MOTC: An Interactive Aid for Multidimensional Hypothesis Generation*

K. Balachandran J. Buzydlowski G. Dworman
S.O. Kimbrough T. Shafer
W. Vachula
University of Pennsylvania
3620 Locust Walk, Suite 1300
Philadelphia, PA 19104-6366
Kimbrough@Wharton.upenn.edu
(215) 898–5133
http://grace.wharton.upenn.edu/~sok/

March 3, 1999

Note: This paper is an expanded version of "MOTC: An Aid to Multidimensional Hypothesis Generation," by K. Balachandran, J. Buzydlowski, G. Dworman, S.O. Kimbrough, E. Rosengarten, T. Shafer, and W. Vachula, in Nunamaker and Sprague, eds., *Proceedings of the Thirty-First Hawai'i International Conference on System Sciences,* IEEE Computer Press, 1998.

*Special thanks to James D. Laing for introducing us to Prediction Analysis, for encouraging noises as these ideas were developed, and for insightful comments on an earlier version of this paper. Thanks also to Balaji Padmanabhan for some useful comments and suggestions. None of the shortcomings of this paper should be attributed to either Laing or Padmanabhan. Send correspondence to Steven O. Kimbrough at the address in the heading. File: motcjmis5.tex, from motcjmis4.tex, motcjmis3.tex, motcjmis2.tex, motcjmis.tex, from MotcHICS.TEX, 19970929, from motc.tex. 970616, 970619, 970621, 970622, 970624b, 970625; 19980630, 19980701, 19980702. This material is based upon work supported by, or in part by, the U.S. Army Research Office under contract/grant number DAAH04-1-0391, and DARPA contract DASW01 97 K 0007.

Contents

List of Figures

List of Tables

Abstract

This paper reports on conceptual development in the areas of database mining, and knowledge discovery in databases (KDD). Our efforts have also led to a prototype implementation, called MOTC, for exploring hypothesis space in large and complex data sets. Our KDD conceptual development rests on two main principles. First, we use the crosstab representation for working with qualitative data. This is by now standard in OLAP (on-line analytical processing) applications and we reaffirm it with additional reasons. Second, and innovatively, we use Prediction Analysis as a measure of goodness for hypotheses. Prediction Analysis is an established statistical technique for analysis of associations among qualitative variables. It generalizes and subsumes a large number of other such measures of association, depending upon specific assumptions the user is willing to make. As such, it provides a very useful framework for exploring hypothesis space in a KDD context. The paper illustrates these points with an extensive discussion of MOTC.

Krishnamohan Balachandran is a Ph.D. candidate at the Operations and Information Management department at the Wharton School of the University of Pennsylvania. He received a B.Tech from the Indian Institute of Technology, Bombay in 1993 and an M.S. from the Systems Engineering department of the University of Pennsylvania in 1995. His current research interests are benchmarking, consumer behavior and data mining applications related to these fields.

Jan Buzydlowski holds a bachelor's degree in mathematics, and masters degrees in statistics and computer science. His interests are in data storage and analysis as well as object technologies. His current research is in data scrubbing and object-oriented data warehouse design. He has taught for 15 years and worked as a statistician for 5 years. He is currently finishing his Ph.D. in information systems at Drexel University.

Garett Dworman is a Ph.D. candidate in the Department of Operations and Information Management at the Wharton School of the University of Pennsylvania. The main theme of his research is the design of cognitively motivated information access systems. For his dissertation he is developing pattern-oriented systems for accessing document collections. This technology is currently being applied to collections in museums and the health-care industry.

Steven O. Kimbrough is a Professor at The Wharton School, University of Pennsylvania. He received his Ph.D. in philosophy from the University of Wisconsin. His main research interests are in the fields of electronic commerce, decision support and expert systems, logic modeling, and computational rationality. His active research areas include: computational approaches to belief revision and nonmonotonic reasoning, formal languages for business communication, evolutionary computation (includng genetic algorithms and genetic programming) and context-based information retrieval. He is currently co-Principal Investigator of the Logistics DSS project, which is part of DARPA's Advanced Logistics Program.

Tate Shafer has a Bachelor of Science degree from The Wharton School of the University of Pennsylvania, where he studied Information Systems and Finance. He served as an undergraduate Research Assistant to Prof. Steven O. Kimbrough, and researched various IS fields, including computer programming, artificial intelligence, and information retrieval. Additionally, he served for three semesters as a Teaching Assistant for the introductory Information Systems course at the Wharton School. He currently resides in New York City and is a strategy consultant for Oliver, Wyman & Co.

William J. Vachula is currently a Ph.D. candidate in the Operations and Information Management department at the Wharton School of the University of Pennsylvania. Mr. Vachula received a BSEE from Carnegie-Mellon University(1983) and an MSEE from the University of Pennsylvania(1989). His industry experience includes software and systems engineering, project and program management, and information systems consulting. His research interests include various application domains for software agents, computational economics techniques, and advanced systems analysis and design processes.

1 Introduction

It stands to reason that existing databases are underexploited. Organizational databases are typically created to record and facilitate business transactions. These databases often contain valuable information which fails to be recognized and used by the organizations that own and maintain them. Such, at least, is a widespread belief. This has led to a burgeoning industry of research papers, start-up firms, and professional seminars, focusing on what has come to be called KDD (knowledge discovery in databases; see [10] for a recent collection of representative papers). Real money is being bet that valuable knowledge is there to be discovered and that software innovations will help discover and exploit this knowledge economically.

We share the widespread belief in the efficacy, or at least potential, of KDD, and are exploring a concept that—we believe—addresses a central problem in KDD, viz., hypothesis generation. In what follows we describe our concept and our implementation in a prototype system called MOTC. First, however, some comments to set the context.

The premise of KDD is that software innovations can materially contribute to more effective exploitation of databases. But just how can KDD software do this and what is its relation to standard statistical methods? Put bluntly, here is a question we have heard posed by many statisticians and statistically-trained practitioners: What does KDD have to offer that isn't done well already by multiple regression techniques? Put briefly, the answer is "plenty." Standard statistical methods, including regression analysis, are hypothesis *testing* methods. For example, what regression analysis does is accept a functional form for a model/hypothesis and then find the "best" instance of a model/hypothesis of that form. Even if we were to grant that computational—e.g., KDD or AI—approaches could never improve on this basic statistical task, very much remains to be done—and to be researched—in the interests of effective KDD.

Examples of "non-statistical" issues in KDD include the following.

1. Data cleaning

   What can be done to locate and ameliorate the pervasive problems of invalid or incomplete data?

2. "First cut" analysis

   What can be done to automatically provide an initial assessment of the

patterns and potentially useful or interesting knowledge in a database? The aim here is, realistically, to automate *some* of the basic work that is now done by skilled human analysts.

3. Hypothesis generation

   What can be done to support, or even automate, the finding of plausible hypotheses in the data? Found hypotheses would, of course, need to be tested subsequently with statistical techniques, but where do you get "the contenders" in the first place?

