AutoDomainMine: A Graphical Data Mining System for Process Optimization

Aparna S. Varde[1], Elke A. Rundensteiner[2] and Richard D. Sisson Jr.[3]
1. Department of Math and Computer Science, Virginia State University, Petersburg, VA, USA
2. Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA, USA
3. Manufacturing and Materials Science Program, Worcester Polytechnic Institute, Worcester, MA, USA
avarde@vsu.edu, rundenst@cs.wpi.edu, sisson@wpi.edu

ABSTRACT

This paper describes a graphical data mining system called AutoDomainMine. It is based on our proposed approach of integrating clustering and classification to mine scientific data stored in a database. The data consists of input conditions of scientific experiments and graphs plotted as their results. This system mines the stored data in order to submit exact or approximate ranked responses to user queries intended to optimize the scientific processes.

Categories and Subject Descriptors: H.2.8 [Database Management]: Database Applications - data mining, scientific databases

General Terms: Design, Experimentation, Human Factors

Keywords: Approximate Query Processing, Ranking, Search Heuristics

1. INTRODUCTION

Experimental results in scientific domains are often plotted as 2-dimensional graphs of dependent versus independent variables. Such graphs visually assist the comparison of the corresponding processes. Data on the input conditions and graphs from existing experiments is typically stored in a database. Users are often interested in posing queries whose responses help to optimize the concerned scientific processes. Typical examples of such queries include:

1. Given all the input conditions of an experiment, display the most likely resulting graph.
2. Given the desired graph in an experiment, determine the most appropriate input conditions to achieve it.
3. Given a few but not all of input conditions of an experiment, return the graph that would most probably be obtained.
4. Given desired ranges of some but not all features on a graph, return the input conditions likely to achieve a similar result.

The responses to the first two queries can be based on exact matches if data on the corresponding experiment happens to be available in the database. However, if the particular experiment has not been performed yet, then ranked responses can be returned based on approximate matches. The third and fourth queries would need to be answered using approximate matches, since the user provides only partial data on the input conditions or graphs.

The problem of providing approximate answers to queries in scientific databases has been addressed in the literature through approaches such as naive and weighted similarity searches for multimedia data [5] and mathematical modeling for simulations [6]. However, such approaches are not found suitable in our targeted applications due to insufficient accuracy, excessive domain expert involvement during query processing and lack of efficiency. This motivates the development of an approach that provides more accurate answers to such queries, with minimal intervention by experts and in distinctly less time than a real experiment.

We have proposed such a query processing approach called AutoDomainMine [8] based on graphical data mining. In this paper we describe the AutoDomainMine system developed using this approach. In AutoDomainMine data analysis is performed by integrating clustering and classification. Graphs from existing experiments are first clustered taking into account their semantics by learning a domain-specific distance metric [10]. Decision tree classification is then used to identify the conditions characterizing each cluster in order to build a representative pair of input conditions and resulting graph per cluster. This data analysis forms the basis for similarity-driven query processing.

Consider that the user submits a query in terms of all or a few of the input conditions of an experiment. The given conditions are compared with the decision tree paths in order to find the closest match. The relative importance of the conditions in the domain is incorporated in this comparison through our proposed decision tree weight heuristic [9]. The representative graph of the cluster corresponding to the closest matching path is returned as the most likely graph that would be obtained in the given experiment.

Alternatively, imagine that the user query includes a desired graph or some of its features. The given graph is compared with all the representative graphs (or the given features are compared with the corresponding features of the representative graphs). During this comparison the semantics of the graphs is captured through our learned domain-specific distance metric [10]. The representative conditions of the closest matching graph are conveyed as most probable conditions to achieve the desired result.

Thus, our search for an approximate answer incorporates semantics by learning domain-specific notions of distance for graphs and conditions. In addition, while displaying answers to users, ranking is performed by capturing targeted user interests through our proposed encodings [7, 9] analogous to the Minimum Description Length principle [3]. The encodings are based on factors such as levels of detail and type of information needed in different applications.

