Measuring and Evaluating Dissimilarity in Data and Pattern Spaces

Irene Ntoutsi, Yannis Theodoridis

Database Group, Information Systems Laboratory
Department of Informatics, University of Piraeus, Greece
{ntoutsi, ytheod}@unipi.gr

Abstract

Nowadays, the amount of patterns extracted from Knowledge Discovery and Data Mining (KDD) is rapidly growing, thus imposing new challenges regarding their management. One of the most important operations on pattern sets (or pattern bases) is that of similarity comparison. In our research, we investigate issues regarding pattern comparison and the degree of distance preservation in pattern space with respect to the original data space.

1. Motivation

In our days a huge quantity of raw data is collected from different application domains (business, scientific, etc.). Due to their quantity and complexity, it is not simple for humans to thoroughly investigate these data collections. Knowledge Discovery and Data Mining (KDD) provides a solution to this problem by generating compact and rich in semantics representations of raw data, called *patterns* (decision trees, association rules, frequent itemsets, clusters, neural networks etc.) [3]. Patterns preserve information existing in the original raw dataset; however, the degree of preservation strongly depends on the parameters of the KDD algorithm used for their extraction. For example, in frequent itemset mining one can get different subsets of the itemset lattice by tuning the minimum support threshold, in classification one can get different sub-trees of the complete decision tree by applying different pruning levels (in order to avoid overfitting), and so on.

The need for efficient management of patterns has become compulsory nowadays due to the spreading of the data mining technology. Management includes representation, storage, retrieval, indexing and visualization issues. If possible, patterns should be considered as "first-class citizens" in data / pattern management systems [3].

One of the most important operations on patterns is that of comparison, i.e. detecting how similar two patterns are to each other. Defining a similarity / distance operator for patterns is not straightforward since distance measures for several pattern types must be defined (for example, decision trees, clusters, association rules and even clusters of association rules) (Figure 1). Apart from the dissimilarity between patterns of the same type, the dissimilarity between patterns of different types (e.g. between a decision tree and a cluster) should be considered as well. Another interesting issue is whether and how dissimilarity in pattern space depends on the dissimilarity in the original data space (where patterns were extracted from).

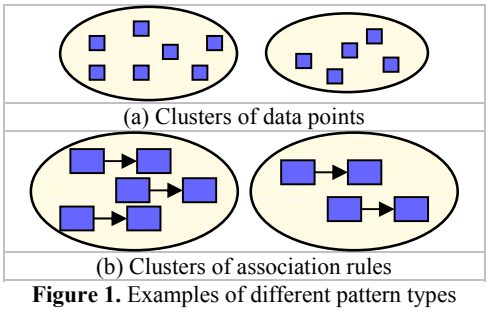

Figure 1. Examples of different pattern types

The aim of this research is to study pattern comparison issues and examine the feasibility of a generic framework for pattern comparison. Furthermore, issues regarding the preservation of distance from data to pattern space will be investigated.

2. Importance – Applications – Related Work

As already mentioned, the similarity / distance operation is one of the most important among

several interesting operations that could be defined over patterns and has many applications.

Defining a similarity operator between patterns could be used to express similarity queries over a pattern set (or pattern-base) including *k-nearest neighbour queries* (i.e. find the *k*-most similar pattern(s) to a query pattern) and *range queries* (i.e. find the most similar pattern(s) to a given pattern within a given range). The efficient computation of dissimilarity is one of the core issues of a Pattern Base Management System (PBMS) with applications in indexing, as well as in retrieval and visualization [3], [8].

Another application is *monitoring* and *detecting pattern changes* (e.g. detecting changes in customers' behaviour over time). This is useful, for example, in *KDD synchronization*, in order to keep patterns up to date with respect to the corresponding raw data (e.g. synchronizing patterns only when the corresponding raw data have significantly changed). It is also useful to pattern base *versioning support* (e.g. getting a differential backup of the pattern base or comparing consequent versions of a pattern base).

A common technique for the comparison of datasets involves the corresponding pattern sets extracted from these datasets through a data mining algorithm (e.g. comparing images – in data space – through a set of features – in pattern space) [2], [5], [6], [7]. This approach sounds reasonable since patterns reflect the information contained in raw data and therefore dissimilarity in pattern space could be considered as a measure of dissimilarity in the original data space. Discovering a *mapping* (either *exact* or *approximate*) between the dissimilarity in data and pattern space is really useful. For example, we could avoid the hard task of comparing the original datasets whenever the corresponding pattern sets are available for comparison or even avoid the hard task of mining a dataset whenever it is found to be similar to another dataset for which the results of mining are already available.

