Contextual Domain Knowledge for Incorporation in Data Mining Systems

Alex G. Büchner[*], John G. Hughes[*] and David A. Bell[†]

[*] Northern Ireland Knowledge Engineering Laboratory, University of Ulster
[†] School of Information and Software Engineering, University of Ulster
Shore Road, Newtownabbey, BT37 0QB, UK
email: {ag.buchner, jg.hughes, da.bell}@ulst.ac.uk

Abstract

The concept of contextual domain knowledge is proposed and incorporated in a generic data mining architecture. Various types of domain knowledge (taxonomies, constraints, previously discovered knowledge and user preferences) are allotted with context information, which is organised in a hierarchical topology. Mediation is defined in order to chose context-dependent domain knowledge for incorporation in a data mining exercise. All components are embedded in a contextual knowledge discovery architecture.

Introduction

Data and domain knowledge are the two most essential input ingredients for data mining applications. The former is either provided by operational databases or by their warehoused counterparts in form of materialised views, which can be reused for multiple model building exercises (Büchner, Hughes, and Bell, 1999). The latter, in whatever form provided, has to be re-specified, or at least modified, depending in which context patterns are to be discovered. This limits the concept of domain knowledge, and thus, the objective of this paper is to propose the notion of contextual domain knowledge, which can be reused across multiple related knowledge discovery exercises.

The structure of this paper is as follows. First, a domain knowledge classification is provided, which distinguishes between taxonomies, constraints, previously discovered knowledge, and user preferences. Then, contexts are specified and organised in a hierarchical topology. The section that follows, combines the two components to contextual domain knowledge, which is based on context mediation. In order to benefit from the newly introduced concepts, contextual domain knowledge is incorporated in a data mining framework. Finally, conclusions are drawn and further research is outlined. Formal specifications and definitions are supported by examples from an electronic commerce scenario in which marketing intelligence has been discovered from Internet log files (Büchner and Mulvenna 1998).

Domain Knowledge Classification

Domain knowledge can be utilised in a number of ways. It can be used for making patterns more visible, for constraining the search space, for discovering more accurate knowledge, and for filtering out uninteresting knowledge (Anand and Büchner 1998). There have been numerous classifications of domain knowledge for data mining purposes. A comprehensive taxonomy has been presented by Klösgen and Żytkow (1999), which is used for this paper in order to discuss context-related issues. The set of chosen domain knowledge types is by no ends exhaustive. However, it embodies a representative collection that has proven sufficient for most data mining applications.

Taxonomies

Taxonomies provide classifications, which allow the grouping of many attribute values into a smaller number thereof. The three most typical types of taxonomical domain knowledge are bandings, concept hierarchies and networks.

Definition 1. A **banding** b is defined as $b = [b_{min}, b_{max}]$, where b_{min} and b_{max} represent the lower and upper limit of a range, respectively.

Typical examples of bandings are customer age groups, login time ranges, and marketing seasons. The outcome of a banding operation is $b_{min} \times b_{max} \rightarrow \tau$, where τ represents a linguistic term.

Definition 2. A **concept hierarchy** h is a connected, undirected, acyclic graph, which is defined as the tuple $h = (L, E)$, where $L = \{l_0, l_{1_1}, l_{1_2}, l_{2_1}, l_{2_2}, \ldots\}$ and $E = \{e_1, e_2, e_3, \ldots\}$. Each e has the form $e = <l_i, l_j>$; $l_i, l_j \in L$, l_0 has indegree 0, $l_1 \ldots l_n$ have indegree 1. l_i is subconcept of l_j iff $l_i \subset l_j$; l_i is superconcept of l_j iff $l_j \subset l_I$

Multi-level concept hierarchies can, for instance, represent Internet domain names, customer post codes, or product ranges. Concept hierarchies also cover groupings, which are represented as a tree with a single root and all leaves being connected directly to the top-level node.

Definition 3. A **network** *w* is a directed, connected, cyclic graph, which is defined as the tuple $w = (N, E)$, where $N = \{n_1, n_2, n_3, \ldots\}$ and $E = \{e_1, e_2, e_3, \ldots\}$, each *n* representing a node in *w*. Each *e* has the form $e = <n_i, n_j>$; $n_i, n_j \in N$.

