Data Min Knowl Disc (2007) 15:87–97
DOI 10.1007/s10618-007-0067-9

Future trends in data mining

Hans-Peter Kriegel · Karsten M. Borgwardt · Peer Kröger · Alexey Pryakhin · Matthias Schubert · Arthur Zimek

Received: 12 May 2006 / Accepted: 7 February 2007 / Published online: 23 March 2007

Abstract Over recent years data mining has been establishing itself as one of the major disciplines in computer science with growing industrial impact. Undoubtedly, research in data mining will continue and even increase over coming decades. In this article, we sketch our vision of the future of data mining. Starting from the classic definition of "data mining", we elaborate on topics that — in our opinion — will set trends in data mining.

Keywords Data Mining · Knowledge Discovery · Future trends

1 The classic definition

With the advent of high-throughput experimental technologies and of high-speed internet connections, generation and transmission of large volumes of data has been automated over the last decade. As a result, science, industry, and even individuals have to face the challenge of dealing with large datasets which are too big for manual analysis. While these large "mountains" of data are easily produced nowadays, it remains difficult to automatically "mine" for valuable information within them.

"Data Mining", often also referred to as "Knowledge Discovery in Databases" (KDD), is a young sub-discipline of computer science aiming at the automatic interpretation of large datasets. The classic definition of knowledge

Responsible editor: Geoffrey Webb.

H.-P. Kriegel (✉) · K. M. Borgwardt · P. Kröger · A. Pryakhin · M. Schubert · A. Zimek
Ludwig-Maximilians-Universität,
Oettingenstr. 67, Munich 80538, Germany
e-mail: kriegel@dbs.ifi.lmu.de

discovery by Fayyad et al. from 1996 describes KDD as "the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" (Fayyad et al. 1996). Additionally, they define data mining as "a step in the KDD process consisting of applying data analysis and discovery algorithms[...]".[1] Over the last decade, a wealth of research articles on new data mining techniques has been published, and the field keeps on growing, both in industry and in academia.

In this article, we want to present our vision of future trends in data mining and knowledge discovery. Interestingly, the four main topics we anticipate are indirectly described in the classic definition of KDD.

Fayyad et al. define KDD as searching for "patterns in data". Originally, these data were exclusively feature vectors, and they were static, i.e. without temporal evolution. Over recent years, structured data such as strings and graphs and temporal data such as data streams and time series have moved into the focus of data mining, yet a lot remains to be done. We sketch our vision of future developments in the field of mining complex objects in Sect. 2 and of temporal data mining in Sect. 3.

In order to be able to "identify[ing] valid, novel [...] patterns in data", a step of preprocessing of the data is almost always required. This preprocessing has a significant impact on the runtime and on the results of the subsequent data mining algorithm. We discuss potential future progress in the understanding and improvement of data preprocessing in Sect. 4.

Finally, knowledge discovery should detect "potentially useful, and ultimately understandable patterns". While important steps towards finding patterns have been taken, non-experts may still encounter lots of difficulties, both in applying data mining algorithms and interpreting their results. Advances in making data mining algorithms more convenient in use and their results easier to understand will have a positive impact on the data mining community. Sect. 5 presents some considerations in this respect.

2 Mining complex objects of arbitrary type

Modern automated methods for measurement, collection, and analysis of data in all fields of science, industry, and economy are providing more and more data with drastically increasing complexity of its structure. This growing complexity is justified on the one hand by the need for a richer and more precise description of real-world objects, and on the other hand by the rapid progress in measurement and analysis techniques allowing versatile exploration of objects. In order to manage the huge volume of such complex data, database systems are employed. Thus, databases provide and manage manifold information concerning all kinds of real-world objects, ranging from customers and molecules to shares and patients.

[1] Note that this definition is opposite to the common habit of using knowledge discovery as a synonym for data mining.

Traditionally, relational databases keep this information in the form of attributes from a certain range of possible domains, usually as numbers, dates, or strings, or, possibly, restricted to a certain list of values. Object-relational databases even allow one to define types to model arbitrary objects. In view of the fact that the manual analysis of enormous volumes of complex data collected in a database is infeasible in practice, there is an ever growing need for data mining techniques that are able to discover novel, interesting knowledge in this complex and voluminous data. Various methods for a wide range of complex data types have been proposed over recent years, such as mining Multi-Instance Objects (Dietterich et al. 1997; Gärtner et al. 2002; Weidmann et al. 2003; Kriegel et al. 2006), or mining Multi-Represented Objects in a supervised (Kittler et al. 1998; Kriegel et al. 2004, 2005) or unsupervised manner (Yarowsky 1995; Blum and Mitchell 1998; Kailing et al. 2004), or such as graph mining (Washio and Motoda 2003).

