ELSEVIER

Interacting with Computers 11 (1998) 137–146

Interacting with Computers

The Attribute Explorer: information synthesis via exploration

Robert Spence*, Lisa Tweedie

Imperial College, London, UK

Accepted 12 February 1998

Abstract

The Attribute Explorer is a visualization tool in which the graphical and interactive presentation of data supports the human acquisition of insight into that data. The underlying concept employed is that of interactive linked histograms. The advantage of the Attribute Explorer derives from its ability to support both qualitative exploration and quantitative design decisions, as well as a smooth transition between these two activities.

Keywords: Attribute Explorer; Qualitative exploration; Quantitative decision making

1. Data and information

1.1. The traditional approach

The conventional approach to so-called 'information retrieval' appears to be based upon a number of tacit assumptions. It is assumed, for example, that the user is generally familiar with the relevant database(s) and therefore has no need to gain such familiarity. It is also assumed that the user will come to the database with a precisely formulated question, and that the answer will usually provide a satisfactory solution. There will therefore be little need to formulate a revised question. Furthermore, it is usually assumed that the user has been trained to access the database.

1.2. Information synthesis

Like many others, we take the view that the traditional approach to 'information retrieval' is relevant only to an extremely small fraction of 'real world' problems. In most cases, the user—whether novice or professional—may not be familiar with the

* Corresponding author.

0953-5438/98/$ - see front matter
PII: S0953-5438(98)00022-8

database, and will need to explore its contents to gain that overall insight needed to formulate a useful question (''X seems to trade-off against Y''; ''Z seems to be bimodally related to W except for…''). Having gained such an overview, the user will in all probability revise the question. Indeed, we suggest that problem formulation is a major activity and takes place concurrently with, and is an indispensable component of, problem solution. Indeed, a principal objective of visualization tools such as the Attribute Explorer is to inform the user in such a way that the problem formulation can be refined. We therefore prefer to speak of 'information synthesis' rather than information retrieval, since the latter term might be assumed to imply that information can be instantly retrieved, whereas we believe that, through exploration, the user is gradually informed. A similar philosophy appears to be implicit in the work of many others, e.g. [1–3].

1.3. Exploration

Our thesis can be illustrated by the task of home finding. The prospective buyer will enter an estate agent's office with only an imprecisely formulated need (''not more than £60K, hopefully more than two bedrooms and quite close to a railway station'') and with little or no knowledge of available houses. The first need here is to explore the available houses to gain insight into general characteristics (''prices increase steeply near that park''; ''there's no chance of a third bedroom at £60K''), so that a more focussed question can be posed. Exploration of this kind will eventually lead to the identification of a house deemed acceptable, though it may not satisfy many or even any of the original subjectively expressed requirements. Similar remarks apply to a wide variety of tasks, for example the problem faced by a marketing manager within an investment house in deciding upon a new marketing strategy on the basis of accumulated data.

1.4. The Attribute Explorer

Our consideration of information synthesis has led to the invention of a new tool called the Attribute Explorer. The task to which it is directed, and which has been described by the example above, can be stated more formally as

> Given a collection of **objects**, each described by the values associated with a set of **attributes**, find the most acceptable such object or, perhaps, a small number of candidate objects suited to more detailed consideration.

As already pointed out, the objects may be as different as houses and marketing schemes. Similarly, the definition of 'most acceptable' is purposely vague because the refinement of a problem is an essential part of the process of choosing one object from many on the basis of its attributes. The Attribute Explorer [4] is so named because that refinement is achieved primarily by exploration.

1.5. Visualization and externalization

The Attribute Explorer is characterized by its graphical representation and presentation of data, and by the nature of the interaction it supports. It is the interactive and graphical

exploration of data that, in our opinion, facilitates the user's acquisition of insight into that data. In this context, we are reminded of a Chinese proverb to which we have added the last line:

> I hear and I forget, I see and I remember, I interact and I understand, *I interact* ***responsively*** *and I discover*.

where, by 'responsively', we mean that an effect is perceived in less than about 0.1 s following its cause.

A key feature of the Attribute Explorer is that it forms an external representation of the user's problem. Constraints relevant to the problem are hard-coded into the tool's functionality so that the user does not need to focus on remembering those constraints. In this way, the rules of the problem are made implicit and the user can concentrate directly on the solution of their problem [5]. External representations of this kind are useful because they also serve as a constant reminder of the current state of the solution without recourse to memory.

To establish the salient aspects of the Attribute Explorer, we first briefly examine a traditional information retrieval scheme to identify its shortcomings. We then describe a tool (Dynamic Queries) which, in 1992, heralded a new approach to information synthesis and which stimulated the invention of the Attribute Explorer.