Our attention, and the research results reported in this paper, have focused on the hypothesis generation problem for KDD. Because hypothesis space is generally quite large (more on this below), it is normally quite impossible to enumerate and investigate all the potentially interesting hypotheses. Heuristics are necessary and, it would seem, a decision support philosophy is called for. What, then, are the main requirements, or desired features, of a decision support tool for investigating hypothesis space? We identify the following as among the principal requirements. Such a tool should:

1. Support users in hypothesizing relationships and patterns among the variables in the data at hand (we call this *hypothesis hunting*).

2. Provide users with some indication of the validity, accuracy, and specificity of various hypotheses (*hypothesis evaluation*).

3. Provide effective visualizations for hypotheses, so that the powers of human visual processing can be exploited for exploring hypothesis space.

4. Support automated exploration of hypothesis space, with feedback and indicators for interactive (human-driven) exploration.

5. Support all of the above for data sets and hypotheses of reasonably high dimensionality, say between 4 and 200 dimensions, as well on large data sets (e.g., with millions of records).

What is needed, conceptually, to build such a tool?

1. A general concept or representation for data, hypotheses, and hypothesis space. This representation need not be universal, but should be broadly applicable. We call this the *hypothesis representation,* and we discuss it in §2.

2. Given a hypothesis representation, we also need an indicator of quality for the hypothesis in question. We call this the *measure of goodness,* and we discuss it in §3.

3. The hypothesis representation and the measure of goodness should fit with, cohere with, the requirements (and implicit goals, described above) of a DSS for exploring hypothesis space. We discuss our efforts and results in this regard in §§4–5.

2 Hypothesis Representation

There are three main elements to our hypothesis representation concept:

1. Focus on qualitative data.

2. Use the crosstab (aka: data cube, multidimensional data, cross classifications of multivariate data) form for data (rather than, say, the relational form as in relational databases).

3. Represent hypotheses by identifying error values in the cells of the multidimensional (crosstab) data form.

These aspects of the concept, and why we have them, are perhaps best understood through a specific example.[1] Suppose we have data on two variables: X_1, party affiliation, and X_2, support for an increased government rôle in social services. X_1 can take on the following values: *Dem, Ind,* and *Rep* (Democrat, Independent, and Republican). X_2 can have any of the following values: *left, left-center, center, right-center, right.* Suppose we have 31 observations of the two variables taken together, as follows in Table 1.[2]

Focus on qualitative data. The variables X_1 and X_2 in Table 1 are qualitative (aka: categorical) because they take on discrete values (three such values in the case of X_1 and five for X_2). X_1 is arguably a *nominal* variable because

[1] The example that follows is from [14]. We invite the reader to examine that discussion as a way of following up on this paper.

[2] We use the two-variable case for illustration only. As noted above, an important requirement for a hypothesis exploration DSS is that it handle reasonably high-dimensionality hypotheses. Except where noted—e.g., limitations of screen space in MOTC-like implementations—our points and methods generalize to arbitrarily many dimensions, at least in principle.

Support	*Party Affiliation*			
	Dem	*Ind*	*Rep*	
Left	12	3	1	16
Left-center	1	2	2	5
Center	0	3	4	7
Right-center	0	1	1	2
Right	0	0	1	1
	13	9	9	31

Table 1: Party Affiliation and Support for Social Services by Top-Level Bureaucrats in Social Service Agencies ([14, p. 11])

there is no compelling natural ordering for its three values.[3] *Dem* for example is neither more nor less than *Ind.* Similarly, in a business database *Sales-Region* and *Division* are nominal because, e.g., *Mid-Altantic* is neither more nor less than *New England* and *Marketing* is neither more nor less than *Manufacturing.* X_2 on the other hand is an *ordinal* variable because there is a natural ordering for the values it takes on: *left, left-center, center* and so on. Similarly, in a business database, *Quarter* (*first, second, third, fourth*) is naturally ordered and therefore ordinal. If a variable, e.g., *Sales,* is quantitative, then (for our framework) it will have to be quantized, or *binned.* Thus, for example, *Sales* (V_2) might be binned as follows into five categories or bins (aka: *forms* [20]):[4]

V_2^1 [0 - 20,000)

V_2^2 [20,000 - 40,000)

V_2^3 [40,000 - 60,000)

[3]Nothing much turns on this. One could argue that, at least for certain purposes, this is an ordinal variable. No matter. Our point is that this approach can handle nominal variables, if there are any.

[4]How a basically quantitative variable should be binned—including how many forms it should have—is typically determined by the investigator, although some principles for automatic binning are available [39]. It is well known that infelicitous binning can lead to anomalies and distortions. In general for a quantitative variable it is better to have more bins than fewer, in order to reduce or even eliminate loss of information. Having more bins does have increased computational cost. Neglecting computational costs, Prediction Analysis transparently accommodates arbitrarily large numbers of bins (and cells); in particular it is unaffected by the presence of crosstab cells without data instances.

V_2^4 [60,000 - 80,000)

V_2^5 [80,000 +]

By way of justification for this assumed focus, we note the following: (1) Many variables, perhaps the majority, occurring in business databases are naturally qualitative; (2) A general framework, including both qualitative and quantitative variables, is highly desirable; (3) With felicitous binning quantitative variables can typically be represented qualitatively to a degree of accuracy sufficient for exploratory purposes; and (4) Transformation of inherently qualitative variables to a quantitative scale is inherently arbitrary and is known to induce results sensitive to the transformation imposed.

Use the crosstab form for data. This aspect of our focus requires less explanation and justification, since it is also standard practice in OLAP (on-line analytical processing) applications (cf., [16, p. 179] on "the 'cube' foundation for multidimension DBMS datamarts"; [8, p. 45] on "hypercube data representations"; [27] and [7] on "cubes"). Our reasons for using the crosstab form for data representation are simple and essentially identical to why it is now used so widely in OLAP applications (and has long been essential in statistics): the crosstab form easily accommodates qualitative variables and (most importantly) it has been demonstrated to be a natural representation for the sorts of reports and hypotheses users—managers and scientists—typically are interested in.[5] (See also the literature on information visualization. For a review see [21].)

Represent hypotheses by identifying error values in the cells of the multidimensional data form. Recalling our example data, in Table 1, suppose that an investigator has a hypothesis regarding how each bureaucrat's party affiliation predicts the bureaucrat's support for increased social services. Following the notation of [14, 15], we use the statement $x \rightsquigarrow y$ to mean, roughly, "if x then predict y" or "x tends to be a sufficient condition for y."[6] Suppose

[5]We do not want to suggest that the data format evident in Table 1 is the only kind of crosstab representation for qualitative data. It isn't and the methods we discuss here, including MOTC itself, are not limited to this particular format, but further elaborating upon the point would be a diversion here. See the discussion in [14] of the condensed ordinal form for one example of an alternative crosstab representation.

[6]Or for the cognoscenti of nonmonotonic or defeasible reasoning, "if x then presumably y." But this is a subtlety we defer to another paper.

our investigator's hypothesis, or prediction (call it $\mathcal{P}_1$), is that Democrats tend to be left or left-center, Independents tend to be at the center, and Republicans tend to be center, right-center, or right. Equivalently, but more compactly, we can say:

> $\mathcal{P}_1$: *Dem* $\rightsquigarrow$ (*left* or *left-center*) and *Ind* $\rightsquigarrow$ *center* and *Rep* $\rightsquigarrow$ (*center* or *right-center* or *right*)

Equivalently, and in tabular form, we can label cells in the crosstab representation as either predicted by $\mathcal{P}_1$, in which case they receive an error value of 0, or as not predicated by $\mathcal{P}_1$, in which case they receive an error value of 1. Table 2 presents $\mathcal{P}_1$ in this form.