However, data mining approaches usually tackle certain subtypes of data. For example, item set mining is specialized to string data or lists of possible values, many classification or clustering approaches need numerical data, whereas others allow mining of categorical data. Often the different approaches are combined to yield more appropriate results. For example, certain classes of string kernels assess frequent substrings of sequences or texts and basically count their occurrence (which resembles item set mining of some sort) to finally build numerical features (comprising the number of occurrences of a certain set of substrings). This in turn allows the application of methods engineered for numerical data.

Modeling the world obviously creates a merely simplified representation. Considering the real complexity of the objects as adequately as possible remains a worthwhile goal for all directions of science. In computer science, the concept of "object-oriented modeling" intends to describe complex objects in a simple and thoroughly formalized manner. Here, attributes of an object may be primitive types or objects themselves. Object-oriented and also object-relational databases are able to present collections of such objects. It seems highly desirable to be able to directly mine on these objects instead of mining only parts of them (like their numerical attributes or numerical models of their complex attributes). In recent years, many steps were taken to mine objects modeled as graphs, or multi-represented, multi-relational or multi-instance data. In some respects, these approaches are generalizations of former approaches on unstructured data. On the other hand, the very same approaches could be understood as adjustments to certain more general, but not universal types of representations. We envision data mining being universalized to tackle truly general objects. However, all these methods consider static properties of objects. The picture of "object-oriented modeling" does also include a modeling of behavior of objects, called "methods", i.e. dynamic properties. Furthermore, sequence diagrams or activity diagrams model the chronology of behavior patterns. We consider these temporal aspects below (cf. Sect. 3). Indeed, the behavior of software is a common data mining task (cf. e.g.

(Liu et al. 2005, 2006b)). Some steps towards directly mining object-oriented systems can be found e.g. in Kanellopoulos et al. (2006).

Representing complex objects by means of simple objects like numerical feature vectors could be understood as a way to incorporate domain knowledge into the data mining process. The domain expert seeks ways to use the important features of an object to e.g. classify new objects of the same type, eventually by employing sophisticated functions to transform attributes of some type to features of some other type. In the progress to generalized data mining one should not disregard of course the advances made so far. Incorporating domain knowledge fundamentally facilitates meaningful data mining. However, the specific way to make use of the domain knowledge of experts should also be generalized to keep pace with more complex ways of mining complex objects.

Furthermore, the knowledge specific to a certain domain is increasing in amount and complexity itself. Thus, it usually cannot be surveyed by a single human expert anymore — the communities therefore in turn begin to provide their knowledge often in databases or knowledge bases. Thus, in the future, data mining algorithms should be able to automatically take reliable domain knowledge available in databases into account in order to improve their effectiveness.

In order to process complex objects, distributed data mining seems to become increasingly important (Liu et al. 2006a). Several application domains consider the same complex object according to the same characteristics at different locations and/or at different times (e.g. a patient can consult different doctors, or continuous observation of a star is only possible by involving several telescopes around the world). On the other hand, data mining on complex objects requires significantly more computational power than data mining on feature vectors. Finally, not all participants in a joint activity of data mining would like to share all of their collected data, possibly in order to protect the privacy of their customers. Thus, there is a growing need for distributed, privacy preserving data mining algorithms for complex data.

3 Temporal aspects: dynamics and relationships

As indicated above, knowledge about the behavior of objects is an integral part of understanding complex relationships in real-world systems and applications. More and more modern methods for observation and data generation provide suitable data to capture such complex relationships. Nonetheless, — sometimes due to historical reasons — many research directions in data mining focus only on static descriptions of objects or are not designed to take data with dynamic behavior and/or relationships into account. Obviously, this is a severe limitation since several important aspects of objects that are urgently needed to get a better understanding of complex relationships are thus not considered for data mining.

In addition to more complex data models, relationships between objects are also determined by temporal aspects hidden in the data. Among others, there

are two major challenges derived from these temporal aspects which future data mining approaches should be able to cope with.