Another application refers to the *distributed data mining* domain [5], [6], [7]. Here, mining is not centralized since locally strong patterns should be preserved. A common approach is to "cluster" similar datasets and mine afterwards each cluster independently [6]. For the comparison of the datasets, even the original data space or the corresponding pattern space could be involved [6]. Another idea is based on mining each database independently and "cluster" afterwards the sets of patterns based on their dissimilarity. This is also applicable in the secure mining domain since only patterns, and not the original raw data are required. Depending on the mining parameters, it is hard to recover the original raw dataset (in [9], it is proved that deciding whether there is a dataset compatible with a given set of frequent itemsets is NP-hard).

Currently, the experimental comparison between two algorithms (or the same algorithm with different criteria) is limited to interpreting 'visual' representations of the results. However, algorithms' evaluation could be achieved by comparing their results over the same raw dataset. Depending on the dataset characteristics (e.g. dense vs. sparse datasets) an algorithm might be more suitable than another. Therefore, one might need to experiment with many algorithms and evaluate their results in order to conclude to the algorithm that best fits his/her needs (Figure 2).

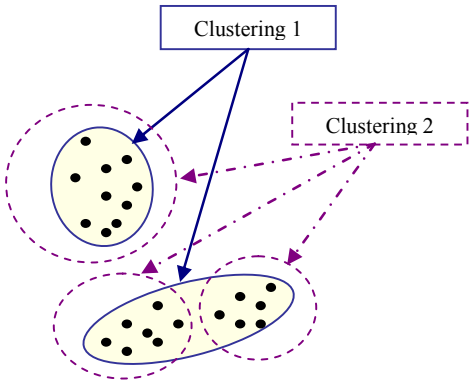

Figure 2. Comparing results of data mining algorithms over the same raw dataset

As an extension of this application consider the scenario where one might want to acquire a specific target pattern set from a given set of raw data.

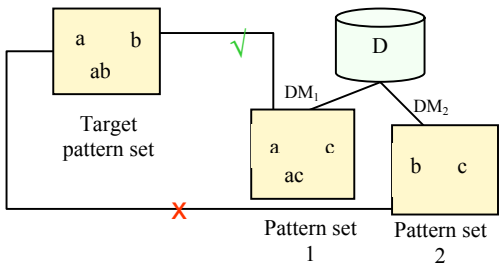

Figure 3. Comparing results of data mining algorithms with respect to a target pattern set

One solution is to apply different data mining algorithms (or even the same algorithm with different criteria e.g. minimum support in case of frequent itemset mining) and keep the one that its output is most similar to the target pattern set (Figure 3).

Besides ad hoc approaches for particular cases, to the best of our knowledge the single related work to define a framework for the comparison of patterns is proposed in [2]. In this work, authors propose FOCUS, a framework for measuring the deviation between two datasets in terms of the pattern sets ('models' in authors' terminology) they induce. The central idea of their work is to first represent patterns induced by the datasets in terms of a structure and a measure component (called, 2-component property). The structure component identifies "interesting regions" and the measure component summarizes the subset of the data that is mapped to each region. If the patterns have different structure components, a first step is needed to make them identical by extending them to their greatest common refinement (GCR) - in case of frequent itemsets, for example, the GCR of two sets of itemsets is their union. Then the deviation between the datasets is considered to be equal to the deviation between them over the set of all regions in the GCR. A difference function that calculates the deviation of two regions with identical structure and an aggregate function that aggregates all these differences are required. Referring to the difference function, authors in [2] provide two instantiations, the absolute difference and the scaled difference function, whereas for the aggregate function *sum* and *max* could be used. By tuning the difference and the aggregate functions, various distance functions can be defined within this framework.

The FOCUS framework assumes that the GCR of the patterns to be compared can be defined and provides, as examples, frequent itemsets, decision trees and clusters. Furthermore, the computation of pattern deviation requires the measures of all regions in GCR to be computed with respect to both datasets, so the comparison in pattern space also involves the original data space.

3. Research Agenda and Preliminary Results

3.1 A framework for pattern comparison

Our approach to the problem of defining a general framework for pattern comparison is based on the logical model proposed in [8] where each pattern type *pt* includes a structure schema *ss*, defining the pattern space, and a measure schema *ms*, describing the measures that quantify the quality of the source data representation achieved by each pattern. (We mention only the components of the framework used for the assessment of dissimilarity between patterns - see [8] for a detailed description of the model.) A pattern *p* of type *pt* instantiates the structure schema and the measure schema, thus leading to a structure, $p.s$, and a measure, $p.m$.