Support	*Party Affiliation*		
	Dem	*Ind*	*Rep*
Left	0	1	1
Left-center	0	1	1
Center	1	0	0
Right-center	1	1	0
Right	1	1	0

Table 2: Error-cell representation for the hypothesis, or prediction, $\mathcal{P}_1$.

Given that the data are to be presented in crosstab form, the error-cell representation for hypotheses is natural and, we think, quite elegant. Note as well two things. First, we can now give an operational characterization of hypothesis space. If the number of cells in a crosstab representation is C and the number of possible error values (2 in Table 2: 0 for no error and 1 for error) is n, then the number of possible hypotheses is $(n^C - n)$. (We subtract n to eliminate the cases in which all cells have the same error value. Presumably, these cannot be interesting predictions.) Thus even for our little example, $\mathcal{P}_1$ is just one of $2^{15} - 2 = 32,766$ possible hypotheses for predicting and explaining these data. Second, as we have implied in our first comment just given, it is possible to use more than 2 (0 or 1) error-cell values. Perhaps observations falling in certain cells are intermediate and should have an error value of, say, 0.5. There is nothing in these representations or in Prediction Analysis (see §3) that prevents this sort of generalization.

3 Prediction Analysis

Put briefly, Prediction Analysis [14, 15] is a well-established technique that uses the crosstab and error-cell representations of data and predictions, and also provides a measure of goodness for a prediction (on the given data). We can describe only the basic elements of Prediction Analysis here; much more thorough treatment is available in the open literature. What we find especially intriguing about Prediction Analysis—besides its intuitiveness and its fit with our preferred data representations—are two things. First, it has been shown to subsume most, if not all, standard measures of association for qualitative data, such as Cohen's Kappa, Kendall's τ, and Goodman and Kruskal's gamma (see [14, 15] for details). Second, Prediction Analysis was originally motivated to evaluate predictions *ex ante,* for example on the basis of prior theory. But it also can be used *ex post* to select propositions from the data, in which case it is, as one would expect, asympotically χ^2. Used *ex post,* Prediction Analysis is good for finding "the contenders," hypotheses that merit careful scientific investigation using standard statistical techniques.

The principal measure of hypothesis value in Prediction Analysis is ∇ (pronounced "dell"), which is defined as follows:

$$\nabla = 1 - \frac{\text{observed error}}{\text{expected error}} \tag{1}$$

Let n_{ij} be the number of observations in cell row i column j, and ω_{ij} be the error value for the cell in row i column j. (Again, although we are holding the discussion in terms of a two-dimensional example, all of this generalizes in a straightforward way.) Then, we may define the observed error for a particular prediction (error-cell table) as

$$\text{observed error} = \sum_{i=1}^{R} \sum_{j=1}^{C} \omega_{ij} \cdot n_{ij} \tag{2}$$

where the number of forms in the row variable is R and the number of forms in the column variable is C.

Finally, the expected error formula is

$$\text{expected error} = \sum_{i=1}^{R} \sum_{j=1}^{C} \omega_{ij} \cdot n_{i\bullet} \cdot n_{\bullet j}/n \tag{3}$$

where

$$\begin{aligned} n_{i\bullet} &= \text{The number of observations in category } i \text{ of the first (row) variable} \\ n_{\bullet j} &= \text{The number of observations in category } j \text{ of the second (column) variable} \\ n &= \text{The total number of observations} \end{aligned}$$

That is, $n_{i\bullet}$ and $n_{\bullet j}$ are the row and column marginals, which are presented in Table 1. Note as well:

1. If the observed error equals 0, then ∇ is 1. This is the highest possible value for ∇.

2. If the observed error equals the expected error, then ∇ is 0. This indicates, roughly, a prediction no better than chance, rather like a correlation of 0. (But remember: standard correlation coefficients apply to real numbers, quantitative variables, not qualitative variables.)

3. ∇ may be negative, arbitrarily so. A negative value is like a negative correlation, but may go lower than -1.

4. In general a higher ∇ indicates a better prediction, but this neglects considerations of parsimony. After all, if all the error cells are set to 0 then ∇ will equal 1.[7] Prediction Analysis uses what it calls the *precision,* which is the expected error rate for a prediction, $\mathcal{P}$. Precision in this sense is called U and is defined as

$$U = \sum_{i=1}^{R} \sum_{j=1}^{C} \omega_{ij} \cdot n_{i\bullet} \cdot n_{\bullet j} / (n \cdot n) \tag{4}$$

   Note that if $\omega_{ij} = 0$ for all i, j (i.e., nothing is an error), then $U = 0$ and if $\omega_{ij} = 1$ for all i, j (i.e., everything is an error), then $U = 1$.

5. In finding good hypotheses, we seek to maximize ∇. We might think of maximizing ∇ and U jointly, as in $\alpha \cdot \nabla + (1 - \alpha) \cdot U$ or in $\nabla \cdot U$;[8] or

[7] Of course if expected error is 0, the ratio is undefined.

[8] $\nabla \cdot U = U - K$ or the absolute reduction in error of the prediction. One might instead, e.g., prefer to use the relative reduction in error.

we might think of U as a constraint on this maximization problem.We might also think of imposing other constraints, such as "naturalness" conditions. For example, in the error cell representation, one might require that there should not be gaps in columns between error and non-error cells. But this is a topic beyond the scope of the present paper. For present purposes, we rely on the user's judgment to impose reasonableness criteria on hypotheses explored.

4 MOTC: A DSS for Exploring Hypothesis Space

MOTC is a prototype implementation of a DSS for exploring hypothesis space. It assumes the two main frameworks we have just discussed (crosstabulation of qualitative data for hypothesis representation, and Prediction Analysis for a measure of goodness for hypotheses) and it meets, or at least addresses, the main requirements we identified above for such a DSS. MOTC is implemented in Visual Basic 5 and Microsoft Access, and runs in a Windows NT environment.

The central, dominating metaphor in MOTC is the representation of variables (dimensions) as binned bars. A single bar corresponds to a single variable. Bars are arrayed horizontally, and are divided by vertical lines indicating bins. Each bin corresponds to a category for the variable in question. Thus, in our previous example the bar for *Party Affiliation* would have three bins, while the bar for *Support* would have five bins. A user may right-click on a bar and MOTC will present information about the underlying binning arrangement. See the figures in §5 for illustration. The width of a bin as displayed represents the percentage of records in the relevant data set that have values falling into the bin in question. Wider bins indicate proportionately larger numbers of records. MOTC as presently implemented allows up to eight variables to be represented as bars on the display. A bar may have any number of bins. This is in fact an interesting and nontrivial degree of multidimensionality (and see our discussion in §6 of the focus+context technique used by Rao and Card in their Table Lens program [32]).

