 2003.
6. R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J. Vitter. "Efficient Indexing Methods for Probabilistic Threshold Queries over Uncertain Data". In *Proc. 30th Int. Conf. on Very Large Databases (VLDB'04), Toronto, Canada*, pages 876–887, 2004.
7. C.-K. Chui, B. Kao, and E. Hung. Mining frequent itemsets from uncertain data. In *In Proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-2007), Nanjing, China, May 22-25*, 2007.
8. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96), Portland, OR*, 1996.
9. H.-P. Kriegel, P. Kunath, M. Pfeifle, and M. Renz. "Probabilistic Similarity Join on Uncertain Data". In *Proc. 11th Int. Conf. on Database Systems for Advanced Applications (DASFAA'06), Singapore, Singapore, pp. 295-309*, 2006, (Best paper).
10. H.-P. Kriegel and M. Pfeifle. Density-based clustering of uncertain data. In *Proc. 11th Int. Conf. on Knowledge Discovery and Data Mining (KDD'05), Chicago, IL, pp. 672-677*, 2005.
11. H.-P. Kriegel, M. Renz, M. Schubert, and A. Züfle. Statistical density prediction in traffic networks. In *SDM*, pages 692–703, 2008.
12. M. Soliman, I. Ilyas, and K. Chen-Chuan Chang. "Top-k Query Processing in Uncertain Databases". In *Proc. 23rd Int. Conf. on Data Engineering (ICDE'07), Istanbul, Turkey*, pages 896–905, 2007.
13. J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Proceedings of PAKDD-02*, 2002.
14. Y. Tao, R. Cheng, X. Xiao, W. Ngai, B. Kao, and S. Prabhakar. "Indexing Multi-Dimensional Uncertain Data with Arbitrary Probability Density Functions". In *Proc. 31th Int. Conf. on Very Large Data Bases (VLDB'05), Trondheim, Norway*, pages 922–933, 2005.
15. K. Yi, F. Li, G. Kollios, and D. Srivastava. "Efficient Processing of Top-k Queries in Uncertain Databases". In *Proc. 24th Int. Conf. on Data Engineering (ICDE'08), Cancún, México*, 2008.