First, data can describe developments over time or temporal mechanisms. Typical examples are data streams and time-series data. Although mining these temporal data types has received increasing attention in the past years, (see e.g. Gaber et al. 2005; Keogh and Kasetty 2002 for overviews) there are still many challenges to be addressed. For example, future data mining algorithms will have to cope with finding different types of correlations in high-dimensional time series or should explore novel types of similarity models for temporal data in order to address different practical problems. Thus, following a more application-oriented approach can offer novel challenges to the data mining community.

Secondly, the patterns that are observed have also a temporal aspect, i.e. these patterns usually evolve over time in dynamic scenarios. An important challenge is to keep the patterns up-to-date without a complete recalculation from scratch. In general, this is especially needed for mining data streams. However, also in a dynamic database environment (which is usually the realistic scenario in most companies) where inputs, deletions, and updates occur frequently, keeping patterns up-to-date is a challenging problem of great practical importance (Achtert et al. 2005; Domeniconi and Gunopulos 2001). In addition, it is very interesting for many applications to monitor the evolution of patterns and to derive knowledge concerning these changes or even the complete dynamic behavior of patterns. Finding "patterns of evolving patterns" is an important challenge which has not attracted a lot of research yet. Thus, data mining approaches for these temporal aspects are envisioned, since they will play a key role in the process of understanding complex relationships and behavior of objects and systems.

4 Fast, transparent and structured data preprocessing

Anyone who has performed data mining on a real-world dataset agrees that knowledge discovery is more than pure pattern recognition: Data miners do not simply analyze data, they have to bring the data in a format and state that allows for this analysis. It has been estimated that the actual mining of data only makes up 10% of the time required for the complete knowledge discovery process (Pyle 1999). In our opinion, the precedent time-consuming step of preprocessing is of essential importance for data mining (Han and Kamber 2001). It is more than a tedious necessity: The techniques used in the preprocessing step can deeply influence the results of the following step, the actual application of a data mining algorithm. We therefore feel that the role of the impact on and the link of data preprocessing to data mining will gain steadily more interest over the coming years.

A very nice example to support this claim originates from the field of data mining in microarray gene expression data. As microarrays often miss to produce data for a considerable amount of genes, one has to impute these missing

values for the following step of data analysis. Depending on which algorithm is chosen for missing value estimation, the data mining results vary significantly, as repeatedly reported in the microarray community (Troyanskaya et al. 2001; Bø 2004; Jörnsten et al. 2005). This impact of data preprocessing on data mining results is similarly reported in completely different application domains such as operations research (Cronea et al. 2005).

In addition to format and completeness of the data, data mining algorithms generally implicitly require data to originate from one single source. Entries of different databases, however, may have different scales and may have been generated by different experimental techniques with varying degree of noise. Before data analysis starts, these differences between data from different sources have to be balanced via data integration. Otherwise one risks discovering patterns within the data that are caused by their different origins, and not by phenomena in the application domain one wants to study. In addition to this *statistical* integration, different formats and different semantics in disparate data sources require further efforts in format and semantic integration, which form long-standing challenges for the database community (Halevy 2003). Hence, data integration is another central step in data preprocessing for knowledge discovery.

It is important to point out that data preprocessing faces problems similar to those of data mining. High-dimensional data can lead to scalability problems for preprocessing algorithms, and missing value imputation and data integration on structured data such as strings and graphs are even theoretically challenging problems. Especially the statistical data mining community will be challenged to design statistical tests and algorithms for efficient and scalable data preprocessing on high-dimensional and structured data.

What will the future of data preprocessing for data mining look like? We envision that preprocessing will become more powerful, faster and more transparent than it is today. For fast and user-friendly data mining applications, data preprocessing will be automated, and all steps undertaken will be reported to the user or even interactively controlled by the user. A common data representation and a common description language for data preprocessing will make it easier for both computer and data miner to study and to decide which preprocessing steps have been applied or should be applied to these data. Advanced systems will automatically perform preprocessing in several different fashions and report the results — and the differences between results of different preprocessing techniques — to the user. Novel statistical tests and preprocessing algorithms will enable the efficient preprocessing of large-scale, high-dimensional and structured data. From a theoretical point of view, general classes of preprocessing algorithms will likely be defined, such that the multitude of existing techniques can be regarded as special cases of these broader categories. In any case, there will be a lot to gain and a lot to study in preprocessing for data mining over the next years.