MOTC as currently implemented has two modes of operation: hypothesis hunting (aka: brush) mode, and hypothesis evaluation (aka: prediction) mode. In hypothesis hunting mode, users use brushing with the mouse to dis-

play relationships among variables. Users choose particular bins and brush them with a chosen color by clicking on them. MOTC responds by applying the same color to bins associated with other variables. For example, if the user brushes bin 3 of variable 1 with purple, MOTC might respond by covering 25% of bin 2 of variable 4 in purple, indicating thereby that 25% of the records associated with bin 2 of variable 4 also are associated with bin 3 of variable 1. (See the various figures, below, for illustrations.) A user may brush more than one bin with a single color, either within or without a single variable. The effect is a logical "or" for bins within a single variable (bar) and an "and" for bins in different variables. Further, suppose purple is used to brush bins 1 and 2 of variable X, bins 4 and 5 of variable Y, and bins 7 and 8 of variable Z. Suppose further that we are in prediction mode (see below) and that we want X and Y to predict Z. Then, the equivalent representation in Prediction Analysis terminology is:

$$((X_1 \vee X_2) \wedge (Y_4 \vee Y_5)) \rightsquigarrow (Z_7 \vee Z_8)$$

MOTC presently supports up to five colors for brushing. Each color used corresponds to a separate $\rightsquigarrow$ rule in terms of Prediction Analysis. Working in brush mode, the user explores hypothesis space, with MOTC providing feedback by coloring bins in the unbrushed bars (predicted variables). The user thus gets a rough idea of where the "big hits" in the predictions lie.

In hypothesis evaluation, or prediction, mode the user brushes—clicks and colors—bins in the predictor *and* predicted variable bars. In essence, the user is interactively populating a higher-dimensional version (up to 8 dimensions in the current implementation) of an error-cell table, as in Table 2. Doing so specifies a hypothesis and MOTC responds by calculating and displaying ∇ and U for the hypothesis.

Working iteratively, the user may explore hypothesis space by switching back and forth between hypothesis hunting mode and hypothesis evaluation mode. This continues until the user reaches reflective equilibrium.

5 A Sketch of MOTC at Work

Our purpose in this section is to give the reader a sense of what it is like to work with MOTC to explore a collection of data. We shall work with a hypothetical, rather abstract example and shall use drawings, rather than original screen dumps, in our illustrations. We do this for several reasons.

Most importantly, our aim is to communicate the essential concepts associated with MOTC. We want to discuss the forest, rather than the trees. Screen dumps from, and descriptions of, MOTC are available in considerable detail elsewhere, including the open literature [1, 2], as well as Web sites (`http://grace.wharton.upenn.edu/~sok/motc` and `http://www.practicalreasoning.com/motc`). Here, our aim is to communicate in as brief a manner as possible the core ideas of how MOTC works from a user's perspective.

A user's interaction with MOTC begins with the data, which must be stored in a Microsoft Access database and must reside in a single table or query.[9] Once such a table or query exists, the user may launch MOTC, open the appropriate Access database, and select for investigation the particular table or query of interest.

Once the user identifies the data source (table or query), MOTC presents the user with a list of attribute names (from the database) for the data source. The user selects which attributes to explore and may choose up to eight.[10] For each attribute or dimension, the user must also make decisions about binning the data. MOTC will guess whether the data for a given attribute are continuous (e.g., sales in dollars) or discrete (e.g., sales regions). The user must either override or confirm this quess. MOTC will then guess how best to categorize, or bin, the data. Again, the user may override the guess and indicate how MOTC should bin the data by dimension. (On binning, see our discussion above, in §4.)

[9]How essential is Microsoft Access? In principle, MOTC could be converted easily to work with any ODBC-compliant database, but MOTC makes essential use of Microsoft-specific features, particularly crosstab queries, which are not part of standard SQL. In the future, we intend to completely reimplement MOTC in order to make it database-neutral. That will require a substantial amount of work.

[10]The limitation to eight attributes is arbitrary. We chose it to be large enough to make the point that MOTC could handle a nontrivial number of dimensions (8 is interesting), and small enough to fit conveniently on most screens. We intend to relax this in future editions. Doing this right—to allow, say, 200 attributes—will require more sophisticated screen management techniques. See our discussion in §6.

*** Place the figure about here. ***

Figure 1: Binned, Four-Dimensional Data Set Ready for Exploration in MOTC

Once these decisions are taken, MOTC presents the user with a display showing each attribute as a horizontal bar, with vertical lines indicating bins. See Figure 1. In this Figure, which is a drawn schematic of the real program, we see that there are four attributes under joint consideration. These are labelled A, B, C, and D. Attributes A and D are each binned into three categories (1, 2, and 3, call them low, medium, and high), while attributes B and C each have four bins. (The number of bins in MOTC is open-ended, but it seldom is useful to have more than 8 or 10.)

At this point, MOTC is by default in brush (or hypothesis finding) mode. The user would select a color (MOTC supports up to five colors) and begin to explore by "brushing" a bin on an attribute. Here, we will use shading and patterns instead of colors. Figure 2 shows a notional display in which the user has selected the horizontal line pattern and brushed the left-most (1, or "low") bin on attribute A.

*** Place the figure about here. ***

Figure 2: MOTC in Hypothesis Generation (Brushing) Mode

MOTC has responded by shading bins in the other three attributes.[11] These MOTC shadings should be interpreted as histograms. Remember that every observation fits into some (exactly one) bin on each dimension. Recalling our party affiliation example, if you are left-center, then there is some party affiliation you have. MOTC is for discovering interesting patterns in the distribution of observations across bins. What MOTC is telling us here is that if an observation is from bin 1 (left-most bin) of attribute A, then it will tend to be in bins 3 or 4 of attrbute B, bins 1 or 2 of attribute C, and bin 2 of attribute D. This would appear to be a significant, or at least interesting, pattern. How good is this as a hypothesis? How well does it predict?

At this point, the user is in position to state a hypothesis and have MOTC calculate its dell and precision values, from Prediction Analysis. The user

[11] There is nothing significant about brushing the A (topmost) attribute. The ordering of the attributes on the screen is arbitrary. The user can brush a bin in any of the attributes and MOTC will respond appropriately.

then switches to prediction mode, chooses a color (pattern) and clicks on the bins corresponding to the hypothesis. See Figure 3.

*** Place the figure about here. ***

Figure 3: MOTC in Prediction Mode

In the Figure, the user has clicked on bin 1 of attribute A, bins 3 and 4 of attribute B, bins 1 and 2 of attribute C, and bin 2 of attribute D. Notice that the shading completely fills each selected bin. What this display is indicating to MOTC is the error-cell representation for the hypothesis. From this display, MOTC constructs the analog of Table 2, calculates ∇ and U (dell and precision), and displays then for the user. The user is then free to continue exploring other hypotheses.

In the case at hand, it is likely that ∇ would be reasonably high (which is good), but that U (precision) would be fairly low (which is bad). Typically, the user will want to explore more complete hypotheses (what if the observation is in bin 2 of A?). The end result of this kind of exploration might produce a complete hypothesis as in Figure 4.[12] With such a hypothesis

*** Place the figure about here. ***

Figure 4: MOTC with a Complete Prediction on the Data

expressed, MOTC would then calculate ∇ and U (dell and precision), and display them for the user.

And the user, as we have said, can continue exploring in this manner until reaching reflective equilibrium.

