DATA MINING OVER GRAPHICAL RESULTS OF EXPERIMENTS WITH DOMAIN SEMANTICS[1]

Aparna Varde, Elke Rundensteiner, Carolina Ruiz, Mohammed Maniruzzaman and Richard Sisson Jr.

Worcester Polytechnic Institute (WPI)
Worcester, Massachusetts 01609, USA
Phone: (508) 831- 5681, Fax: (508) 831- 5776
E-mail: (aparna | rundenst | ruiz | maniruzz | sisson)@wpi.edu

Abstract

The results of experiments in science and engineering are often represented graphically, since graphs serve as good visual tools for analysis of the corresponding processes to aid decision-making. Performing a laboratory experiment consumes time and resources. This motivates the need to estimate the results (graphs) that would be obtained from experiments given the input conditions. We propose an approach called "AutoDomainMine" for estimation. This consists of first clustering graphical results of existing experiments, and then using classification to learn the clustering criteria to build a representative pair of input conditions and graph per cluster. The representatives and learnt criteria form domain knowledge useful for estimation. We have found that this strategy provides significantly better estimation compared to similarity search and other methods, since it automates a learning method of scientists in discovering knowledge from experimental results. Clustering graphs involves issues such as preserving domain semantics, defining similarity measures and reducing dimensionality with minimal loss. AutoDomainMine, with its challenges and evaluation, is discussed here.

Keywords: *Artificial Intelligence, Graphical Data Mining, Decision Support Systems*

1. Introduction

Experimental data in science and engineering domains is a good source of knowledge useful in making decisions about industrial applications. The results of experiments are often plotted graphically since graphs are good visual tools for analysis and comparison of the corresponding processes [V-04]. Performing an experiment in the laboratory consumes significant time and resources. This motivates the need to *estimate* the results given the input conditions to avoid conducting all possible experiments. For example, in the domain "Heat Treating of Materials", that inspired this research, running a laboratory experiment takes approximately 5 hours, while the resources incur a one-time cost worth thousands of dollars and recurrent costs worth hundreds of dollars [M-95, SMMV-04]. The result of the experiment is a graph called a heat transfer coefficient curve, i.e., a plot of heat transfer coefficient h_c versus temperature *T,* where h_c characterizes an experiment by representing how a material reacts to rapid cooling [M-95]. Materials Scientists are interested in analyzing this graph to support decisions about process optimization in the industry. For instance, in the steel *ST4140*, a graph with a steep slope implies fast heat extraction that leads to high strength. The conditions used to obtain this graph could be used to treat the steel in an industrial application that needs such a feature. Since the inferences drawn from graphs aid decision-making, it would be beneficial if the graph obtained in an experiment could be estimated given the input conditions. Similarly, decision-making about selection of products and process parameters could be supported by estimating the conditions that would obtain a desired graph. This motivates the need for an estimation technique.

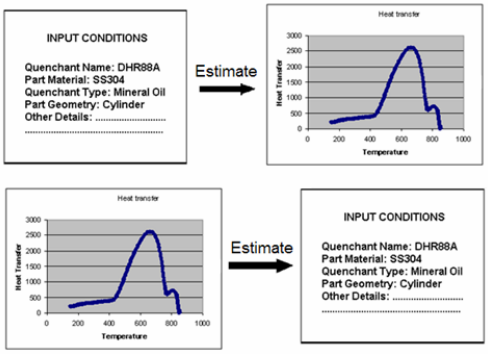

Figure 1: Goals of Estimation Technique

[1] This work is supported by the Center for Heat Treating Excellence (CHTE) and its member companies, and by the Department of Energy - Office of Industrial Technology (DOE-OIT) Award Number DE-FC-07-01ID14197.

In general, the goals of the required estimation technique are: (See also visually depicted in Figure 1).

1. Given the input conditions of an experiment, estimate the graph obtained.
2. Given the desired graph in an experiment, estimate the input conditions needed to obtain it.

The graphs involved here are 2-dimensional curves that plot one dependent variable versus one independent variable. There is a correlation between input conditions and graphs, as is seen in most science and engineering domains. Data from existing experiments is stored in a database. Approaches such as naive similarity search [HK-01], weighted similarity search [WF-00], case-based reasoning [AP-03] and mathematical modeling [M-95], all using existing experimental data are not adequate for estimation in our targeted domains as observed by us and corroborated by domain experts [SMMV-04]. There is a need to develop a technique that performs estimation efficiently with accuracy acceptable for decision support.