Copyright is held by the author/owner(s).
SIGMOD'07, June 11–14, 2007, Beijing, China.
ACM 978-1-59593-686-8/07/0006.

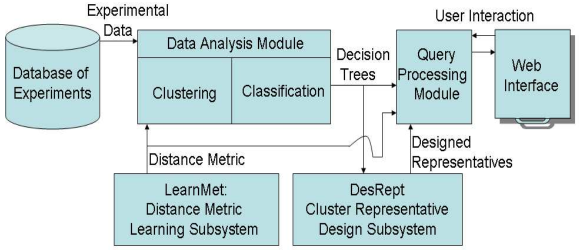


Figure 1: AutoDomainMine System Architecture

2. THE AUTODOMAINMINE SYSTEM

Figure 1 shows the architecture of the AutoDomainMine system. This system has been developed in Java. Its main modules are described below.

Database of Experiments. This stores the data from actually performed scientific experiments in terms of the input conditions and the resulting graph for each experiment. The database has been developed using MySQL.

Data Analysis Module. This module integrates clustering and classification to analyze data from existing experiments. It performs clustering using the k-means algorithm [3] adapting it to graphs that are curves by our proposed mapping [8]. Decision tree classification is performed using the J4.8 algorithm [3] using our proposed heuristic [8].

Distance Metric Learning Subsystem. We have proposed a technique called LearnMet [10] to learn domain-specific distance metrics for graphs. This technique forms the basis for the development of a subsystem with the goal of distance metric learning. See Section 3.

Cluster Representative Design Subsystem. We have proposed a methodology called DesRept [9, 7] in order to design semantics-preserving cluster representatives for input conditions and graphs. This is used to implement the subsystem for cluster representative design. See Section 3.

Query Processing Module. This uses the decision trees and designed representative pairs to process the queries. It provides exact or approximate answers based on the extent to which the user-submitted conditions / graphs match the decision trees / representative graphs. Thresholds for comparison are used based on our decision tree heuristic for conditions [9] and distance metric for graphs [10].

Web Interface. The user interacts with the system through the web interface. This has been developed using Javascript and Apache Tomcat.

The AutoDomainMine system targets scientific domains where process optimization is often performed by analyzing the graphical results of existing experiments. AutoDomainMine has been rigorously evaluated with real data from the domain of Heat Treating of Materials [6] that motivated this work. It is evident from formal user surveys that this system meets the goals of accuracy, efficiency and minimal domain expert intervention in real applications including parameter selection and decision support systems.

3. TECHNICAL CHALLENGES

Distance Metric Learning with LearnMet. Several distance metrics such as Euclidean and statistical distances exist in the literature [3]. However it is not known a priori which metric(s) would best preserve semantics if used as the notion of distance in clustering. Experts at best have vague notions about the relative importance of regions on the graphs but do not have a defined metric. Existing distance metric learning techniques, e.g., [2, 11] are either inapplicable or not accurate enough in this context. We have proposed a technique called LearnMet [10] to learn domain-specific distance metrics for graphs. A LearnMet metric D is a weighted sum of components where each component is an individual metric such as Euclidean or statistical distance [3] and its weight gives its relative importance in the domain. Using the LearnMet approach, actual clusters of graphs given by experts are iteratively compared with predicted clusters obtained from a clustering algorithm. In the first iteration, a guessed metric D is used for clustering. This metric is adjusted based on error between predicted and actual clusters using our proposed weight adjustment heuristic [10] until the error is below a given threshold or a maximum number of epochs is reached. The metric with error below threshold or with minimal error among all epochs is returned as the learned metric.