[12]The following remarks will perhaps be useful for interpreting Figure 4, and specifically the hypothesis it represents. First, recall Figure 3, which is a simpler figure of MOTC in prediction mode. There, the hypothesis represented is, roughly, "If A is low, and B is high (bins 3 and 4), and C is low (bins 1 and 2), then D is middling (bin 2)." (We say "roughly" because the shading is really serving to determine the error–cell representation.) Call this hypothesis α. It is indicated by the horizontal shading, which is retained in Figure 4. In addition, Figure 4 contains two other hypotheses. β (indicated by vertical shading): "If A is high, and B is low and C is high, then D is low." γ (indicated by cross–hatched shading): "If A is middling, and B is in bin 2 and C is in bin 3, then D is high." In total, Figure 4 represents the conjunction of these three hypothesis: α and β and γ. This is a complete hypothesis in that every bin is associated with some hypothesis (or prediction).

6 Comparison with Alternatives

MOTC, as we have seen, assumes two main frameworks (the crosstabulation form for representing hypotheses, and Prediction Analysis for measuring goodness of hypotheses), and provides an interactive environment of some promise for discovering interesting hypotheses. Here we want to consider the question, How does MOTC, or the ideas it embodies, compare with what has appeared in the relevant literature? Two points first: (1) MOTC is nearly unique, or at least unusual, among database mining tools in using the crosstabulation form,[13] and (2) MOTC is unique in being an end-user interactive tool for supporting Prediction Analysis. For these reasons, we are less concerned in this section with establishing originality and are more focused on placing MOTC within the nexus of data visualization techniques. This serves the purposes of better understanding what MOTC is about and of pointing towards future research.

6.1 Design goals of the MOTC interface

Stepping back and looking at the larger picture, the purpose of MOTC is to help the user discover interesting patterns in data and to provide an evaluation of the predictive value of those patterns. To this end, we identified three main desiderata for MOTC's interface design.

1. Present a display that can represent a very large number of records.

   The simple fact is that modern databases are huge and we need tools for dealing with them. Of course, for purposes of pattern discovery it is always possible—even desirable—to sample from the underlying data. Even so, having the option of examing larger datasets is always a good thing, since patterns evident in large datasets may not be apparent in smaller sample sets.

2. Effectively display a large number of variables.

   It is also a simple, or brute, fact that modern databases present large numbers of dimensions, or fields, among which users have an interest in discovering patterns. To limit a user's view of the data to only a subset of the data's variables is a severe restriction on the user's ability to

[13] Thanks to Balaji Padmanabhan for this point. See also [28].

discover patterns. Unfortunately, too many variables (dimensions) in a display can quickly overwhelm a user's cognitive resources. Therefore, a second goal of MOTC's interface is to maximize the number of displayed dimensions without overwhelming the user.

3. Provide for visualization that helps users discover associations among variables.

   Passively displaying information only goes so far in helping users discover patterns in the data. To be a truly effective interface the display must actively highlight associations among variables in the data by providing users with feedback about the quality of the apparent associations.

These are general goals, goals that have attracted study outside the context of MOTC. We now briefly review and discuss this literature.

6.2 Present a display that can represent a very large number of records

It is generally accepted that people more easily process visual information than textual or numerical information. "Scanning a thousand tiny bars with your eyes requires hardly any conscious effort, unlike reading a thousand numbers, which takes a great deal of mental energy and time" [31]. Information visualization techniques can take advantage of this fact by displaying enormous amounts of information on the screen. For example, the SeeSoft system effectively displays over 15,000 lines of code on the screen [9] by representing code with pixel-thin lines that reflect the code's visual outline. InXight's "wide widgets" [31] are visual components that can be incorporated into a GUI information system to display several orders of magnitude more data than traditional display tools (e.g., spreadsheets or hierarchical trees). Wide widgets are focus+context interfaces [12, 35] which dynamically distort spatial layouts so that users can zoom in on several records or variables while the rest of the records shrink to fit within the remaining space. In this way, users can focus on several items without losing the context provided by the remaining items. One wide widget, the Table Lens, has been demonstrated with a table of baseball statistics containing 323 rows by 23 columns = 7429 cells [30]. Others include the Perspective Wall [26] and the Hyperbolic Tree

Viewer [25].[14]

Wright [40] demonstrates several applications that make use of 3D effects. One application, a financial portfolio manager displays more than 3,000 bonds on a single screen. This system uses color to indicate long and short positions, height for the bond's value, and the x and y axes to represent subportfolios and time to maturity.

Unfortunately, these techniques will fall short for very large databases, because, ultimately, we are limited to the number of pixels on the screen. Even with techniques like VisDB's pixel-oriented approach [22, 23], which displays a data record per pixel, we are still limited to the number of pixels on the screen. With today's technology, this means approximately 1024 $\times$ 1024 $\approx$ 1MB records which will not do for multi-million, gigabyte, and certainly not terrabyte-sized databases.

To present an unlimited number of records on the screen at once we need to present summaries of the data. If summaries are provided for each variable, then the only limitation is the number of variables that can be displayed regardless of the number of records in the database. The InfoCrystal [36] uses an innovative extension of Venn diagrams to visualize data summaries. MineSet's Evidence Visualizer [3] uses rows of pie charts to summarize the data. One row for each variable, one pie chart for each attribute. The pie chart represents the number of records matching the query variable's chosen value with the pie chart's value.

The approach of presenting summaries of *all* the data is strongly endorsed by Ben Shneiderman who preaches the following mantra (as he calls it) for designing visual information seeking systems:

> Overview first, zoom and filter, then details-on-demand. [34, p. 2]

To overview very large numbers of records, we must sample or summarize. MOTC represents a summarization strategy (the crosstabulation form), but there is nothing to prevent applying MOTC to sampled data.

6.3 Effectively display a large number of variables

The problem of displaying multidimensional data in an effective manner, one comprehensible to users, has been studied for some time (see [20, 21]

[14] Images of these systems can be found on InXight's home page at http://www.inxight.com/vizcontrols

for useful reviews). Perhaps the most natural and widespread approach for adding dimensions to a display is to add visual cues to an existing display. For example, the three dimensions of a 3D graph can be augmented by encoding points on the graph with color, texturing, shapes (glyphs), shading, and other such techniques. Becker [3] demonstrates the use of such techniques with the MineSet system, and various forms of these techniques are supported by contemporary data visualization software tools (e.g., Advanced Visual Systems).

This family of techniques has two important limitations. First, there are only so many visual cues that can be employed. Perhaps 5–10 variables can be represented on a 2D display using the 3 geographic dimensions, color (divided into hue, saturation and brightness), shape, size, texture, and shading. Second, and more limiting, is that humans cannot effectively process that many visual cues of this sort at once. More than a few visual cues quickly overwhelm users. Projecting multiple dimensions onto a two-dimensional plane also becomes quickly illegible. Jones [21, Chapter 14], for example, reports that 8 dimensions is too much for this technique and even 6 and 7 dimensions are difficult to comprehend.