A typical network in an electronic commerce scenario is the topology of a retail web site, which provides information about the links among different pages and page clusters.

Constraints

Constraints represent limitations to data values, which are domain as well as discovery dependent. Constraints can either exclude or include certain data values. Furthermore, constraints are usually specified in form of attribute-relationship rules, or can be transformed to such (Anand et al. 1995).

Definition 4. A **constraint** *l* is specified as a rule such that l: *antecedent* → *consequent*.

Both, the antecedent as well as the consequent can have multiple values, which allows the flexible description of constraints.

An example rule in the electronic commerce scenario mentioned above states that the profession of the user of an URL that is not a proxy and has the top level domain *.edu* is a lecturer, student or researcher.

Previously discovered Knowledge

Previously discovered knowledge can be reused as domain knowledge if it is in first-order normal form. Depending on the type of pattern discovery method that has been applied (rule induction, neural networks, Bayesian belief networks, et cetera), the knowledge can theoretically be in any form as defined above or in any pattern discovery-dependent format.

Typical patterns being reused usually have a very high degree of support and / or confidence (also known as quasi facts), which indicates their trueness across data mining and other reasoning exercises.

User Preferences

User preferences specify upper or lower limits for certain discovery measures in the form of thresholds, for example, support, confidence, deviation, sequence length, and so forth. Silberschatz and Tuzhilin (1996) have distinguished between interestingness (sub-divided into actionability and unexpectedness) and (soft as well as hard) beliefs, which depend on the pattern discovery technique being used. Without going into great detail, all these measures can be classified as subjective or objective. This classification is used for investigating domain knowledge in context, since it represents adequately the individualistic nature of humans, being involved in a knowledge discovery process (see 'Domain Knowledge in Context' Section).

Context Organisation

In a knowledge discovery environment, a context represents behavioural aspects which are shared by attributes of the same ontology. Assuming an underlying electronic commerce ontology, possible properties are a top-level domain's location, the exchange rate of a currency, or the login time zone offset.

Definition 5. A **context** *c* contains a set of properties $P_c = \{p_{c_1}, p_{c_2}, p_{c_3}, \ldots\}$ where $P_c \subseteq P$ and $P = \{p_1, p_2, p_3, \ldots\}$, which represents an application-specific ontology *O*.

The original idea of the context identity concept is that every single attribute instance is being allotted an additional attribute context identifier in a (multi-) database scenario, where each attribute instance is represented by a semantic value (Büchner, Bell, and Hughes 1998). The same concept is applied to domain knowledge so that the same context information can be used for both, data as well as domain knowledge (Büchner, Hughes, and Bell 1999).

To minimise the redundancy of context declarations (for instance, related sub-level domains, or dates in different time zones but from a single calendaric system), contexts are organised hierarchically. Thus, structure and behaviour can be inherited, and overloading and overriding mechanisms can be applied to contexts. The whole spanning context tree is representing the underlying ontology and is represented as a **context hierarchy** *O*, which is an undirected, connected, acyclic graph (see Definition 2). A further advantage of the hierarchical arrangement of contexts is the possibility of packaging context sub-trees; we refer to these sub-trees as resources *r* such that $r \subset o$, that is $\forall r_i \in r \{r_i \in o\}$ and $r \neq o$.

For handling contextual databases, further constructs have been defined, which encompass contextualised equivalence specifications for atomic and complex types, distance measures, as well as context mediation. These notions have been embedded in the ODMG object data model and an object definition language as well as its query counterpart have been proposed (Büchner, Bell, and Hughes 1998). However, for the purpose of connecting context to domain knowledge these operations are not required. The linkage between contexts and the defined types of domain knowledge is performed through the allotment of context identifiers to each instance of domain knowledge.

Domain Knowledge in Context

As described earlier, there exists a multitude of domain knowledge types, with respect to structure, content and reusability. For the purpose of this paper, each domain knowledge type is investigated from two different dimensions. The first is concerned with the degree of reality, where reality is represented in a spectrum from a physical world to a logical model world (see Figure 1). The second is interested in the degree of reusability of the specified types of domain knowledge.