We distinguish two types of dissimilarity between patterns depending on the structure of the patterns to be compared: comparison between *simple patterns* (e.g. between two association rules) and comparison between *complex patterns* - complex patterns are patterns whose structure consists of other patterns, for example a cluster of association rules (see Figure 1). The notion of dissimilarity can be further distinguished into that *between patterns of the same pattern type* (e.g. two decision trees, two clusters etc.) and that *between patterns of different pattern types* (e.g. a decision tree with a cluster).

In [1], we propose a framework for the assessment of dissimilarity between either simple or complex patterns. We adopt the 2-component property introduced by FOCUS [2], thus we express patterns in terms of a structure and a measure component. The dissimilarity between patterns is computed by taking into account both the dissimilarity between their structures and the dissimilarity between their measures.

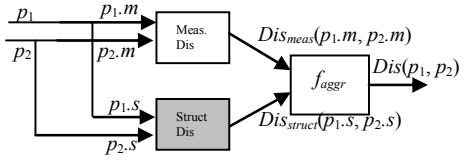

Figure 4. Measuring dissimilarity between patterns

As is illustrated in Figure 4, the dissimilarity is evaluated by aggregating dissimilarities between measure and structure components, by means of an *aggregation function* f_{aggr}.

According to [1], the dissimilarity between two simple patterns of the same type *pt* can be computed as follows:

$$Dis(p_1, p_2) = f_{aggr}(dis_{struct}(p_1.s, p_2.s), dis_{meas}(p_1.m, p_2.m))$$

If the two patterns have the same structural component, then their dissimilarity equals to the measures' dissimilarity. In the general case, however, the patterns to be compared have different structural components, thus a preliminary step is needed to "reconcile" the two structures so as to make them comparable.

Evaluation of dissimilarity between complex patterns follows the same basic rationale shown in Figure 4. However, the structure of complex patterns now consists of several other patterns. In our framework, the dissimilarity between structures of complex patterns depends in turn on the dissimilarity between component patterns. Dissimilarity is conceptually evaluated in a bottom-up fashion, and can be adapted to specific needs/constraints by acting on two fundamental abstractions:

- the *coupling type*, which is used to establish how component patterns can be matched;
- the *aggregation logic*, which is used to combine the dissimilarity scores obtained for coupled component patterns into a single overall score representing the dissimilarity between the complex patterns.

Depending on the instantiations of the different blocks of the framework (coupling type, aggregation logic, structure dissimilarity and measure dissimilarity) various distance functions configurations can be defined within it.

Our framework extends FOCUS, which is limited to the comparison of patterns for which the GCR can be defined, since it allows for a wide variety of matching criteria (*coupling type*). Furthermore, unlike FOCUS, our framework supports the recursive definition of arbitrarily complex patterns. Also, it works exclusively in the pattern space, in contrast to FOCUS that involves both the pattern and the data space from which patterns were extracted. Such a property is useful in case of an autonomous PBMS [3].

We are currently working on comparing sets of frequent itemsets using the framework idea. According to preliminary results, our framework can capture the "controlled" changes of the pattern set, as illustrated in Figure 5.

The framework proposed in [1] could be extended towards two directions:

a) support the comparison between patterns of different types and

b) relate the distance in pattern space with the distance in the corresponding raw data space.

As a scenario referring to the first direction, consider the comparison between a decision tree and a clustering scheme. As a scenario of the second direction, consider studying how the parameters of the data mining algorithm affect the information preserved in the extracted patterns (lossless vs. lossy transformations).

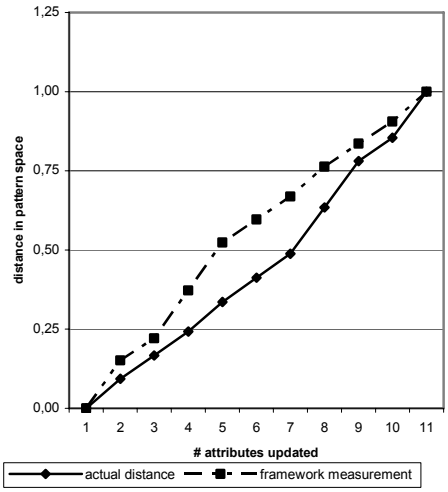

Figure 5. Preliminary results on capturing changes in pattern space

3.2 Dissimilarity between different pattern types

The comparison between patterns of different types is an open and challenging issue since, to our knowledge, there is no related work. We plan to transform patterns into a common basis (a graph model could be used for this purpose) and perform the comparison between the transformed patterns using the framework. Another idea is to transform patterns of one type to the other type, when applicable, and reduce the problem to that of comparing patterns of the same pattern type (e.g. when comparing a decision tree with a set of rules, the decision tree could be transformed firstly to the equivalent set of rules). We are currently investigating both directions.

