in: Proc. 11th Int. Symp. on Spatial and Temporal Databases (SSTD 2009), Aalborg, Denmark, 2009, 436-440.

ELKI in Time: ELKI 0.2 for the Performance Evaluation of Distance Measures for Time Series

Elke Achtert, Thomas Bernecker, Hans-Peter Kriegel, Erich Schubert, and Arthur Zimek

Ludwig-Maximilians-Universität München
Oettingenstr. 67, 80538 München, Germany
http://www.dbs.ifi.lmu.de
{achtert,bernecker,kriegel,schube,zimek}@dbs.ifi.lmu.de

Abstract. ELKI is a unified software framework, designed as a tool suitable for evaluation of different algorithms on high dimensional real-valued feature-vectors. A special case of high dimensional real-valued feature-vectors are time series data where traditional distance measures like L_p-distances can be applied. However, also a broad range of specialized distance measures like, e.g., dynamic time-warping, or generalized distance measures like second order distances, e.g., shared-nearest-neighbor distances, have been proposed. The new version ELKI 0.2 now is extended to time series data and offers a selection of these distance measures. It can serve as a visualization- and evaluation-tool for the behavior of different distance measures on time series data.

1 Introduction

In high dimensional data, and especially in time series data, the choice of a distance measure suitable and meaningful w.r.t. the data in question is essential. In many implementations of algorithms, either provided by authors or implemented in general frameworks, the Euclidean distance is invariably used as a standard distance measure.

In the software system described in this paper, we facilitate the use of a wide range of different algorithms along with a wide choice of distance measures. The framework provides the data management independently of the tested algorithms. So all algorithms and distance measures are comparable on equal conditions. But even more important is an intuitive and easy-to-understand programming style to invite additional contributions. This way, the interested users can easily provide a new algorithm or a new customized distance measure and compare their performance with existing solutions.

2 An Overview on the Software System

Focus and strength of Weka [1] and YALE [2] as popular environments for data mining algorithms is mainly in the area of classification, while clustering approaches are somewhat under-represented. However, first steps towards incorporating subspace clustering into Weka have been presented recently [3]. Although

both, Weka and YALE, support the connection to external database sources, they are based on a flat internal data representation. Thus, experiments assessing the impact of an index structure on the performance of a data mining application are not possible using these frameworks. Furthermore, in both frameworks, the user often cannot select the distance measure to be used by a certain algorithm but the distance computation is coded deeply in the implementation of an algorithm. Especially, neither Weka nor YALE do support special requirements like specialized distance measures for time series data. On the other hand, frameworks for index structures, such as GiST [4], do not provide any precast connection to data mining applications. Finally, data mining tools specialized for time series data, like T-Time [5], do support specialized distance measures for time series but usually provide only a small selection of algorithms for clustering or classification of time series data that can be used in combination with suitable distance measures.

To combine these different aspects in one solution, we built the Java Software Framework ELKI (**E**nvironment for Deve**L**oping **K**DD-Applications Supported by **I**ndex Structures). ELKI version 0.1 [6] already comprised a profound and easily extensible collection of algorithms for data mining applications, such as item-set mining, clustering, classification, and outlier-detection. On the other hand, ELKI incorporates and supports arbitrary index structures to support even large, high dimensional data sets. But ELKI does also support the use of arbitrary data types and respective distance functions. Thus, it is a framework suitable to support the development and evaluation of new algorithms at the cutting edge of data mining as well as to incorporate experimental index structures or to develop and evaluate new distance measures, e.g., to support complex data types.

ELKI intends to ease the development of new algorithms and new distance measures by providing a wealth of helper classes and methods for algebraic and analytic computations, and simulated database support for arbitrary data types using an index structure at will.

2.1 The Environment: A Flexible Framework

As a framework, our software system is flexible in a sense, that it allows to read arbitrary data types (provided there is a suitable parser for your data file or adapter for your database), and supports the use of any distance or similarity measure appropriate for the given data type. Generally, an algorithm needs to be provided with a distance function of some sort. Thus, distance functions connect arbitrary data types to arbitrary algorithms.