As an example, Feiner and Beshers' Worlds Within Worlds technique [11], which plots n dimensions by successively embedding 3-dimensional coordinate systems inside one another, can theoretically display any number of dimensions on the screen. However, Jones [21, Chapter 14] points out that more than three levels (9 dimensions) is incomprehensible and even 2 levels (6 dimensions) can be difficult to assimilate.

In MOTC, we present the same visual cue for each variable (a horizontal bar on the screen, with coloring), and use secondary visual cues (position, color) to distinguish the categories associated with a variable (the bins). A popular set of techniques using this approach are graphical matrices in which rows and columns represent variables, and each cell in the matrix is a comparison of the pair of variables represented by the cell's row and column. Perhaps the most common representation of the two variables associated with a matrix cell is a scatter plot [21, 5, 4]. However, other representations are possible, such as histogram profiles [38], boxplots and sunplots [20, Chapter 5].

Unfortunately, graphical matrices only allow direct comparisons between two variables. A simpler technique is to display a row of variables. When combined with brushing (see above), variable rows allow any number of variables to be directly compared. MineSet's Evidence Visualizer [3], with its

rows of pie charts, does just this. The Influence Explorer [38] presents rows of histograms, each histogram summarizing the values of a single variable. Thus, MOTC's display approach for variables should, in future research, be assessed as a member of this category of representation. Very likely it will be possible to improve the display, but that is something to be determined by extended empirical testing, something that has yet to be done for nearly all the interesting techniques.

Even using graphical matrices of variable rows, the number of variables that can be displayed is limited to the number of rows or columns that can fit on the screen. A natural extension of this technique to use the focus+context ability of Table Lens [30] to augment the number of rows and columns displayed, thereby augmenting the number of variables. Indeed, the interface for MOTC is an elementary example of this idea: the underlying dataset can have a very large number of dimensions, among which the user picks up to eight for a particular analysis; different dimensions can be picked in different analyses. In future editions of MOTC (or MOTC-like systems), we would think that this process could be made smoother and easier and that doing so would benefit the user.

One more technique is worth noting. Inselberg's parallel coordinates system [17, 19, 18, 21] represents variables as vertical bars, and database records as "polylines" which connect each of the variables' vertical bars. Where a polyline crosses a variable's vertical bar represents that polyline's record's value for the variable. This technique allows for a very large number of variables to be displayed—as many variables as vertical lines that will fit on the screen. The drawback of this approach is that each polyline represents one record, so the technique is limited to displaying only a relatively small number of records.

6.4 Visualizing associations between variables

Visualization techniques are known to be very helpful for discovering patterns in data. This is especially so for relationships between two variables. Things are more difficult when multiple variables are involved. For this problem, MOTC's approach is of a kind that is accepted in the literature: present multiple variables and support active display of linkages among them. For example, selecting a record or range of records in one of the Influence Explorer's histograms highlights the corresponding records in the other histograms [38]. Similarly, the Lifelines system [29] displays compact medical patient histories

in which users can, say, click on a particular patient visit and immediately see related information, such as other visits by the same patient, medication, reports, prescriptions and lab tests. Visage [24, 33] presents multiple views of the same data. One window may present geographic data in map form, while another window presents the data as a histogram, and yet another presents the data in a table. Selection of a subset of data in any window, highlights the corresponding representation of the data in the other windows. Graphical matrices can by dynamically linked through brushing [4, 5] in which selecting a set of records in one scatterplot (or whatever graphical technique is used for the graphical matrix) simultaneously highlights the same records in the rest of the matrix's cells.

MOTC's use of brushing (see above) should be seen as a visualization approach of the kind explored in this literature. As with the issue of display of multiple dimensions, much further research is needed in order to find the optimal design (if there is one) of this sort.

7 Summary & Discussion

So, what have we got and how good is it? Recall that earlier we argued for a series of goals for any tool to support the hypothesis generation activity in KDD and database mining. Here, with additional comments, is that list again.

1. *Support users in hypothesizing relationships and patterns among the variables in the data at hand.* MOTC has hypothesis hunting mode, in which users may use the mouse quickly and interactively to try out and test arbitrary hypotheses, and thereby explore hypothesis space.

2. *Provide users with some indication of the validity, accuracy, and specificity of various hypotheses.* MOTC employs Prediction Analysis for this.

3. *Provide effective visualizations for hypotheses, so that the powers of human visual processing can be exploited for exploring hypothesis space.* MOTC contributes an innovation in visualization by representing multidimensional hypotheses as binned bars that can be brushed with a mouse. Also, MOTC innovates by tying together hypothesis hunting and evaluation, and does so with a common visual representation.

4. *Support automated exploration of hypothesis space, with feedback and indicators for interactive (human-driven) exploration.* MOTC does not do this at present, although we have plans to add these features. Briefly, we intend to begin by using a genetic algorithm to encode and search for hypotheses (see Table 2). As in our candle-lighting work [6], we envision storing the most interesting solutions found by the genetic algorithm during its search and using these solutions as feedback to the user.

5. *Support all of the above for data sets and hypotheses of reasonably high dimensionality, say between 4 and 200 dimensions, as well as on large data sets (e.g., with millions of records).* MOTC is not computationally very sensitive to the number of underlying records. We have worked successfully with much larger data sets than those we report here. But, MOTC is sensitive to the number of cells in the crosstab grid. With 10 variables and 10 bins per variable, the mulidimensional data grid has 10^{10} cells, a number perhaps too large for practical purposes. On the other hand, 12 variables with only 4 bins each is only $4^{12} \approx 16$ million cells, and this is quite manageable on today's PCs. In short, MOTC-like systems will work over a wide range of useful and computationally feasible problems.

All of this, we think, looks very good and very promising. Still, the ultimate value of any system like MOTC has to be determined by testing real people on real problems. Our experience to date, which is admittedly anecdotal, is very encouraging. Moreover, we note that if you value Prediction Analysis, then you need to calculate ∇, U and so on. MOTC makes these calculations and does them quickly and easily from a user's point of view. All this is excellent reason to proceed to experiments with real people and real problems. But that is subject for another paper.

References

[1] K. Balachandran, J. Buzydlowski, G. Dworman, S.O. Kimbrough, T. Shafer, and W. Vachula. MOTC: An aid to multidimensional hypothesis generation. In Jay F. Nunamaker, Jr. and Ralph H. Sprague, Jr., editors, *Proceedings of the Thirtieth Hawaii International Conference*

on System Sciences, Los Alamitos, CA, 1998. IEEE Computer Society Press.

[2] K. Balachandran, J. Buzydlowski, G. Dworman, S.O. Kimbrough, T. Shafer, and W. Vachula. Examples of MOTC at work for knowledge discovery in data. *Communications of AIS*, forthcoming 1999. Also available at
`http://www.practicalreasoning.com/motc/`.

[3] Barry G. Becker. Using mineset for knowledge discovery. In *IEEE Computer Graphics and Applications* [13], pages 75–78. Special Edition on Information Visualization.

[4] R.A. Becker and W.S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[5] R.A. Becker, P.J. Huber, W.S. Cleveland, and A.R. Wilks. Dynamic graphics for data analysis. *Statistical Science*, 2(4):355–395, 1987.