3.3 Relating dissimilarity in data and pattern spaces

We are currently working on extending the framework proposed in [1] so as to relate dissimilarity in pattern space with that in the corresponding data space. From our point of view, the KDD procedure can be thought of as a transformation task that transforms data into knowledge. Depending on the transformation parameters, i.e. data mining algorithms parameters, we distinguish between lossless and lossy transformations. The former preserve information in the pattern space (e.g. an un-pruned decision tree that completely fits on the

training set or an itemset lattice generated with no minimum support threshold), whereas the latter lose some of the information existing in the original data space (e.g. a pruned decision tree or an itemset lattice generated with a minimum support threshold).

Ideally, we would like the dissimilarity in pattern space to follow that in the original data space. Indeed, preliminary results regarding the frequent itemsets case illustrate such behaviour.

Finally, we plan to manifest the applicability and generality of our framework by including scenarios of comparing patterns in different application domains (decision trees, time series, moving object trajectories, web site structures and contents).

4. Concluding Remarks

In this paper we presented our ongoing work on measuring and evaluating dissimilarity in data and pattern spaces. Through application examples, we demonstrated the importance of defining a general framework for the comparison of patterns. Towards this aim, we presented a framework for measuring dissimilarity between both simple and complex patterns.

Next steps include:

i) manifest the applicability of the framework by applying it to other pattern types apart from clusters and frequent itemsets (e.g. decision trees and moving objects trajectories, web sites contents)

ii) extend the framework so as to support comparison between patterns of different pattern types, and

iii) extend the framework so as to relate dissimilarity in pattern space with that in data space from which patterns were extracted.

Our ultimate goal is a generic framework for measuring and evaluating dissimilarity in data and pattern spaces.

5. Acknowledgments

This research is partially supported by the Greek Ministry of Education and the European Union under a grant of the "Heracletos" EPEAEK II Programme (2003-06).

6. References

[1] I. Bartolini, P. Ciaccia, I. Ntoutsi, M. Patella, and Y. Theodoridis. "A Unified and Flexible Framework for Comparing Simple and Complex Patterns". In *Proceedings of PKDD'04 Conference*, Pisa, Italy, 2004.

[2] V. Ganti, J. Gehrke, R. Ramakrishnan, and W.-Y. Loh. "A Framework for Measuring Changes in Data Characteristics". In *Proceedings of PODS'99 Symposium*, Philadelphia, PA, USA, 1999.

[3] Y. Theodoridis, M. Vazirgiannis, P. Vassiliadis, B. Catania, and S. Rizzi. "A manifesto for pattern bases". PANDA Technical Report TR-2003-03, 2003. Available at `http://dke.cti.gr/panda`.

[4] C. Faloutsos, M. Ranganathan and Y. Manolopoulos. "Fast Subsequence Matching in Time-Series Databases". In *Proceedings of ACM SIGMOD'94 Conference*, Minneapolis, MN, USA, 1994.

[5] S. Parthasarathy and M. Ogihara. "Clustering Distributed Homogeneous Datasets", In *Proceedings of PKDD'00 Conference*, Lyon, France, 2000.

[6] T. Li, S. Zhu, and M. Ogihara. "A New Distributed Data Mining Model Based on Similarity", In *Proceedings of ACM-SAC'03 Symposium*, Melbourne, FL, USA, 2003.

[7] X. Wu, C. Zhang, and S. Zhang. "Database classification for multi-database mining", *Information Systems*, 30(2005) pages 71-88.

[8] S. Rizzi, E. Bertino, B. Catania, M. Golfarelli, M. Halkidi, M. Terrovitis, P. Vassiliadis, M. Vazirgiannis, and E. Vrachnos. "Towards a Logical Model for Patterns". In *Proceedings of ER'03 Conference*, Chicago, IL, USA, 2003.

[9] T. Mielikäinen. "On Inverse Frequent Set Mining". In *Proceedings of PPDM'03 Workshop*, Melbourne, FL, USA, 2003.