2. Conventional database queries

Continuing with the home-finding example, a first and major drawback associated with conventional 'information retrieval' interfaces such as those using SQL is that the user, who may well be a lay person, has to learn a language sufficiently well to be able, for example, to enter

> **select** house address
> **from** my database
> **where** price $< =$ 100 000 **and**
> bathrooms $=$ 2 **and**
> bedrooms $> =$ 3

to indicate an interest in finding a house costing no more than £100K with two bathrooms and no less than three bedrooms. A second drawback is that errors in formulating the query are usually not tolerated, with devastating consequences for the novice user. A third drawback is encountered if, as often happens, no items satisfying the query are found:

> 0 HITS.

Equally unhelpful is the situation in which too many items are identified:

> 10 573 HITS.

In both of these cases, but also in many other situations, a fourth drawback becomes apparent: no guidance is offered as to how the query may usefully be modified to obtain not only a manageable number of hits, but 'movement' towards a useful problem

statement. A fifth, and very serious, drawback is that no data are presented unless specifically requested: data are being unnecessarily hidden, with a consequent loss of contextual information. A sixth drawback is that it is exceedingly difficult for the user to synthesize knowledge of the contents of the database, and to discover such global characteristics as trade-offs (e.g., between price and distance to station) and correlations (e.g., between price and number of bedrooms). Thus, six drawbacks associated with conventional database query interfaces have to be addressed:

- The discretionary user must learn a language.
- Errors are not tolerated.
- Two few or too many hits may be registered.
- No guidance is given regarding potentially beneficial modification to the query.
- Useful contextual data are often unnecessarily hidden.
- It is difficult for a user to build an internal model.

As already stated, most of these drawbacks can be traced to a fundamental—and, in the vast majority of situations, unrealistic—assumption that ‘a user will typically know precisely what single question needs to be answered’ and, additionally, ‘knows a great deal about the content of the database’. Our rejection of this assumption is fundamental to the developments we report below.

3. Dynamic queries

Perhaps the most significant recent change in ‘information retrieval’ attitudes was evidenced by the concept of Dynamic Queries, first reported by Williamson and Shneiderman [6]. An effective illustration (Fig. 1) is again provided by a lay person’s search for a home. To the right of a map are a number of scales with adjustable sliding limits, each associated with a particular attribute: ‘house price’, ‘number of bedrooms’ and ‘journey time’ to work are typical examples. Adjustment of the upper and/or lower acceptable limits to any of these attributes causes the corresponding selection to be made from all houses in the database, and the result of that selection immediately displayed by dots on the map. A significant advantage is that responsive interaction—by which we mean that an effect occurs within less than about 0.1 s of its cause—not only allows answers to be obtained rapidly but, most importantly, supports dynamic exploration, often called the ‘What if?’ activity. Even though no direct indication of how a query might usefully be modified is given, some indication can, to some extent, now be obtained by manual exploratory variation of the attribute limits.

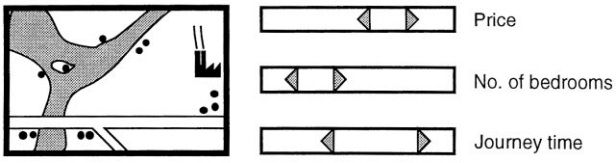

Fig. 1. The dynamic HomeFinder.

4. Disclosure of data

A major drawback to the dynamic queries concept as implemented by Williamson and Shneiderman [6] is that data are disclosed only when they satisfy the query. Especially when attention is—albeit temporarily—focussed on a small fraction of the available data, few data are available to provide **context** and **guidance** for later exploration. We therefore suggested that 'all available data can and usually should be presented'. A convenient presentation could take the form of a histogram associated with each of the attribute scales. Since such a visualization tool is particularly valuable for exploring objects characterized by a number of attributes, the name Attribute Explorer was given to the tool that eventually emerged.

5. The Attribute Explorer

Fig. 2 shows the essential characteristics of the Attribute Explorer: the distribution of all objects (houses) on all attribute scales is represented by a histogram. Again, the example of house buying is adopted for illustration, and for simplicity only three attributes are considered. Attributes need not be continuous: they can be ordinal, categorical and/or discrete.

Each object is represented once—by a rectangle of suitable size—within each of the attribute histograms. A mouse-click on such a rectangle identifies the value of that object's attributes on all the attribute scales, as shown in Fig. 2. An immediate advantage deriving from the display of 'all' the data is an awareness of the attribute ranges and the relative availability of objects with respect to attribute values: a further—and considerable—advantage, soon to be apparent, is the provision of context for specific queries.