The architecture of the software system separates data types, data type-specific distance measures, data management, and data mining applications. So, different tasks can be implemented independently. A new data type can be implemented and used by many algorithms, given a suitable distance function is defined. An algorithm will perform its routine irrespectively of the data handling which is encapsulated in the database. A database may facilitate efficient data

management via incorporated index structures. Index structures are encapsulated in database objects. These database objects facilitate range queries using arbitrary distance functions. Algorithms operate on database objects irrespective of the underlying index structure. So the implementation of an algorithm, as pointed out above, is not concerned with the details of handling the data which can be supported by arbitrary efficient procedures.

2.2 Arbitrary Distance Measures

Often, the main difference between clustering algorithms is the way to assess the distance or similarity between objects or clusters. So, while other data mining systems usually predefine the Euclidean distance as the only possible distance between different objects, ELKI allows to flexibly define and use any distance measure. This way, for example, subspace clustering approaches that differ mainly in the definition of distance between points (like e.g. COPAC [7] and ERiC [8]) can use the same algorithmic routine and become, thus, highly comparable in their performance.

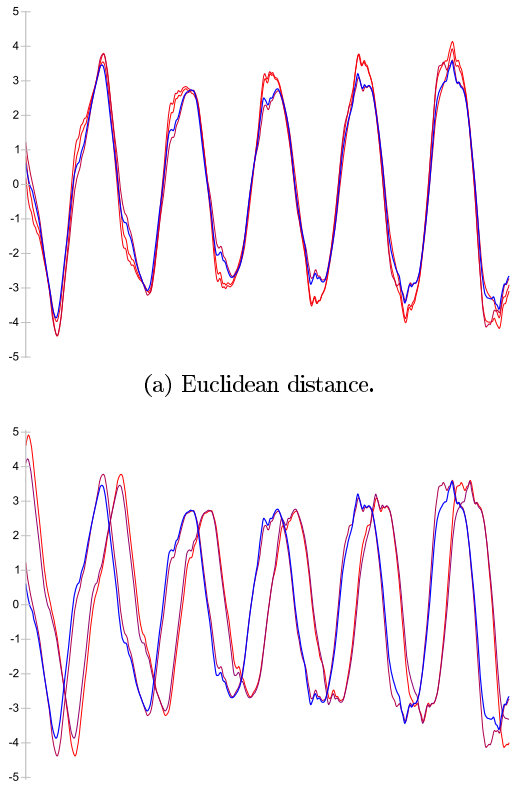
Distance functions are used to perform range queries on a database object. Any implementation of an algorithm can rely on the database object to perform range queries with an arbitrary distance function and needs only to ask for k nearest neighbors not being concerned with the details of data handling.

A new data type is supposed to implement the interface `DatabaseObject`. A new algorithm class suitable to certain data types `O` needs to implement the Interface `Algorithm<O extends DatabaseObject>`. The central routine to implement the algorithmic behavior is `Result run(Database<O> database)`. Here, the algorithm is applied to an arbitrary database consisting of objects of a suitable data type. The database supports operations like

```
<D extends Distance<D>> List<DistanceResultPair<D>> kNNQueryForObject(
    O queryObject, int k, DistanceFunction<O,D> distanceFunction)
```

performing a k-nearest neighbor query for a given object of a suitable data type `O` using a distance function that is suitable for this data type `O` and provides a distance of a certain type `D`. Such a query method returns a list of `DistanceResultPair<D>` objects encapsulating the database IDs of the collected objects and their distance to the query object in terms of the specified distance function. A `Distance` object (here of the type `D`) in most cases just encapsulates a `double` value but could also be a more complex type, e.g. consisting of a pair of values as often used in subspace or correlation clustering algorithms like DiSH [9] or ERiC [8]. This list is sorted in ascending order w.r.t. the distance from the query object. As such, this method or related methods for epsilon-range queries are not only used in any clustering algorithm but also for comparing different distance measures. In ELKI 0.2, the performance of different distance measures can be directly assessed and visualized to enable the researcher to get a feeling for the meaning, benefits and drawbacks of a specific distance measure.