We propose an approach called AutoDomainMine to meet the above goals and needs [V-04]. In this approach, we first cluster existing experiments based on their graphical results. We then use classification to learn the clustering criteria and apply the learnt criteria to build a representative pair of input conditions and graph per cluster. This forms knowledge discovered for estimating the results of new experiments given their input conditions, and vice versa. This approach automates one typical manner in which scientists learn [SMMV-04]. They group experimental results based on similarity, and then reason the causes of similarity by identifying combinations of conditions characterizing each group. Automating this learning strategy of scientists by integrating clustering and classification into one integrated approach as indicated above is one of the key contributions of our research.

There are several challenges that must be tackled in order to ensure the viability of this approach. Clustering graphs that are curves is an issue, since clustering techniques were originally developed for points [HK-01]. We propose a mapping to address this problem. Preserving domain semantics is also crucial. Some aspects on a graph may be more significant than others. We propose to capture domain semantics by defining an appropriate similarity measure for clustering. Moreover, since a graph typically has hundreds of points or is a continuous stream of data, it is not feasible to capture every instance of it. Hence dimensionality reduction is needed [HK-01, GW-02]. Reducing dimensionality with minimal loss depending on the nature of the graphs in domain is another significant issue. These and other challenges are discussed in the paper.

AutoDomainMine is evaluated primarily in the Heat Treating domain. Its main application is enhancing a decision support system "QuenchMiner" [VTRWMS-04] that estimates parameters in an experiment given the input conditions.

Section 2 of this paper introduces the AutoDomainMine approach. Sections 3 and 4 focus on the details of its clustering and classification steps respectively. Section 5 explains the details of estimation. Section 6 outlines related work. Section 7 summarizes evaluation of the tool developed using AutoDomainMine. Section 8 gives the conclusions and ongoing work.

2. Proposed Approach: AutoDomainMine

2.1 Steps of Approach

The proposed approach, namely, AutoDomainMine [V-04] involves a two-step process. It first discovers knowledge from experimental results by integrating clustering and classification, and then uses the discovered knowledge to estimate graphs given input conditions or vice versa. Clustering is the process of placing a set of physical or abstract objects into groups of similar objects [HK-01]. Classification is a form of data analysis that can be used to extract models to predict categorical labels [WF-00]. These two data mining techniques are integrated in AutoDomainMine as explained below.

In the knowledge discovery step, clustering is done over the graphical results of existing experiments stored in the database. Since clustering techniques were originally developed for points [HK-01], a mapping is proposed that converts a 2-dimensional graph into an N-dimensional point. A similarity measure is defined for clustering graphs, taking into account domain semantics. Once the clusters of experiments are identified by grouping their graphical results, a syntactic label is obtained for each cluster. The cluster labels form the classification target. The clustering criteria, namely, the input conditions that characterize each cluster are then learnt by decision tree classification. This helps understand the relative importance of the conditions in clustering. The paths of each decision tree are then traced to build a representative pair of input conditions and graph for each cluster. The decision trees and representative pairs form the discovered knowledge.

Estimation is then performed using this knowledge. In order to estimate a graph, given a new set of input conditions, the decision tree is searched to find the closest matching cluster. The representative graph of that cluster is the estimated graph for the given set of conditions. To estimate input conditions, given a desired graph in an experiment, the representative graphs are searched to find the closest match. The representative conditions corresponding to the match are the estimated input conditions that would obtain the desired graph. Since the relative importance of the conditions has been learnt in the knowledge discovery step, the estimation is more accurate than that with a similarity search [V-04].

The AutoDomainMine steps are summarized below. *Knowledge Discovery* is a one-time process executed offline with existing experiments in the database, while *Estimation* is a recurrent process executed online with each user-submitted case.

THE AUTODOMAINMINE APPROACH

1. ***Knowledge Discovery:*** *Discover knowledge from experimental results*
   - a. ***Clustering:*** *Cluster existing experiments based on their graphs*
     - i) *Develop mapping from graphs to points, preserving domain semantics*
       - *Perform dimensionality reduction if needed*

ii) Define similarity measure in clustering
iii) Cluster graphs using mapping and similarity measure
- Thus obtain a cluster label for each experiment

b. ***Classification:*** *Use classifiers to learn the clustering criteria and build representatives*
i) Treat cluster label of each experiment as its classification target
ii) Use decision tree classifiers to identify combinations of input conditions that predict the target
- Thus learn conditions that characterize each cluster
iii) Build a representative pair of input conditions and graph per cluster by tracing paths of decision trees

2. ***Estimation:*** *Use discovered knowledge to estimate graphs or conditions*

a. ***Graph:*** *Estimate graph, given input conditions*
i) Accept input conditions from user
ii) Trace decision tree paths to estimate cluster for given conditions
iii) Convey representative graph of that cluster as the estimated graph

b. ***Conditions:*** *Estimate conditions, given desired graph*
i) Accept desired graph from user
ii) Compare with representative graphs of clusters to find closest match
iii) Convey corresponding representative conditions as estimated conditions for desired graph

2.2 Learning Analogous to Scientists

The proposed approach automates a learning strategy of scientists [SMMV-04]. They often group experiments based on the similarity of the graphs obtained. They then reason the causes of similarity between groups in terms of the impact of the input conditions on the graphs. This is illustrated in Figure 2. For example, the following facts were learnt by Materials Scientists from the results of experiments [Source: CHTE, WPI].