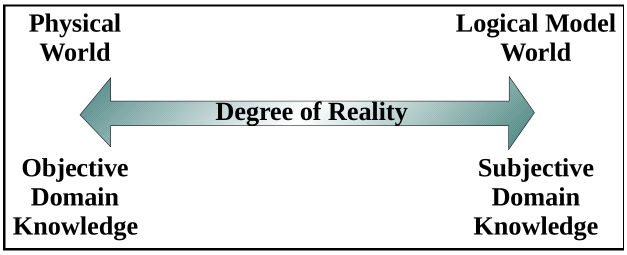

Figure 1. Domain Knowledge Degrees of Reality

Objective and Subjective Domain Knowledge

Objective domain knowledge consists of a set of quasi facts of the domain a data mining exercise is performed in. Although it can have a certain degree of context-dependency, it is almost always kept as holos and only exchanged in total for its contextual counterpart.

Examples of objective domain knowledge are the topology of the e-retailer's web site, the hierarchical organisation of Internet domains, or any constraints based on legal grounds. However, each of these exemplars can contain a degree of subjectiveness. A marketing expert might only be interested in all paths going through a recent campaign page leading to the purchase page, whereas the web administrator might be interested in the most regularly visited pages for caching purposes; an Internet service provider might want to cluster the sub-level domains *.edu*, *.ac.uk*, *uni-*.de* and *.edu.** into one virtual domain; and tax deduction schemes vary, depending from where an electronic buyer is logging into an electronic commerce site.

Subjective domain knowledge has a higher degree of context-dependency than its objective counterpart. As a consequence, either entire domain knowledge entities or large parts thereof (for example, resources) have to exist for multiple contexts and its incorporation in data mining exercises.

Typical cases of subjective domain knowledge are patterns (decision trees, neural networks, Dempster-Shafer pieces of evidence, et cetera), which have been discovered in a specialised data mining application, chosen threshold values, or age group bandings. Similar to objective domain knowledge, each of the examples can have a certain degree of objectiveness. Discovered rules with a very high support and confidence might be interpreted as quasi facts; thresholds can depend on business target goals; and age group bandings can be legally grounded.

It is desirable to handle the entire range of domain knowledge degrees of reality using the same underlying techniques. Thus, contextual domain knowledge is proposed in the next sub-section, independent of its degree of reality.

Contextual Domain Knowledge

Let *D* be the set of supported domain knowledge types and *C* a set of contexts as defined above. Let us further assume that each element in *D* can be represented as d_1, d_2, d_3, …, where each *d* represents a component of *D*. Then, each *d* can be allotted at least one context *c*.

Definition 6. **Contextual domain knowledge** is specified as $D=\{d_1^{C_1}, d_2^{C_2}, d_3^{C_3}, \mathbf{K}\}$, where each *d* is part of *D* and each *C* is a set of contexts (C ≠ ∅).

For further illustration, the supported types of domain knowledge are kept separately, supported by illustrative examples from the e-commerce domain.

Contextual bandings can be handled in three different ways, viz. totally reused, totally replaced and partially reused. Where the total options are straightforward, partial reusability can further be sub-divided. A sub-banding can either have the same upper or lower limit, be in between, or over-lapping with other ranges. Although it is technically feasible, it has not proven practical to deal with sub-bandings. It is therefore proposed that only total replacement and reusability of bandings is performed.