Cluster Representative Design with DesRept. Since several sets of conditions lead to a single cluster, randomly selecting any one as a representative causes information loss. Also, random selection of a representative for graphs does not incorporate semantics and ease of interpretation based on user interests. Approaches in the literature such as [1, 4] do not address the problem of preserving cluster semantics through representatives as per our needs. We have proposed a methodology called DesRept [7, 9] to design semantics-preserving cluster representatives for input conditions and graphs. Two design methods of guided selection and construction are used to build candidate representatives capturing various types of information within the cluster. Candidates are compared using encodings we have proposed [7, 9] analogous to the Minimum Description Length principle [3]. Criteria in these encodings are the complexity of the representative and the information loss due to it based on user interests. The winning candidate for conditions and graphs in each cluster is output as its designed representative. Likewise, designed representatives showing information at different levels of detail are ranked based on the interests of targeted users conveyed in the encoding.

Key Contributions. We summarize the main technical contributions of this work as follows.

1. AutoDomainMine for Graphical Data Mining
   - Integrating clustering and classification for data analysis
   - Adapting clustering to curve data types
   - Proposing search methods for approximate matching
2. LearnMet for Distance Metric Learning
   - Intelligently guessing an initial metric
   - Defining a notion of error
   - Proposing weight adjustment heuristics
   - Learning simple metrics capturing domain knowledge
3. DesRept for Designing Cluster Representatives
   - Defining a notion of distance for conditions and graphs
   - Developing strategies to design candidate representatives
   - Proposing encodings for comparison to select winners
   - Ranking responses based on user preferences in encodings

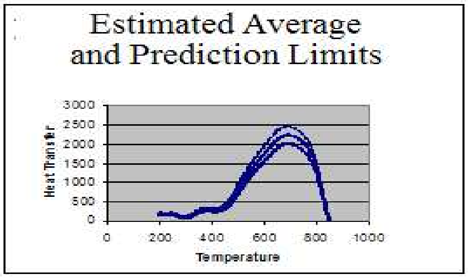
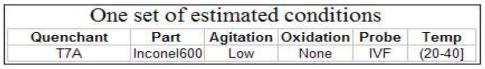


Figure 2: Summarized Graph



One set of estimated conditions

Quenchant	Part	Agitation	Oxidation	Probe	Temp
T7A	Inconel600	Low	None	IVF	(20-40]

Figure 3: Single Set of Conditions

4. SYSTEM DEMONSTRATION

We demonstrate AutoDomainMine using real data from the Heat Treating domain [6]. We sketch a few sample scenarios below. However AutoDomainMine is a full-fledged trademarked tool and many more scenarios can be shown in a live demonstration.

User-submits conditions, requests graph. The following scenarios are demonstrated for approximate match. This reasoning in these scenarios is justified through our decision tree heuristic [9].

- Scenario 1 - The more important conditions (at or above half the height of the tree) do not match: This is considered insufficient to provide an approximate answer.

- Scenario 2 - The more important conditions are matching but the less important conditions (below half the height of the tree) are not: An approximate match is provided based on majority class as will be demonstrated in the graphical user interface.

User submits graph, requests conditions. We demonstrate the following scenarios for approximate match. We use a similarity threshold for graphs here [10].

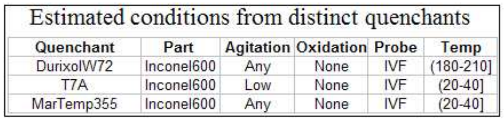
- Scenario 3 - No match is found within the given similarity threshold: It is conveyed that the conditions to obtain this graph cannot be displayed.

- Scenario 4 - More than one graph matches within threshold but there is no complete match: The graph with the closest match is selected with majority class in case of ties. This will be demonstrated in the graphical user interface.

Ranking of responses. We rank answers to queries based on user preferences learned through our encodings. The following scenarios are demonstrated.

- Scenario 5 - User of a parameter selection application submits input conditions and requests the graph: It is found from the encodings that such users prefer responses showing moderate detail and summarizing information. Hence an approximate answer is constructed, e.g., as shown in Figure 2.