- *Thin* oxide on a part surface causes vapor blanket around the part to break, resulting in *fast* cooling.
- *Thick* oxide on a part surface acts as an insulator, resulting in *slow* cooling.

This learning [SMMV-04] was done by:

- Performing laboratory experiments with *thick* or *thin* oxide among the input conditions,
- **Grouping** based on **graphical results,**
- **Reasoning** based on **input conditions.**

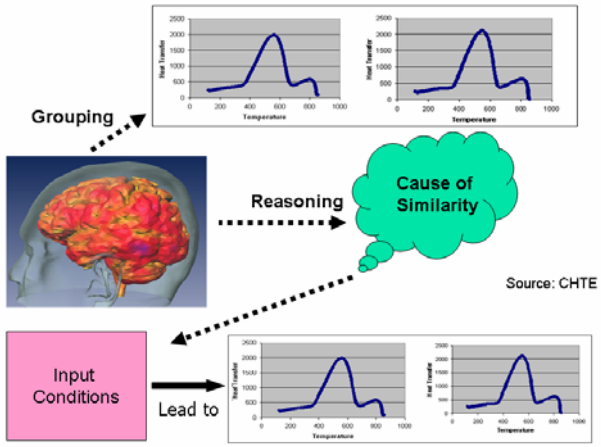

Figure 2: A Learning Strategy of Scientists

This learning strategy is automated in our approach by integrating clustering and classification. The two main aspects are:

A) Clustering followed by Classification: Clustering is basically unsupervised learning with no predefined labels [HK-01]. It is therefore analogous to a scientist grouping similar graphs by observation. Classification on the other hand is supervised learning where the class labels of the training samples are provided. The classification target is pre-defined [WF-00]. This is analogous to the scientist reasoning what combinations of input conditions characterize a group after the groups are formed. Hence in our context, it is essential to first do clustering and then classification. The combined approach works better than either one individually in solving the estimation problem. Only clustering would not help categorize a new user-submitted experiment to estimate its results. The causes of similarities need to be identified by classification. However, only classification also would not help, since initially there are no labeled groups. Graphs themselves cannot be classification targets. Thus clustering is needed to group graphs and define labels before classification [V-04].

B) Clustering based on Graphical Results: It is significant to note that the clustering is done based on the graphs, i.e., results, not based on input conditions. This is because clustering the input conditions adversely affects accuracy since the clustering algorithm by default attaches the same weight to all the conditions. This cannot be corrected by adding weights to the conditions before clustering because the weights are not known apriori. For example in Heat Treating, moderate agitation of the cooling medium may be less significant than a thick oxide layer on the part surface, while high agitation may be more significant [SMMV-04]. There is a need to learn this from experiments by analyzing their results [V-04].

Several machine learning approaches have been integrated in the literature such as classification and association rules [LHM-98], case-based and rule-based reasoning [PC-97], association rules and clustering [LSW-97], and some forms of supervised and unsupervised learning [CDKM-98]. Neural networks have been used for both clustering and classification [M-97]. Clustering has been done with hidden labels and evaluated for aiding classification [BL-04]. However, to the best of our knowledge, AutoDomainMine is among the first to integrate clustering and classification into one single learning strategy for estimation, by clustering graphical results of experiments and then using classification to learn the clustering criteria, thus automating a learning method of scientists [V-04].

3. Clustering

Clustering places items into groups such that items within a group have high similarity but are different from items in other groups [HK-01]. The notion of similarity is based on the distance between the items. Among clustering algorithms in the literature such as k-means [KR-90], Expectation Maximization (EM) [WF-00] and COBWEB [HK-01], one popular method in AutoDomainMine is k-means due to its simplicity and partitioning-based approach.

3.1 Applying Clustering to Graphs

Mapping: Clustering algorithms such as k-means [KR-90] have been originally developed for points. In order to apply them to graphs that are curves, we propose a mapping that converts a 2-dimensional curve to an N-dimensional point. The x-coordinates on each curve are treated as separate axes, forming the *N* dimensions while the y-coordinates are treated as lengths along these *N* dimensions respectively. Figure 3 shows an example of this mapping. *X_100... X_850* indicate the *N* dimensions [V-04].