5 Increasing usability

An ultimate trend can be subsumed under the slogan "increased usability". Sections 2 and 3 have highlighted a growing demand for algorithms and systems that can cope with more and more complex data objects which are structured and observed over a certain period of time. Though future algorithms might handle this complexity, the need for user guidance during preprocessing and data mining will dramatically increase. Even in current data mining algorithms, many established methods employ quite a few different input parameters. For example, weighted Euclidian distance is very popular to tune distance-based data mining algorithms, and edit distance (Bille 2005) is rather popular for comparing trees, sequences and graphs. Though these methods often prove to be very useful, it is still necessary to find out a good parameter setting before exploiting the gained flexibility. In a wider sense, selecting a data mining algorithm and a data transformation method itself can also be considered as a problem of user guidance. The extensive use of parameters leaves data miners with a large choice of algorithms and allows them to squeeze out that little bit of extra performance by spending additional time on parameter tuning. However, the gained flexibility comes at a price. Selecting the best possible methods and finding a reasonable parameter setting are often very time consuming. For future data mining solutions, this problem will become more dramatic because more complex objects usually mean more parameters. Furthermore, many approaches will employ several steps of data mining and each of them will have its own parameters. For example, set-valued objects can be compared by multiple kernels and distance measures (Eiter and Mannila 1997; Ramon and Bruynooghe 2001; Gärtner et al. 2002) which compare the elements of each set by using another kernel or distance measure in the feature space of single instances. Therefore, data mining algorithms having a very small number of parameters will gain more and more importance in order to reduce the necessary user interaction. Other related aspects of usability are the intuitiveness when adjusting the parameters and the parameter sensitivity. If the results are not strongly dependent on slight variations of the parametrization, adjusting the algorithms becomes less complex. To fulfill these requirements, we will first of all distinguish two types of parameters and afterwards propose four goals for future data mining methods.

The first type of parameter, called *type I*, is tuning data mining algorithms for deriving useful patterns. For example, k for a k-NN classifier influences directly the achieved classification accuracy and thus, the quality of classification.

The second type of parameter, called *type II*, is more or less describing the semantics of the given objects. For example, the cost matrix used by edit distance (Bille 2005) has to be based on domain knowledge and thus, varies from application to application. The important aspect of this type of parameter is that the parameters are used to model additional constraints from the real world.

Based on these considerations, the following proposals for future data mining solutions can be formulated:

1. Avoid type I parameters if possible when designing algorithms.
2. If type I parameters are necessary, try to find the optimal parameter settings automatically. For many data mining algorithms, it might be possible to integrate the given parameters into the underlying optimization problem.
3. Instead of finding patterns for one possible value of a type II parameter, try to simultaneously derive patterns for each parameter setting and store them for postprocessing. Having all patterns for all possible parameter settings, will allow one to gain important information. For example, if we want to check if a certain pattern can be observed in a given data set and we do not know which parameter setting to use, we could derive patterns using all possible parameter values. Afterwards, we can check whether the pattern was observed for one of the settings at all. It might be impossible to distinguish meaningful from meaningless parameter settings, but for the same problem, it might be possible to judge the quality of the results.
4. Develop user friendly methods to integrate domain knowledge where it is necessary. Often the only applicable approach for selecting type II parameters is to include additional domain knowledge into the data mining task. However, transforming expert knowledge of a domain expert into a parameter value is often quite difficult since the domain expert might not know the meaning of the parameter.

In summary, future data mining applications will be capable to tune themselves as far as possible, help domain experts to integrate their knowledge into data transformation and generate a variety of possible patterns.

The second important aspect of increasing usability deals with the derived patterns themselves. Currently, most of the data mining algorithms generate patterns that can be defined in a mathematical sense. However, methods for explaining the meaning of the found patterns are still in a minority. In the light of the increased object complexity, this problem will gain additional importance. Though it might be possible to interpret the meaning of a surface in a given vector space, the patterns derived for more complex objects might not be interpreted that easily even by an expert. Thus, it is likely that not only the input data for data mining is getting more complex, but also the gained patterns will increase in complexity. This trend is amplified by another challenge that recently came up in the data mining community. In many applications, the very general patterns derived by the standard methods do not yield a satisfying solution to the given task. In order to solve a problem, the found patterns need to fulfill a certain set of constraints which make them more interesting for the application. Examples for this type of patterns are correlation clusters (Böhm et al. 2004) and constrained association rules (Srikant et al. 1997). For more complex data with mutual relationships, the derived patterns will be even more complex. Thus, we can formulate additional challenges:

1. The patterns described by the data mining algorithms are still too abstract for being understood. However, a pattern that is misinterpreted is of great danger. For example, many data mining algorithms do not distinguish between causality and co-occurrence. Consider an application that aims at

finding the reason for a certain type of disease. There is a great difference between finding the origin of the disease or finding just an additional symptom. Therefore, a very old challenge will remain very important for the data mining community: Developing systems which derive understandable patterns and making already derived patterns understandable.
2. As stated above, current algorithms mostly focus on a limited set of standard patterns. However, deriving these patterns often does not yield a direct and complete solution to many problems where data mining could be very useful. Furthermore, with an increasing complexity of the analyzed data, it is likely that the derived patterns will increase in complexity as well. Thus, a future trend in data mining will be to find richer patterns.
3. A final task when working with future patterns is the increased number of valid patterns, we might find in a large data set of complex objects, e.g. through trying out several parameter values at once as mentioned above. Therefore, the number of potentially valid patterns will be too large to be handled by a human user, without a system organizing the results. Thus, future systems must provide a platform for pattern exploration where users can browse for knowledge they might consider as interesting.

To conclude, future data mining should generate a large variety of well understandable patterns. Due to variations in the parameterizations, the number of possibly meaningful and useful patterns will dramatically increase and thus, an important aspect is managing and visualizing these patterns.

6 Conclusions

In this article, we surveyed major challenges for data mining in the years ahead. We started with the classic definition of knowledge discovery and data mining. Although we believe that this definition still describes the essence of this important area of computer science, its interpretation has broadened over the last couple of years and will continue to do so in the future. We highlighted what is our vision of future interpretations of this definition.

First, we started with the type of "patterns in data" which knowledge discovery is examining. While original data mining concentrated on vectorial data, future data will predominantly be stored in much more complex data types and data mining will have to cope with this increasing volume of structured data. Another aspect of "patterns in data" in the future is the increasing importance of studying their evolution over time. Considering time, allows to observe the dynamics of patterns as well as the behavior and the interactions of data objects.

Second, the data to be studied is usually drawn from several sources. For this reason, another important trend in data mining will be the growing importance of data preprocessing and integration, ensuring that the "patterns in data" found are "valid" on the complete set of data objects and not just on a particular subset.

Third, an ultimate trend that data mining faces is increased usability to detect "understandable patterns", and to make data mining methods more

user-friendly. If future data mining methods have to handle all this complex input and intelligent preprocessing, it is very likely that the user will have to adjust more and more switches and knobs before getting any result. Hence, achieving user-friendliness with transparent or even reduced parameterization is a major goal. Usability is also enhanced by finding new types of patterns that are easy to interpret, even if the input data is very complex.

Although no human being can foretell the future, we believe that there are plenty of interesting new challenges ahead of us, and quite a few of them cannot be foreseen at the current point of time.

References

Achtert E, Böhm C, Kriegel H-P, Kröger P (2005) Online hierarchical clustering in a data warehouse environment. In: Proceedings of the 5th international conference on data mining (ICDM), Houston, TX, pp 10–17

Bille P (2005) A survey on tree edit distance and related problems. Theor Comput Sci 337(1–3):217–239

Blum A, Mitchell T (1998) Combining labeled and unlabeled data with Co-training. In: Proceedings of the 11th annual conference on computational learning theory (COLT), Madison, WI, pp 92–100

Bø TH, Dysvik B, Jonassen I (2004) LSimpute: accurate estimation of missing values in microarray data with least squares methods. Nucleic Acids Res 32(3)

Böhm C, Kailing K, Kröger P, Zimek A (2004) Computing clusters of correlation connected objects. In: Proceedings of the SIGMOD conference, Paris, France, pp 455–466

Cronea SF, Lessmann S, Stahlbock R (2005) The impact of preprocessing on data mining: an evaluation of classifier sensitivity in direct marketing. Eur J Oper Res

Dietterich TG, Lathrop RH, Lozano-Perez T (1997) Solving the multiple instance problem with axis-parallel rectangles. Artif Intell 89:31–71