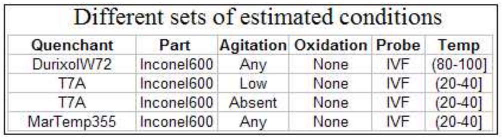
- Scenario 6 - User of a decision support application submits a graph and requests the conditions: It is found that such users prefer various levels detail from less to more. Thus approximate responses are returned in a ranked order, e.g., a single set of estimated conditions nearest to all others in the matching cluster (Figure 3); multiple sets of estimated conditions summarizing cluster information in a domain-specific manner (Figure 4); all sets of estimated conditions in the matching cluster abstracted using domain knowledge (Figure 5).



Estimated conditions from distinct quenchants

Quenchant	Part	Agitation	Oxidation	Probe	Temp
DurixolW72	Inconel600	Any	None	IVF	(180-210]
T7A	Inconel600	Low	None	IVF	(20-40]
MarTemp355	Inconel600	Any	None	IVF	(20-40]

Figure 4: Summary - Multiple Sets of Conditions



Different sets of estimated conditions

Quenchant	Part	Agitation	Oxidation	Probe	Temp
DurixolW72	Inconel600	Any	None	IVF	(80-100]
T7A	Inconel600	Low	None	IVF	(20-40]
T7A	Inconel600	Absent	None	IVF	(20-40]
MarTemp355	Inconel600	Any	None	IVF	(20-40]

Figure 5: All Sets of Conditions Abstracted

Based on several scenarios with real user studies it is found that the AutoDomainMine system meets user expectations. It is a novel technology of integrating clustering and classification to analyze existing experimental data for approximate query processing. In general, the AutoDomainMine approach is useful for graphical data mining in scientific domains to aid process optimization. It has useful applications such as computational estimation, parameter selection, simulation tools, intelligent tutors and decision support systems.

5. ACKNOWLEDGMENTS

This work has been supported by the Center for Heat Treating Excellence (CHTE) and its member companies, and by the Department of Energy - Industrial Technology Program (DOE-ITP) Award Number DE-FC-07-01ID14197. The AutoDomainMine tool is a trademark of the Center for Heat Treating Excellence.

6. REFERENCES

[1] O. Buyukkokten, H. Garcia-Molina and A. Paepcke, Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices, *WWW*, May 2001, pp. 652-662.

[2] A. Hinneburg, C. Aggarwal and D. Keim, What is the Nearest Neighbor in High Dimensional Spaces, *VLDB*, Aug 2000, pp. 506-515.

[3] J. Han and M. Kamber, *Data Mining Concepts and Techniques*, Morgan Kaufmann, 2001.

[4] P. Janecek and P. Pu, *Opportunistic Search with Semantic Fisheye Views*, Swiss Federal Institute of Technology, Lausanne, 2004, TR IC/2004/42.

[5] D. Keim and B. Bustos, Similarity Search in Multimedia Databases, *ICDE Tutorial*, Mar 2004, pp. 873-874.

[6] G. Stolz, *Heat Transfer*, John Wiley, 1960.

[7] A. Varde, E. Rundensteiner, C. Ruiz, M. Maniruzzaman and R. Sisson Jr., Effectiveness of Domain-Specific Cluster Representatives for Graphical Plots, *SIGMOD IQIS*, Jun 2006, pp. 24-29.

[8] A. Varde, E. Rundensteiner, C. Ruiz, D. Brown, M. Maniruzzaman and R. Sisson Jr., Integrating Clustering and Classification for Estimating Process Variables in Materials Science, *AAAI Poster*, Jul 2006.

[9] A. Varde, E. Rundensteiner, C. Ruiz, D. Brown, M. Maniruzzaman and R. Sisson Jr., Designing Semantics-Preserving Cluster Representatives for Scientific Input Conditions, *CIKM*, Nov 2006, pp. 708-717.

[10] A. Varde, E. Rundensteiner, C. Ruiz, M. Maniruzzaman and R. Sisson Jr., Learning Semantics-Preserving Distance Metrics for Clustering Graphical Data, *KDD MDM*, Aug 2005, pp. 107-112.

[11] E. Xing, A. Ng, M. Jordan and S. Russell, Distance Metric Learning with Application to Clustering with Side Information, *NIPS*, Dec 2003, pp. 503-512.