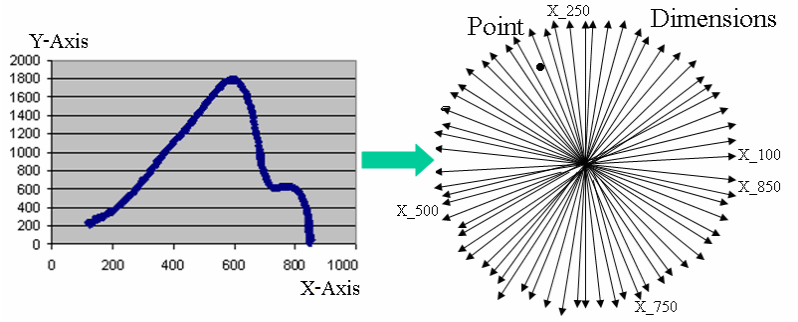

Figure 3: Mapping a 2-D curve to an N-D point

Notion of Similarity: The default notion of similarity in clustering algorithms is Euclidean distance. AutoDomainMine uses this basic notion as the similarity measure between graphs, but weights the dimensions based on their relative importance as needed. This depends on the domain and the nature of the graphs.

For example, in Heat Treating, it is known that there are mainly 3 critical points on a heat transfer coefficient curve, i.e., the point of maximum heat transfer coefficient, the Leidenfrost point at which rapid cooling begins, and the heat transfer corresponding to the Boiling Point of the Quenchant. These points denote the separation of phases that correspond to different physical processes in the domain [M-95]. These are depicted in Figure 4. The critical points are stored along with the graphs in the database. The critical points do not always occur at the same x-coordinates. For instance, in one experiment, the h_{Max} or maximum heat transfer coefficient may occur at a temperature of 650 degrees Celsius, while in another, it may be at 550 degrees Celsius. Thus the critical points are stored as additional dimensions, namely, *X_Max, X_Leidenfrost* and *X_BP*. These are considered over and above the dimensions already represented by mapping the 2-dimensional curve to an N-dimensional point.

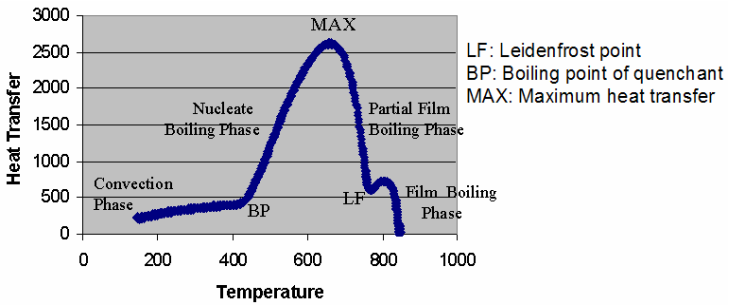

Figure 4: Critical Points on a Heat Transfer Coefficient Curve

In domains with graphs having critical points, AutoDomainMine uses weighted Euclidean distance as the similarity measure by attaching greater weights, as provided by domain experts, to the corresponding critical dimensions. In other domains, AutoDomainMine uses basic Euclidean distance as a similarity measure [V-04].

3.2 Dimensionality Reduction

Since a curve is typically composed of hundreds or thousands of points it is inefficient to convert each x-coordinate into a separate dimension. Moreover in some domains the curve may not be discrete but rather a continuous stream of data and it is clearly not feasible to capture every instance of it. Hence dimensionality reduction [HK-01] is used. Among various reduction techniques the two suitable in the context of our problem are described here.

Selective Sampling: This method is a variation of Random Sampling [HK-01]. In Random Sampling, points are sampled at random intervals. In Selective Sampling, we sample the points at regular intervals, and in addition sample critical points that correspond to important aspects in the domain. Knowledge of the domain as gathered from literature surveys and domain experts help in determining the important aspects. An original curve with thousands of points is thus reduced to an N-dimensional point with as many dimensions as needed to represent the curve, accurately and efficiently. For example in Heat Treating, approximately 53 samples are considered per curve, of which 50 are taken at regular intervals and 3 correspond to critical dimensions *X_Max, X_Leidenfrost and X_BP* [V-04].

Fourier Transforms: In this method the curve is mapped to Fourier Coefficients using Equation 1. This equation represents the n-point Discrete Fourier Transform of a curve $x_t, t = 0,1,...,n-1,$ which is defined to be a sequence X of n complex numbers, X_f, $f = 0,1,...,n-1$, where j is the imaginary unit $\sqrt{-1}$. Each X_f is a Fourier Coefficient [AFS-93].

$$X_f = \frac{1}{\sqrt{n}} \sum_{t=0}^{n-1} x_t \exp(-j2\Pi ft/n) \qquad f = 0,1,...,n-1 \quad \text{--- [1]}$$

The most significant Fourier Coefficients with respect to the domain are retained. Domain knowledge obtained from literature surveys and discussions with experts is useful in determining the significant coefficients. In Heat Treating, usually the first 16 coefficients are considered the most significant. This is because heat transfer coefficient curves are such that these 16 low frequency values contain useful information, while higher frequency values correspond to noise [SMMV-04].