5.1. Limits

With each attribute histogram is associated a scale containing two adjustable limits. These limits can be independently adjusted to specify the minimum and maximum acceptable values of an attribute. Alternatively, the bar linking these limits can be moved—taking the limits with it—to study the effect of moving a 'range' of attribute values. However the lower and upper limits are placed, the effect is the identification of all objects having attribute values between those limits. The objects so identified are then

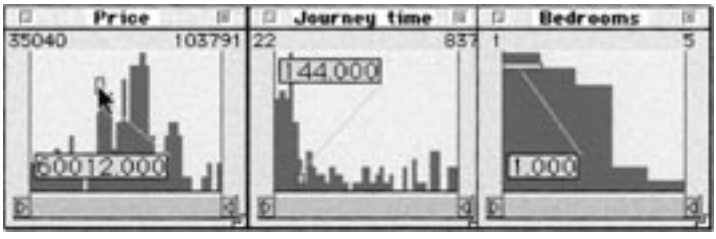

Fig. 2. The Attribute Explorer, showing the distribution over three attributes.

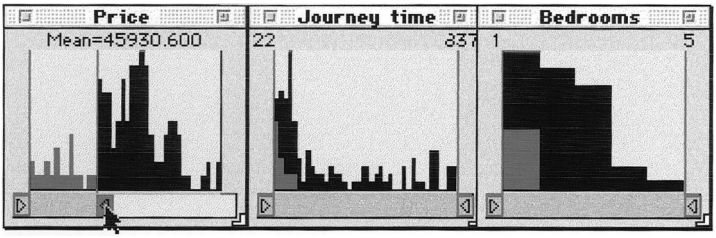

Fig. 3. Selection of a 'Price' range is reflected by colour coding into the remaining histograms.

marked by colour coding in the relevant section of the histogram, as shown for the 'Price' attribute in Fig. 3.

5.2. Attribute interaction

It is a simple matter, as illustrated by the 'number of bedrooms' and 'journey time' histograms in Fig. 3, to arrange that the same set of objects be colour-coded 'on all other attribute histograms', thereby providing an immensely valuable indication of the inter-relations between attributes. Manual variation, either of a limit or of a range defined by limits, leads to a corresponding alteration of the colour coding of the attribute histograms, additionally enhancing the opportunity to explore the inter-relation between attributes. Thus, manual adjustment of the 'Price' range from its lowest to its highest extent may well be seen to cause the 'bunch' of green houses in the 'journey time' histogram to 'move' in the opposite direction, suggesting a trade-off between these two attributes. To aid perception of such an effect, all colour-coded sections of the histogram are 'lower justified'. The interaction between histograms is a special form of the 'brushing' technique first introduced by Newton [7] and later quite widely exploited [8].

5.3. Summary information

It is not always easy, during exploration, to interpret a 'dynamic' alteration of the colour of the separate elements of a histogram. To aid in such interpretation, summary information is available in the form of a yellow circle (Fig. 4) positioned on the scale at the average of the attribute values of the identified set of objects. Now, when the range of 'Price' (Fig. 3) is quickly moved up and down the corresponding scale, the yellow 'average'

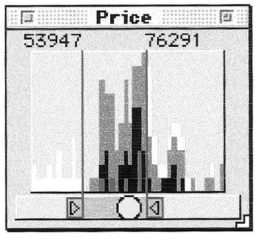

Fig. 4. The round dot indicates the average value for the selected houses.

circles on the ‘number of bedrooms’ and ‘journey time’ scales move correspondingly, allowing the relation between these attributes to be more readily discerned.

5.4. Multiple limits

As happens in the Dynamic Query display (Fig. 1), it is possible to arrange for a set of objects to be identified by limits placed on more than one attribute, and for the objects so identified to be colour-coded within each histogram, as shown in Fig. 5. Stated formally, the object selection shown is the result of performing the Boolean AND operation on the separate attribute ranges. Such a facility is particularly helpful when trying to identify one object—or a small number of candidate objects—to satisfy a particular need. The figure also demonstrates, by the inclusion of the map that would be expected of a house-finding tool, that attribute representations can be two-dimensional, in this case permitting the specification of a desirable geographical location.