For that purpose, in a data set of time series, a specific time series can be picked and a k-NN query can be performed for this time series within the data



(a) Euclidean distance.

(b) LCSS distance.

Fig. 1. ELKI 0.2: Visualization of k-NN query results for different distance measures.

set for any k and any distance function. The result of the query is e.g. visualized by assigning colors of degrading similarity to the time series in the query result according to the decreasing similarity w.r.t. the given distance measure. An example is shown in Figure 1: the query time series (blue) and its 3 nearest neighbors (color blending from blue to red with increasing distance). In this case, the different behavior of Euclidean distance (Figure 1(a)) and LCSS distance (Figure 1(b)) is demonstrated.

For time series data, in ELKI 0.2 especially the following exemplary distance measures are incorporated: the *Dynamic Time Warping* (*DTW*) distance [10], the *Longest Common Subsequence* (*LCSS*) distance [11], the *Edit Distance on Real sequence* (*EDR*) [12] and the *Edit distance with Real Penalty* (*ERP*) [13]. Any clustering or classification algorithm may therefore use a specialized distance function and implement a certain routine using this distance function on an arbitrary database.

2.3 Availability and Documentation

Via `http://www.dbs.ifi.lmu.de/research/KDD/ELKI/` the framework ELKI, documentation of the implementation and usage as well as examples to illustrate how to expand the framework by integrating new algorithms are available.

3 Conclusion

The software system ELKI presents a large collection of data mining algorithms which can be supported by arbitrary index structures and work on arbitrary data types given supporting data classes and distance functions. ELKI 0.2 is also able to visualize the behavior and the possibly different partialities of different distance measures for time series data. We therefore expect ELKI 0.2 to facilitate broad experimental evaluations of algorithms and distance measures – existing and newly developed ones alike.

References

1. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. 2nd edn. Morgan Kaufmann (2005)
2. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: YALE: Rapid prototyping for complex data mining tasks. In: Proc. KDD. (2006)
3. Müller, E., Assent, I., Günnemann, S., Jansen, T., Seidl, T.: OpenSubspace: an open source framework for evaluation and exploration of subspace clustering algorithms in WEKA. In: Proc. OSDM@PAKDD. (2009)
4. Hellerstein, J.M., Naughton, J.F., Pfeffer, A.: Generalized search trees for database systems. In: Proc. VLDB. (1995)
5. Aßfalg, J., Kriegel, H.P., Kröger, P., Kunath, P., Pryakhin, A., Renz, M.: T-Time: threshold-baed data mining on time series. In: Proc. ICDE. (2008)
6. Achtert, E., Kriegel, H.P., Zimek, A.: ELKI: a software system for evaluation of subspace clustering algorithms. In: Proc. SSDBM. (2008)
7. Achtert, E., Böhm, C., Kriegel, H.P., Kröger, P., Zimek, A.: Robust, complete, and efficient correlation clustering. In: Proc. SDM. (2007)
8. Achtert, E., Böhm, C., Kriegel, H.P., Kröger, P., Zimek, A.: On exploring complex relationships of correlation clusters. In: Proc. SSDBM. (2007)
9. Achtert, E., Böhm, C., Kriegel, H.P., Kröger, P., Müller-Gorman, I., Zimek, A.: Detection and visualization of subspace cluster hierarchies. In: Proc. DASFAA. (2007)
10. Berndt, D., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD Workshop. (1994)
11. Vlachos, M., Kollios, G., Gunopulos, D.: Discovering similar multidimensional trajectories. In: Proc. ICDE. (2002)
12. Chen, L., Özsu, M., Oria, V.: Robust and fast similarity search for moving object trajectories. In: Proc. SIGMOD. (2005)
13. Chen, L., Ng, R.: On the marriage of Lp-norms and edit distance. In: Proc. VLDB. (2004)