3.3 Steps of Clustering

Considering all the above aspects, clustering in the AutoDomainMine approach is done using the following steps.

CLUSTERING IN AUTODOMAINMINE

1. *Obtain domain expert input about semantics of graphs*
   - *Thus determine critical regions if any and their weights*
2. *Map each 2-dimensional graph into an N-dimensional point*
3. *If N < = 100, go to step 5*
4. *Do dimensionality reduction*
   - *If critical regions present then dimensionality reduction technique is Selective Sampling*
   - *Else dimensionality reduction technique is Fourier Transforms*
5. *Define similarity measure*
   - *If critical regions present then similarity measure is weighted Euclidean distance*
   - *Else similarity measure is basic Euclidean distance*
6. *Send each graph (N-dimensional point) to clustering technique such as k-means*
   - *Thus obtain a cluster label for each graph and hence each experiment*
7. *Store each experiment with its input conditions and cluster label*

Dimensionality reduction is done if the number of dimensions is greater than 100. This is to reduce the curse of dimensionality in clustering [HK-01]. If there are critical dimensions, then Selective Sampling is the preferred method of dimensionality reduction, otherwise Fourier Transforms is preferred. It is found that the Fourier Transform method is somewhat more efficient than Selective Sampling, because fewer dimensions are required to map a curve to a point. For example in Heat Treating, 50 dimensions are needed for Selective Sampling as opposed to 16 for Fourier Transforms [SMMV-04]. In other domains also, as gathered from the literature, the first few (usually less than 20) Fourier Coefficients are retained since these correspond to low frequency values that are useful [GW-02, AFS-93]. However, in Fourier Transforms the critical dimensions cannot be considered separately in mapping the curves to the frequency domain. Nor can they be weighted before or after the mapping. This is due to the basic property of the Fourier Transform that involves converting a given waveform into frequency sinusoids such that they all sum back to the original waveform [GW-02]. In Selective Sampling we give higher weights to the critical dimensions, thus preserving accuracy with respect to critical regions. Hence this method is preferred in domains with critical regions on graphs [V-04].

Clustering is thus done in AutoDomainMine by sending either the original curves (N-dimensional points) or their Selective Samples or Fourier Transforms to a clustering technique such as k-means. The similarity measure is either Euclidean or weighted Euclidean distance, as justified earlier. Once the clusters are obtained for each graph (which represents each experiment), the output of the clustering needs to be sent to the classifier. Therefore each experiment is stored in terms of its input conditions and cluster label. This helps the classifier to reason the causes of clustering, i.e., the combinations of input conditions that lead to each cluster [V-04].

4. Classification

4.1 Decision Tree Classifiers

A decision tree [KK-95] is a flowchart-like structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions. Decision tree classification is used in AutoDomainMine because it serves as a good representation for categorical decision-making. Moreover, it is an eager learning approach, i.e., learns based on existing data without waiting for a new case to be classified, unlike lazy learners such as CBR [K-93]. It also provides explicit reasons for its decisions to justify the learning.

Figure 5 shows a sample partial output of the clustering step in AutoDomainMine using Heat Treating data. It depicts the input conditions of the experiments and the clusters in which their graphs are placed. This serves as the input to the classification step. Figure 6 shows a snapshot of a partial decision tree created for this data. The combinations of input conditions that characterize each cluster have been identified in this tree. For example, it can be inferred from the tree that a crucial clustering criterion is the Quenchant Name since that is the root of the tree. However, other criteria such as Quenchant Temperature are also important since a difference of temperature ranges in the experiments causes the corresponding graphs to be placed in different clusters. Thus the paths in the decision tree identify the clustering criteria.

QuenchantName	QuenchantTemp	Agitation	PartMaterial	Oxidation	Surface	ClusterID
Argon	(20-30]	Low	ST4140	5min	Medium	E
Water	(40-50]	High	AL6061	5min	Medium	D
DurixolHR88A	(60-70]	Absent	SS304	NO	Medium	B
HoughtQuenchG	(20-30]	Absent	SS304	NO	Medium	H
DurixolHR88A	(20-30]	Absent	SS304	NO	Medium	B
HoughtQuenchG	(60-70]	Absent	ST4140	5min	Medium	F
DurixolHR88A	(60-70]	Absent	ST4140	5min	Medium	B
DurixolV35	(60-70]	Absent	ST4140	5min	Medium	F
DurixolW72	(60-70]	Absent	ST4140	5min	Medium	F
HoughtQuenchG	(60-70]	Absent	SS304	NO	Medium	H
Water	(90-100]	High	AL6061	5min	Medium	C
Air	(20-30]	Low	SS304	5min	Medium	E
HoughtQuenchG	(20-30]	Absent	ST4140	5min	Medium	B