5.5. Additive encoding: sensitivity information

A major disadvantage of both the Dynamic Queries interface (Fig. 1) and the example of an SQL interaction is the absence of any direct guidance as to how a ‘query’ may be modified to lead to more useful information, a need particularly acute in the ‘all-or-nothing’ situation of too many or too few hits. Such a disadvantage can be markedly reduced by additive encoding, in which colour coding is applied not only to those objects satisfying all attribute limits but, additionally and separately, to those failing one, two, three and more such limits. Thus, in the illustration of Fig. 5, and in addition to the colour-coded houses satisfying all limits, black indicates those that fail one limit, dark grey those that fail two limits, and so on. It is now apparent how a limit may usefully be changed: a limit extended to include a black object will cause it to turn green. Even for a query which identifies no acceptable (green) objects, Fig. 6 shows how limits might be relaxed to discover an acceptable house. Such information may be called ‘sensitivity information’, since colour indicates the reduction in the number of violated limits achieved by movement of a limit. This technique has much wider application than to the Attribute Explorer: it could be applied, for example, to Chalmers et al.’s [2] BEAD representation.

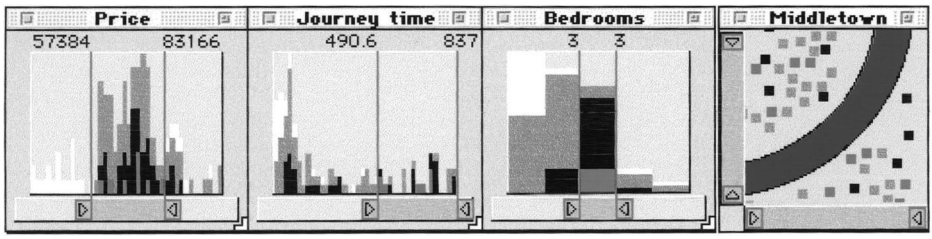

Fig. 5. Grey-scale coding shows how many limits are violated.

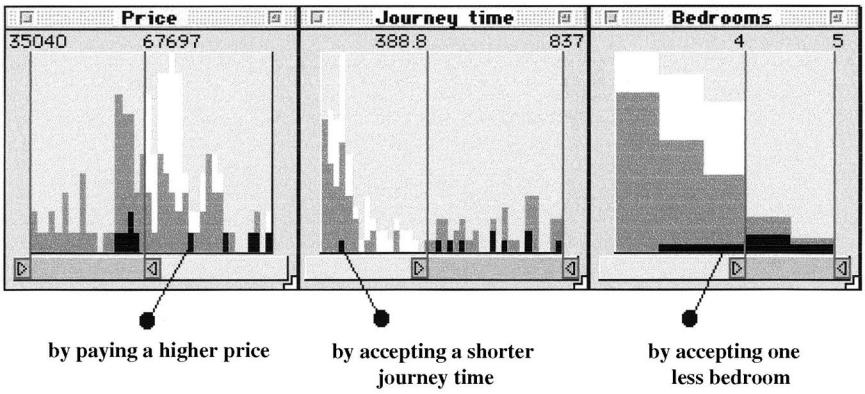

Fig. 6. In this case no houses satisfy the query, but sensitivity information indicates how limits could be altered to satisfy the query.

5.6. Previous work

Certain individual aspects of the Attribute Explorer are closely related to existing published work. The encoding of objects selected according to one attribute (e.g., price) in a display associated with another attribute (e.g., 'journey time', as in Fig. 3) is termed 'brushing' and was first introduced by Newton [7] in 1978. It is found in the Data Desk [9] visualization package. Eick [10], in 1994, reported his data visualization sliders combining a histogram with a slider mechanism, but did not explore the use of sliders as filtering mechanisms or for activating Boolean operations.

5.7. User control of link logic

The Attribute Explorer's facilitation of the Boolean AND function (leading, for example, to identification of the green houses in Fig. 5) prompts the question as to whether other Boolean operations on the attributes can conveniently be supported. Similarly, the valuable sensitivity information provided by colour coding could usefully be generalized. Both these potential enhancements to the Attribute Explorer can be provided via a single tool called the Link Crystal [11].

The Link Crystal was inspired by Spoerri's InfoCrystal [12] and, for a three-attribute situation, can be illustrated by derivation from a three-variable Venn diagram, as shown in

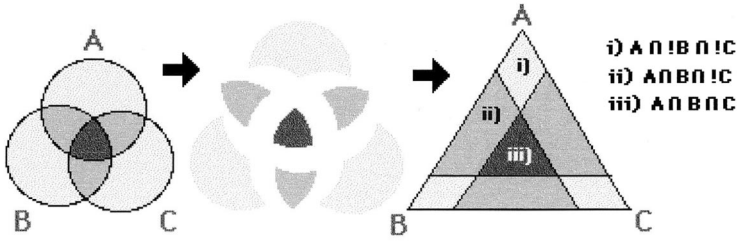

Fig. 7. Development of an InfoCrystal.