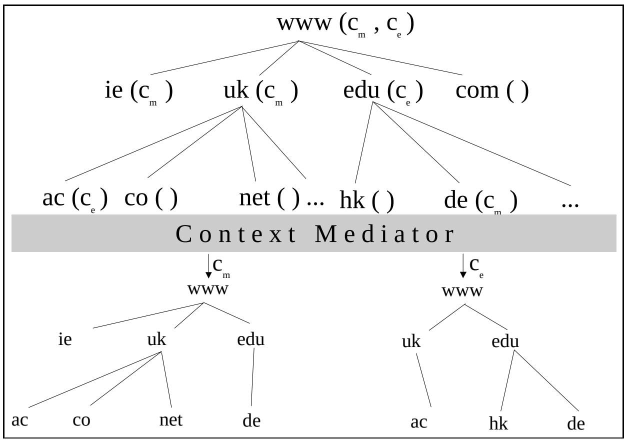

Figure 2. Example Contextual Concept Hierarchy

Contextual concept hierarchies are of more interest to data mining exercises. Owing to the fact that each node in a multi-level concept hierarchy is allotted a context identifier, it is possible to reuse a collection of nodes and their sub-trees. For instance, the marketing manager Europe (in context c_m) might only be interested in the sub-hierarchy with all European countries, whereas the person responsible for introducing a product at educational level (in context c_e) would only be concerned about the according sub-trees (see Figure 2).

Similarly to concept hierarchies, network structures contain context information at each node, which allows the reusability of a set of nodes and the existing links among them. For example, the author of online help pages on a multi-party retail web site (aka. shopping mall), might only be interested in the sub-network that includes all nodes which lead to dynamically created help pages.

Like bandings, parts of constraints as well as previously discovered knowledge can theoretically be contextualised However, it has no value in real-world data mining applications. Since user preferences are atomic numeric values, they are treated only holistically and are not further split apart.

Context Mediation

The original purpose of the context mediator was to resolve conflicts and guarantee interoperability among data sources, which have been accessed from different contexts (Büchner, Bell, and Hughes 1998). In order to use the context mediator for data as well as domain knowledge it has to be polymorph in nature. Here, only the domain knowledge-specific functionality of the context mediator is described.

The context mediator has to decide what domain knowledge is to be included and what is to be excluded from a data mining task. This decision is based on the context the knowledge has been created in and the context it is to be applied to. These two sites are referred to as knowledge source *s* and knowledge receiver *r*.

As outlined in the previous sub-section, each piece of knowledge has a set of contexts allotted to it. Owing to the fact that a user can only be in one context at a time, the context mediator only returns the pieces of domain knowledge that have been allotted the context in which the user is currently in. More formally, this can be expressed as following.

context mediator **ß** c_r

$$D_r := \mathbf{U} d_s \big| \, d_s \in D_s \wedge c_s(d_s) = c_r$$

context mediator **à** D_r

Figure 3. Context Mediator Structure

Now that all components have been designed, viz. contextual domain knowledge as well as its data source-based counterpart and the context mediation facility, the next step is to incorporate the parts into a knowledge discovery architecture.

Contextual Data Mining Architecture

In order to deploy contextual data as well as domain knowledge in a data mining application, a simplified contextual knowledge discovery architecture has been created, which is depicted in Figure 4.

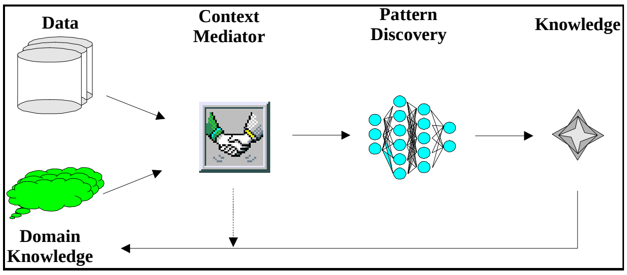

Figure 4. Simplified Contextual KDD Architecture

The context mediator contains all the information about participating contexts, their hierarchical organisation as well behavioural aspects. It is polymorph in nature and thus can deal with requests concerned about data (Büchner, Bell, and Hughes 1998) as well as domain knowledge (this paper). Depending on the context from which information is requested, data and suitable domain knowledge is used as input for knowledge discovery. The discovered patterns are then contextualised, that is labelled with the context the data mining exercise has been performed in, and fed back to the domain knowledge repository.