Figure 5: Sample Partial Input to Classifier

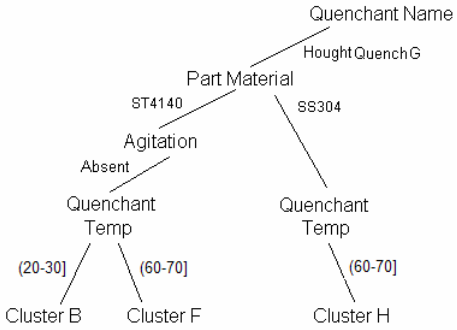

Figure 6: Snapshot of Partial Decision Tree

4.2 Building Representatives

The clusters developed and the clustering criteria learnt in AutoDomainMine form the basis for building a representative pair of input conditions and graph per cluster. Several combinations of input conditions may lead to one cluster. Also there are typically multiple graphs in each cluster. The process of building representatives is as follows [V-04].

BUILDING REPRESENTATIVES IN AUTODOMAINMINE

1. Trace the paths from the root to each leaf of the decision tree
2. Consider each path as a set of input conditions
3. Treat the leaf of each path as the cluster for that set of conditions
4. Among all graphs in that cluster randomly select one as a representative graph
5. Among all the paths (sets of conditions) leading to a particular cluster, select most frequent set as representative conditions
6. Store selected set of conditions and graph for each cluster as its representative pair

The justification for randomly selecting one graph as a representative is based on the concept of intra-cluster distance [HK-01]. The basic property of clustering is that it groups items so as to minimize the intra-cluster distance (and maximize the inter-cluster distance). Hence if several graphs are placed in the same cluster, it implies that the distance between them is low, i.e., they are similar. Therefore any one of them can be treated as a representative. For input conditions, if multiple combinations lead to the same cluster, it is desirable to use the one that resulted from the maximum number of experiments in the database. This is most likely to give the corresponding graph [SMMV-04]. Thus we select the most frequent set of input conditions leading to a cluster is as its representative. A sample representative pair for Cluster B is illustrated in Figure 7 with reference to the decision tree in Figure 6.

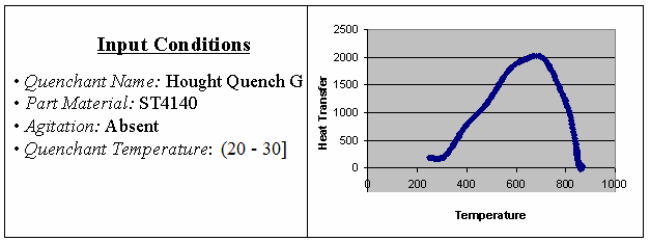

Figure 7: Sample Representative Pair for Cluster B

5. Estimation

5.1 Estimating Graph

Given a new set of user-submitted input conditions, the decision trees and representative pairs form the basis for estimating the graph that would be obtained. The process is as follows [V-04].

ESTIMATION OF GRAPH IN AUTODOMAINMINE

1. *Accept new input conditions from the user*
2. *Compare each path of decision tree of depth D with new input conditions*
3. *If search cannot proceed further, at or before level (D/2) of tree, then convey "cannot estimate graph" to the user, go to Step 1*
4. *If search cannot proceed after level (D/2 + 1), then representative graph of any cluster from that level is the estimated graph, go to Step 7*
5. *If only 1 complete path matches up to leaf, then representative graph of that cluster is the estimated graph, go to Step 7*
6. *If multiple paths match, then representative graph of any one cluster from those paths is the estimated graph*
7. *Display the estimated graph to the user*

Since the relative importance of the input conditions has already been learnt through the decision tree, this helps to make an educated guess about the closest matching cluster for the user-submitted input conditions. Hence if the more important conditions as identified by the higher levels of the tree do not match, this is considered insufficient to provide an estimate. However, if from the lower level onwards no complete match is found, then it is considered acceptable to give an estimate based on a partial match. The distinction between high and low levels is made depending on the depth of the tree. The levels at or above half the depth of the tree are considered as high and those below are half the depth as low.

5.2 Estimating Conditions

Given a desired user-submitted graph, the representative pairs form the basis for estimating the conditions that would obtain it. We define a similarity threshold for graphs as per the domain. The estimation process is then as follows [V-04].