Domeniconi C, Gunopulos D (2001) Incremental support vector machine construction. In: Proceedings of the 1st international conference on data mining (ICDM), San Jose, CA, pp 589–592

Eiter T, Mannila H (1997) Distance measures for point sets and their computation. Acta Informatica 34(2):103–133

Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) Knowledge discovery and data mining: Towards a unifying framework. In: Proceedings of the 2nd ACM international conference on knowledge discovery and data mining (KDD), Portland, OR, pp 82–88

Gaber MM, Zaslavsky A, Krishnaswamy S (2005) Mining data streams: a review. SIGMOD Records 34(2)

Gärtner T, Flach PA, Kowalczyk A, Smola A (2002) Multi-instance kernels. In: Proceedings of the 19th international conference on machine learning (ICML), Sydney, Australia, pp 179–186

Halevy AY (2003) Data integration: a status report. In: BTW, pp 24–29

Han J, Kamber M (2001) Data mining: concepts and techniques. Academic Press, San Diego

Jörnsten R, Wang H-Y, Welsh WJ, Ouyang M (2005) DNA microarray data imputation and significance analysis of differential expression. Bioinformatics 21(22):4155–4161

Kailing K, Kriegel H-P, Pryakhin A, Schubert M (2004) Clustering multi-represented objects with noise. In: Proceedings of the 8th pacific-asia conference on knowledge discovery and data mining (PAKDD), Sydney, Australia, pp 394–403

Kanellopoulos Y, Dimopulos T, Tjortjis C, Makris C (2006) Mining source code elements for comprehending object-oriented systems and evaluating their maintainability. SIGKDD Explorations 8(1):33–40

Keogh E, Kasetty S (2002) On the need for time series data mining benchmarks: A survey and empirical demonstration. In: Proceedings of the 8th ACM international conference on knowledge discovery and data mining (SIGKDD), Edmonton, Alberta, pp 102–111

Kittler J, Hatef M, Duin R, Matas J (1998) On combining classifiers. IEEE Trans Pattern Analysis and Machine Intelligence 20(3):226–239

Kriegel H-P, Kröger P, Pryakhin A, Schubert M (2004) Using support vector machines for classifying large sets of multi-represented objects. In: Proceedings of the 4th SIAM international conference on data mining (SDM), Orlando, FL, pp 102–113

Kriegel H-P, Pryakhin A, Schubert M (2005) Multi-represented kNN-classification for large class sets. In: Proceedings of the 10th international conference on database systems for advanced applications (DASFAA), Beijing, China, pp 511–522

Kriegel H-P, Pryakhin A, Schubert M (2006) An EM-approach for clustering multi-instance objects. In: Proceedings of the 10th pacific-asia conference on knowledge discovery and data mining (PAKDD), Singapore, pp 139–148

Liu C, Yan X, Yu H, Han J, Yu PS (2005) Mining behaviour graphs for "backtrace" of noncrashing bugs. In: Proceedings of the 5th SIAM international conference on data mining (SDM), Newport Beach, CA, pp 286–297

Liu K, Kargupta H, Bhaduri K, Ryan J (2006a) Distributed data mining bibliography, January 2006. http://www.csee.umbc.edu/ hillol/DDMBIB/

Liu C, Yan X, Han J (2006) Mining control flow abnormality for logic error isolation. In: Proceedings of the 6th SIAM international conference on data mining (SDM), Bethesda, MD, pp 106–117

Pyle D (1999) Data preparation for data mining. Morgan Kaufmann Publishers Inc.

Ramon J, Bruynooghe M (2001) A polynomial time computable metric between points sets. Acta Informatica 37:765–780

Srikant R, Vu Q, Agrawal R (1997) Mining association rules with item constraints. In: Proceedings of the 3rd ACM international conference on knowledge discovery and data mining (KDD), Newport Beach, CA, pp 67–73

Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman RB (2001) Missing value estimation methods for DNA microarrays. Bioinformatics 17(6):520–525

Weidmann N, Frank E, Pfahringer B (2003) A two-level learning method for generalized multi-instance problems. In: Proceedings of the 14th european conference on machine learning (ECML), Cavtat-Dubrovnik, Croatia, pp 468–479

Washio T, Motoda H (2003) State of the art of graph-based data mining. SIGKDD Explorations Newslett 5(1):59–68

Yarowsky D (1995) Unsupervised word sense disambiguation rivaling supervised methods. In: Meeting of the association for computational linguistics