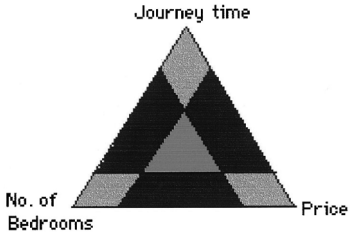

Fig. 8. A Boolean Link Crystal.

Fig. 7. Sub-regions of the Link Crystal correspond to different Boolean functions of the three attributes, and colour can be assigned to the sub-regions to control the colour coding of the displayed histograms. Thus, the colour coding shown in Fig. 5 corresponds to the Link Crystal shown in Fig. 8. If, on the other hand, interest is confined to houses that satisfy the price limits OR the limits on the number of bedrooms, the Link Crystal of Fig. 9 is relevant.

Selection of the colour associated with a sub-region of the Link Crystal is by mouse-click on that region, successive clicks cycling the colour through the available range. The concept of the Link Crystal, like that of the InfoCrystal, generalizes to any number of attributes. Whether or not a Link Crystal will be made available to the user will, of course, depend upon the nature of the user and the task for which the Attribute Explorer is intended.

6. Conclusions

Extensive discussions with a wide range of users have established that the features integrated within the Attribute Explorer support two major concurrent activities. One is **qualitative exploration** of the kind that allows insight to be gained into a body of data and which, for example, supports the discovery of new knowledge such as trade-offs and correlations. The other activity is that of **quantitative decision making** in which, as more detailed insight is acquired, small and usually iterative adjustments are made to attribute limits in order to identify an ‘optimum’ object or, perhaps, a small set of candidate objects worthy of more detailed study. Such a blend of qualitative exploration and quantitative decision making, especially with the smooth transition that can occur between

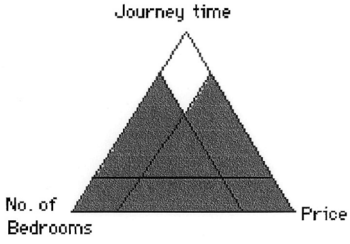

Fig. 9. A Link Crystal showing ‘price OR number of bedrooms’.

these two activities in the Attribute Explorer, is conducive to the activity of information synthesis.

Apart from its generally favourable reception, no formal evaluation of the Attribute Explorer has been performed. Such an evaluation may be useful when systems employing this and other concepts have matured sufficiently.

The concept of linked interactive data histograms has also been applied to the field of design in the form of the Influence Explorer [13].

Acknowledgements

We gratefully acknowledge the valuable collaboration of John Nelder, Huw Dawkes and Zahid Malik. Lisa Tweedie was supported in her research by an EPSRC Research Studentship.

References

[1] C. Ahlberg, C. Williamson, B. Shneiderman, Dynamic queries for information exploration: an implementation and evaluation, ACM, Proceedings CHI'92, 1992, pp. 619–626.
[2] M. Chalmers, R. Ingram, C. Pfranger, Adding imageability feature to information displays, ACM, Proceedings UIST'96, 1996.
[3] M. Hearst, Tile bars: visualization of term distribution information in full text information access, ACM, Proceedings CHI'95, 1995, pp. 59–66.
[4] L. Tweedie, R. Spence, D. Williams, R. Bhogal, The Attribute Explorer, ACM, Video Proceedings CHI'94, 1994.
[5] J. Zhang, D.A. Norman, Representations in distributed cognitive tasks, Cognitive Science 18, 1994, 87–122.
[6] C. Williamson, B. Shneiderman, The Dynamic Homefinder: evaluating dynamic queries in a real estate information exploration system, ACM, Proceedings SIGIR'92, 1992, pp. 339–346.
[7] C.M. Newton, Graphics: from alpha to omega in data analysis, in: P.C.C. Wand (Ed.), Graphical Representation of Multivariate Data, Academic Press, New York, 1978, pp. 59–92.
[8] W.S. Cleveland, M.E. McGill, Dynamic Graphics for Statistics, W. Cole, Belmont, CA, 1988.
[9] P. Vellman, The DataDesk Manual, Data Description Inc., Ithaca, NY, 1985.
[10] S.G. Eick, Data visualization sliders, ACM, Proceedings UIST'94, 1994, pp. 119–120.
[11] L. Tweedie, Exploiting interactivity in graphical problem solving: from visual cues to insight, PhD thesis, University of London, 1997.
[12] A. Spoerri, InfoCrystal: a visual tool for information retrieval, IEEE, Proceedings Visualization'93, 1993, pp. 150–157.
[13] L. Tweedie, R. Spence, H. Dawkes, H. Su, Externalizing abstract mathematical models, ACM, Proceedings CHI'96, 1996, pp. 406–412.