ESTIMATION OF CONDITIONS IN AUTODOMAINMINE

1. Accept desired graph from the user
2. Compare desired graph with all the representative graphs
3. If no match is found within the given threshold then convey "cannot estimate input conditions" to the user, go to Step 1
4. If only 1 graph matches within threshold, then representative conditions of that graph are the estimated conditions, go to Step 6
5. If more than 1 graph matches, then representative conditions of closest matching graph are the estimated conditions
6. Display the estimated conditions to the user

Since a similarity measure for graphs is already defined in the clustering step, it is only necessary to define a threshold for similarity using this measure. If no match is found within the given threshold, then it implies that the desired graph cannot be obtained based on the knowledge discovered from existing experimental data. Thus it is conveyed to the user that the conditions to obtain this graph cannot be estimated. If only one representative graph matches the desired graph within the threshold, then it is obvious that the corresponding representative conditions are the estimated conditions. If several representative graphs match, it is desirable to select the closest match.

6. Related Work

One intuitive estimation approach is a naive similarity search over existing data [HK-01]. The given input conditions of a user-submitted experiment are compared with those of existing experiments to select the closest match as the number of matching conditions. However the non-matching condition(s) could be significant in the given domain. A weighted search [WF-00] guided by basic domain knowledge could possibly be used to overcome this problem. The relative importance of the search criteria, i.e., input conditions are coded as weights into feature vectors. The closest match is the weighted sum of the matching conditions. However these weights are not likely to be known with respect to their impact on the graph. Domain experts may at best have a subjective notion about the relative importance of a few conditions [SMMV-04].

Mathematical modeling in the given domain [M-95] could be another estimation technique. This requires a precise representation of graphs in terms of numerical equations, and exact knowledge of how the inputs affect the outputs. Since precise numerical equations and / or variables in existing models are often not known in experimental domains mathematical modeling does not achieve enough accuracy. For example, in Heat Treating, it has been shown that the existing models do not adequately work for multiphase heat transfer with nucleate boiling [SMMV-04].

Case-based reasoning (CBR) [K-93] could also be used for estimation. In our context, this involves comparing conditions to retrieve the closest matching experiment, reusing its graph as a possible estimate, performing adaptation if needed, and retaining the adapted case for future use. However adaptation approaches in the literature [K-93, L-96] are not feasible for us. For example in Heat Treating, if the condition "agitation" in the new case has a higher value than in the retrieved case, then a domain-specific adaptation rule could be used to infer that high agitation implies high heat transfer coefficients. However, this is not sufficient to plot a graph in the new case. Adaptation rules [K-93] are generally used when the case solution is categorical such as in medicine and law. Case-based adaptation [L-96] could possibly be applied here. In the example where agitation in the new case and retrieved case do not match, the case base could be searched to find another case that matches in terms of agitation. However this second retrieved case may not match another condition

such as "Quenchant temperature". The graphs of the two retrieved cases could then be used to build an average that forms the estimated graph in the new case. However, building such an average requires prior knowledge of the relative importance of conditions and the significant features on graphs. Moreover, adaptation with any approach requires a computational expense for each estimation performed, which is inefficient [V-04]. CBR approaches without adaptation such as exemplar reasoning [AP-03] and instance-based reasoning [M-97] if used in our context face the same problem as naïve and weighted similarity search respectively as elaborated in [V-04].

7. Evaluation

In order to judge the performance of AutoDomainMine, a tool has been built using this approach. This has been evaluated in the Heat Treating domain using a test set of performed experiments distinct from the training set. An example from the evaluation is presented here. The user submits input conditions of a Heat Treating experiment as shown in Figure 8 to estimate the graph obtained. The estimated graph, i.e., heat transfer coefficient curve is shown in Figure 9.

On comparing this with the graph obtained in the laboratory experiment performed with the same input conditions (as stored in the test set), the domain experts conclude that this estimation is satisfactory. Likewise on evaluating with several examples, it is found that the estimation accuracy is approximately 75%. The response time of the tool is on an average 2 seconds. Thus accuracy and efficiency are both within acceptable limits as concluded by domain experts. On comparing the AutoDomainMine estimation with that provided by naïve similarity search, it is found that AutoDomainMine is more accurate and efficient. Similarity search needs an average response time of 4 seconds and gives an accuracy of approximately 60%**.** AutoDomainMine is faster because it searches over decision trees and representatives as opposed to similarity search that searches the entire database of existing experiments. AutoDomainMine is also more accurate since it incorporates domain knowledge discovered from existing experimental results while performing the estimation.