[6] Bill Branley, Russell Fradin, Steven O. Kimbrough, and Tate Shafer. On heuristic mapping of decision surfaces for post-evaluation analysis. In Jay F. Nunamaker, Jr. and Ralph H. Sprague, Jr., editors, *Proceedings of the Thirtieth Hawaii International Conference on System Sciences*, Los Alamitos, CA, 1997. IEEE Computer Society Press.

[7] E. F. Codd, S. B. Codd, and C. T. Salley. Beyond decision support. *Computerworld*, 27(30), July 26, 1993.

[8] Vasant Dhar and Roger Stein. *Seven Methods for Transforming Corporate Data into Business Intelligence*. Prentice-Hall, Inc., Upper Saddle River, NJ, 1997. ISBN: 0-13-282006-4.

[9] S.B. Eick, J.L. Steffen, and E.E.Sumner. Seesoft — a tool for visualizing line oriented software. *IEEE Transactions on Software Engineering*, 18(11):957–968, 1992.

[10] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. The MIT Press, Cambridge, MA, 1996. ISBN: 0-262-56097-6.

[11] S. Feiner and C. Beshers. Worlds within worlds: Metaphors for exploring n-dimenstional virtual worlds. In *Proceedings of the ACM Symposium on User Interface Software 1990*, pages 76–83, New York City, 1990. ACM Press.

[12] George W. Furnas. Generalized fisheye views. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 16–23, New York City, 1986. Association of Computing Machinery (ACM).

[13] Nahum Gershon and Stephen G. Eick. Information visualization. *IEEE Computer Graphics and Applications*, pages 29–78, July/August 1997. Special Edition on Information Visualization.

[14] David K. Hildebrand, James D. Laing, and Howard Rosenthal. *Analysis of Ordinal Data*, volume 8 of *Quantitative Applications in the Social Sciences*. Sage Publications, Newbury Park, CA, 1977. ISBN: 0-8039-0795-8.

[15] David K. Hildebrand, James D. Laing, and Howard Rosenthal. *Prediction Analysis of Cross Classifications*. John Wiley & Sons, Inc., New York, NY, 1977. ISBN: 0-471-39575-7.

[16] W. H. Inmon. *Building the Data Warehouse*. Wiley Computer Publishing. John Wiley & Sons, Inc., New York, NY, 2nd edition, 1996. ISBN: 0-471-14161-5.

[17] A. Inselberg. The plane with parallel co-ordinates. *The Visual Computer*, 1:69–91, 1985.

[18] A. Inselberg and B. Dimsdale. Multidimensional lines: Proximity and applications. *SIAM Journal of Applied Mathematics*, 54:578–596, April 1994. Part 2 of 2-part series.

[19] A. Inselberg and B. Dimsdale. Multidimensional lines: Representation. *SIAM Journal of Applied Mathematics*, 54:559–577, April 1994. Part 1 of 2-part series.

[20] Michel Jambu. *Exploratory and Multivariate Data Analysis*. Statistical Modeling and Decision Science. Academic Press, Inc., San Diego, CA, 1991. ISBN: 0-12-380090-X.

[21] Christopher V. Jones. *Visualization and Optimization*. Operations Research/Computer Science Interfaces. Kluwer Academic Publishers, Boston, MA, 1995. ISBN: 0-7923-9672-3.

[22] Daniel A. Keim. Pixel-oriented visualization techniques for exploring very large databases. *Journal of Computational and Graphical Statistics*, 5(1):58–77, March 1996.

[23] Daniel A. Keim and Hans-Peter Kriegel. VisDB : Database exploration using multidimensional visualization. *IEEE Computer Graphics and Applications*, 14:40–49, Sept 1994.

[24] John Kolojejchick, Steven F. Roth, and Peter Lucas. Information appliances and tools in visage. In *IEEE Computer Graphics and Applications* [13], pages 32–41. Special Edition on Information Visualization.

[25] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In Irvin R. Katz, Robert Mack, and Linn Marks, editors, *ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 401–408, Denver, CO, 1995. Association of Computing Machinery (ACM).

[26] Jock D. Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall: Detail and context smoothly integrated. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 173–179, New Orleans, 1991. Association of Computing Machinery (ACM).

[27] David Menninger. Oracle OLAP products: Adding value to the data warehouse. An Oracle White Paper, Part #: C10281, September 1995.

[28] Andrew Moore and Mary Soon Lee. Cached sufficient statistics for efficient machine learning with large datasets. *Journal of Artificial Intelligence Research*, 8(3):67–91, 1998.

[29] Catherine Plaisant, Anne Rose, Brett Milash, Seth Widoff, and Ben Shneiderman. Lifelines: Visualizing personal histories. In Tauber [37], pages 221–227 (color plate on 518).

[30] R. Rao and S.K. Card. Exploring large tables with the table lens. In Irvin R. Katz, Robert Mack, and Linn Marks, editors, *ACM SIGCHI Conference on Human Factors in Computing Systems — Conference*

Proceedings Companion, pages 403–404, Denver, CO, 1995. Association of Computing Machinery (ACM).

[31] Ramana Rao. From research to real world with Z-GUI. In *IEEE Computer Graphics and Applications* [13], pages 71–73. Special Edition on Information Visualization.

[32] Ramana Rao and Stuart K. Card. The Table Lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proceedings of the CHI '94 Conference*, pages 318–323, 1994.

[33] Steven F. Roth, Peter Lucas, Jeffrey A. Senn, Cristina C. Gomberg, Michael B. Burks, Philip J. Stroffolino, John A. Kolojejchick, and Carolyn Dunmire. Visage: A user interface environment for exploring information. In *IEEE Conference on Information Visualization*, pages 3–12, San Francisco, October 1996. IEEE Computer Press.

[34] Ben Shneiderman. The eyes have it: A task by data type taxonomy of information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages 1996*, pages 336–343, Los Alamos, CA, September 1996. IEEE Publications. An active hyperlinked version of Shneiderman's taxonomy is available at the OLIVE site http://otal.umd.edu/Olive.

[35] Robert Spence and Mark Apperley. Data base navigation: An office environment for the professional. *Behaviour & Information Technology*, 1(1):43–54, 1982.

[36] Anselm Spoerri. Infocrystal: A visual tool for information retrieval & management. In *Conference on Information Knowledge and Management '93*, pages 11–20, Washington, D.C., November 1993. Association of Computing Machinery (ACM).

[37] Michael J. Tauber, editor. *ACM SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, BC, 1996. Association of Computing Machinery (ACM).

[38] L.A. Tweedie, R. Spence, H. Dawkes, and H Su. Externalising abstract mathematical models. In Tauber [37], pages 406–412.

[39] M. P. Wand. Data-based choice of histogram bin width. *The American Statistician*, 51(1):59–64, 1997.

[40] William Wright. Business visualization applications. In *IEEE Computer Graphics and Applications* [13], pages 66–70. Special Edition on Information Visualization.