In order to illustrate the operation of the outlined components in the architecture, consider an electronic commerce example, in which the task is to discover sequential patterns from internet log files, which are then be interpreted as behavioural patterns. Both, the types of stored data as well as incorporated marketing-related domain knowledge depends of the type of e-tailer that is operating the site. Having site-specific log files in an internet bookstore environment which contains information about URI, login time, logoff time, HTTP referrer, status, cookie ID, et cetera, the marketing manager is looking for interesting patterns, and so is the web administrator[1]. The marketing expert has specified his/her domain knowledge in form of region-based concept hierarchies as well as target-related age bandings. The web administrator however, has created a network of the topology of the retailer's web site. The threshold parameters provided by the experts are budget- and cache size-driven, respectively. The types of sequences (associations across time) which are discovered are most likely to be different, since they are goal- as well as context-driven. They can even be contradictionary, which happens more regularly when classifications or associations are to be discovered. Finally, each piece of knowledge that is found and chosen by the domain expert to be kept is tagged with the current context and memorised in the domain knowledge repository for future usage.

Conclusions and Further Work

The consideration of context information has been proposed in related disciplines, mainly in case-based reasoning (Öztürk and Aamodt 1997; Jurišica and Glasgow 1997; Dubitzky et al. 1999), but has, to the best of our knowledge, not yet been applied in data mining scenarios. We have closed that link in proposing an architecture that considers contextual data and domain knowledge, as well as a context mediation facility, which reconciles discrepancies among the participated entities.

Ongoing research is orientated towards three main directions. First is dealing with a higher degree of uncertainty, which allows the allotment of contextual weights (Turner, 1997) and requires a more sophisticated context mediator. On the same terms, the allocation of a data element or a piece of domain knowledge to more than one context has potential applications, but requires even more complicated mediation facilities. Second is investigating the impact of allotting context information not only to nodes but also to edges in graph-based domain knowledge (for example, the link from a home page to a search page on an online bookstore might be less interesting from a marketing perspective than a link from the home page to a special offer). Finally, contextual information as such is used by data mining algorithms themselves in order to discover more personalised patterns.

References

Anand, S.S., Bell, D.A. and Hughes, J.G. 1995. The Role of Domain Knowledge in Data Mining, in *Proc. 4th Int'l ACM Conf. on Information and Knowledge Management*, pp. 37-43.

Anand, S.S. and Büchner, A.G. 1998. *Decision Support using Data Mining*, FT Pitman Publishers.

Büchner, A.G., Bell, D.A. and Hughes, J.G. 1998. A Contextualised Object Data Model based on Semantic Values, in *Proc. 11th Int'l. Conf. on Parallel and Distributed Computing Systems*, pp. 171-176.

Büchner, A.G. and Mulvenna, M.D. 1998. Discovering Internet Marketing Intelligence through Online Analytical Web Usage Mining, *ACM SIGMOD Record*, 27(4) 54-61.

Büchner, A.G., Hughes, J.G and Bell, D.A. 1999. Contextual Data and Domain Knowledge for Incorporation in Knowledge Discovery Systems, submitted to *CONTEXT'99*.

Dubitzky, W., Büchner, A.G., Hughes, J.G. and Bell, D.A. 1999. Towards Concept-Oriented Databases, *Journal on Data and Knowledge Engineering*, forthcoming.

Jurišica, I. and Glasgow, J. 1997, Improving Performance of Case-Based Classification using Context-based Relevance, *Int'l Journal of Artificial Intelligence Tools*, 6(3&4): 511-536.

Klösgen, W. and Żytkow, J. eds. 1999. *Handbook of Data Mining*, Oxford University Press, forthcoming.

Öztürk, P. and Aamodt, A. 1997. Towards a model of context for case-based diagnostic problem solving, *Proc. 1st Int'l and Interdisciplinary Conf. on Modeling and Using Context*, pp. 198-208.

Silberschatz, A. and Tuzhilin, A. 1996. What Makes Patterns Interesting in Knowledge Discovery Systems. *IEEE Trans. on Knowledge and Data Engineering* 8(6):970-974.

Turner, R.M. 1997. Determining the context-dependent meaning of fuzzy subsets, in *Proc. 1st Int'l and Interdisciplinary Conf. on Modeling and Using Context*, pp. 233-242.

[1] In this scenario, only one log file in considered for simplicity. More complex constellations occur regularly in electronic shopping malls with multiple sources and receivers. Also, the example can easily be extended to two marketing experts with different responsibilities.