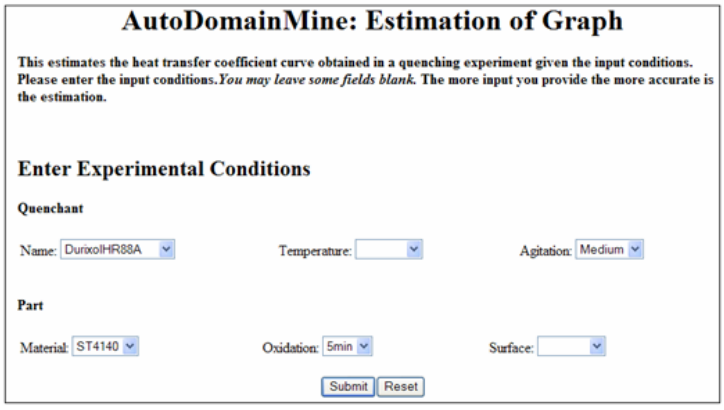

Figure 8: Given Input Conditions

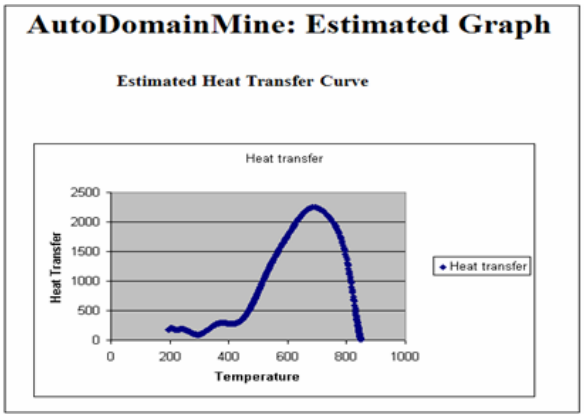

Figure 9: Estimated Graph

8. Conclusions and Ongoing Work

This paper describes the AutoDomainMine approach proposed for data mining over graphical results of experiments with domain semantics. It discovers knowledge from results of existing experiments by integrating clustering and classification in order to perform computational estimation for decision support in the given domain. AutoDomainMine is used to estimate the *graph* that would be obtained in an experiment given its *input conditions* and vice versa. It targets science and engineering domains with experimental results characterized by 2-dimensional graphs representing the behavior of process parameters. Ongoing research includes *learning* a domain-specific distance metric (rather than using Euclidean or weighted Euclidean distance) as the notion of similarity in clustering, and *designing* (as opposed to selecting) domain-specific representatives for classification. This is likely to provide even better estimation as required for decision support.

References

[AP-03] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational Issues, Methodological Variations and System Approcahes", In AICom, 2003, 7(1):39 – 59.

[AFS-93] R. Agrawal, C. Faloutsos and A. Swami, "Efficient Similarity Search in Sequence Databases", In FODO, Oct 93.

[BL-04] A. Banerjee and J. Lingford, "An Objective Evaluation Criterion for Clustering", In KDD, Aug 04.

[CDKM-98] R. Caruna, V. de Sa, M. Kearns and A. McCallum, "Integrating Supervised and Unsupervised Learning", In NIPS Workshop, Dec 98.

[GW-02] L. Gao and X. Wang, "Continually Evaluating Similarity-Based Pattern Queries on a Streaming Time Series", In SIGMOD, Jun 02.

[HK-01] J. Han and M. Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufmann Publishers, 2001.

[K-93] J. Koldner, "Case-Based Reasoning", Morgan Kaufmann Publishers, 1993.

[KR-90] L. Kaufman and P. Rouseeuw, “Finding Groups in Data: An Introduction to Cluster Analysis”, John Wiley, 1990.
[KK-95] H. Kim and G. Koehler, “Theory and Practice of Decision Tree Induction”, In Omega, 23(6):637 - 652, 1995.
[L-96] D. Leake, “Case-Based Reasoning: Experiences, Lessons and Future Directions”, AAAI Press, 1996.
[LHM-98] B. Liu, W. Hsu and Y. Ma, “Integrating Classification and Association Rule Mining”, In KDD, Aug 98.
[LSW-97] B. Lent, A. Swami and J. Widom, “Clustering Association Rules”, In KDD, Aug 97.
[M-95] A. Mills, “Heat and Mass Transfer”, Richard Irwin Inc., 1995.
[M-97] T. Mitchell, “Machine Learning”, McGraw Hill Publishers, 1997.
[PC-97] K. Pal and J. Campbell, “An Application of Rule-Based and Case-Based Reasoning within a single Legal Knowledge-Based System”, The Data Base for Advances in Information Systems, 28(4):48 – 63, 1997.
[SMMV-04] R. Sisson Jr., M. Maniruzzaman, S. Ma and A. Varde, “CHTE Fall-04 Report”, WPI, MA, USA, Nov 04.
[V-04] A. Varde, “Automating Domain-Type-Dependent Data Mining as a Computational Estimation Technique for Decision Support, Ph.D. Proposal, WPI, MA, USA, Sep 04.
[VTRWMS-04] A. Varde, M. Takahashi, E. Rundensteiner, M. Ward, M. Maniruzzaman and R. Sisson Jr., “Apriori Algorithm and Game-of-Life for Predictive Analysis in Materials Science”, In KES Journal, To Appear, 2004.
[WF-00] I. Witten and E. Frank, “Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann Publishers, 2000.