Figure 1: Binned, Four-Dimensional Data Set Ready for Exploration in MOTC

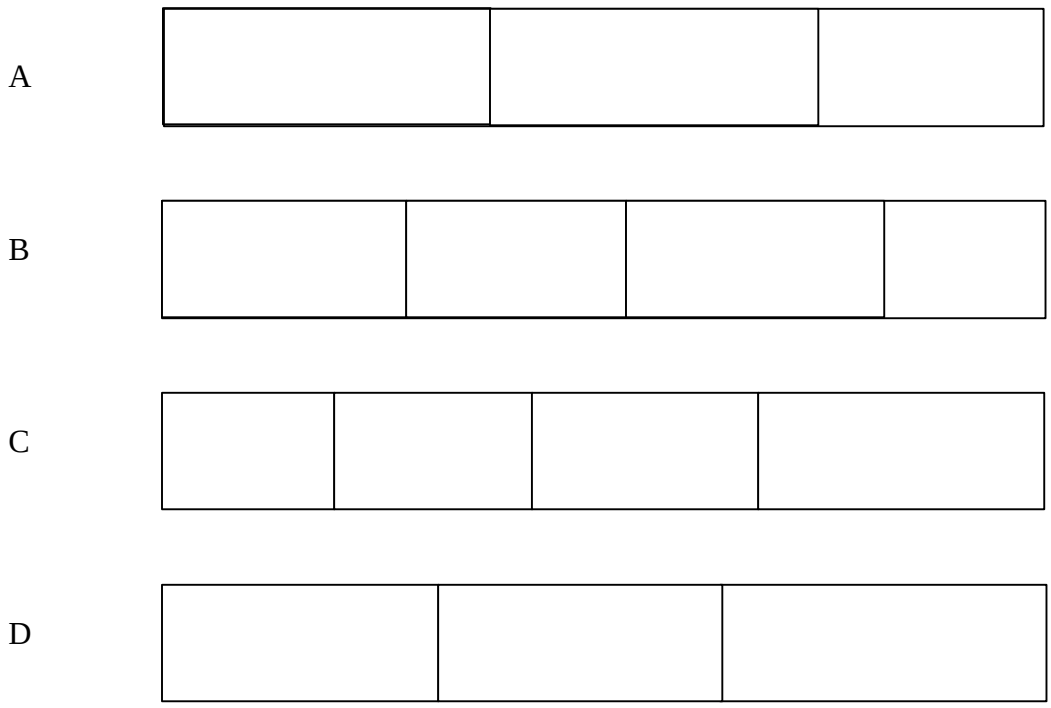

Figure 2: MOTC in Hypothesis Generation Mode

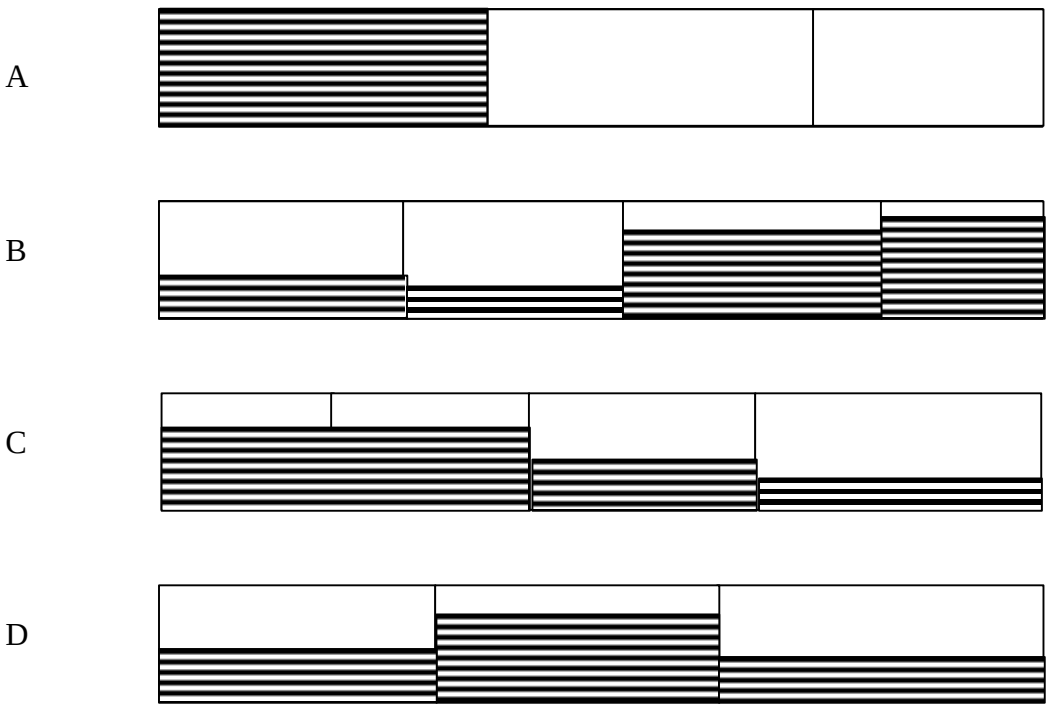

Figure 3: MOTC in Prediction Mode

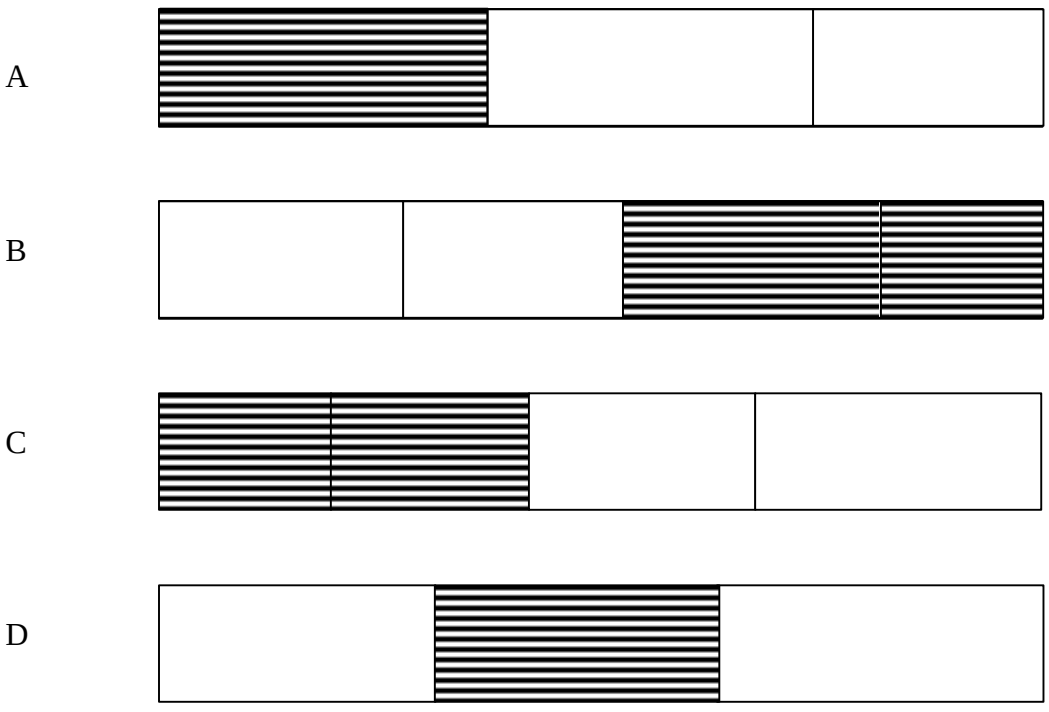

Figure 4: MOTC with a Complete Prediction on the Data