Supporting serendipity: using ambient intelligence to augment user exploration for data mining and web browsing

Russell Beale

Advanced Interaction Group
School of Computer Science
University of Birmingham
Edgbaston
Birmingham
B15 2TT
e: R.Beale@cs.bham.ac.uk
t: +44 121 414 3729

Abstract

Serendipity is the making of fortunate discoveries by accident, and is one of the cornerstones of scientific progress. In today's world of digital data and media, there is now a vast quantity of material that we could potentially encounter, and so there is an increased opportunity of being able to discover interesting things. However, the availability of material does not imply that we will be able to actually find it; the sheer quantity of data mitigates against us being able to discover the interesting nuggets.

This paper explores approaches we have taken to support users in their search for interesting and relevant information. The primary concept is the principle that it is more useful to augment user skills in information foraging than it is to try and replace them. We have taken a variety of artificial intelligence, statistical, and visualisation techniques, and combined them with careful design approaches to provide supportive systems that monitor user actions, garner additional information from their surrounding environment and use this enhanced understanding to offer supplemental information that aids the user in their interaction with the system.

We present two different systems that have been designed and developed according to these principles. The first system is a data mining system that allows interactive exploration of the data, allowing the user to pose different questions and understand information at different levels of detail. The second supports information foraging of a different sort, aiming to augment users browsing habits in order to help them surf the internet more effectively. Both use ambient intelligence techniques to provide a richer context for the interaction and to help guide it in more effective ways: both have the user as the focal point of the interaction, in control of an iterative exploratory process, working in indirect collaboration with the artificial intelligence components.

Each of these systems contains some important concepts of their own: the data mining system has a symbolic genetic algorithm which can be tuned in novel ways to aid knowledge discovery, and which reports results in a user-comprehensible format. The visualisation system supports high-dimensional data, dynamically organised in a three-dimensional space and grouped by similarity. The notions of similarity are

further discussed in the internet browsing system, in which an approach to measuring similarity between web pages and a user's interests is presented. We present details of both systems and evaluate their effectiveness.

Introduction

In this modern world, information is collected all the time: from our shopping habits to web browsing behaviours, from the calls between businesses to the medical records of individuals, data is acquired, stored and gradually linked together. In this morass of data there are many relationships that are not down to chance, but transforming data into information is not a trivial task. Data is obtained from observation and measurement, and has little intrinsic value. But from it we can create information: theories and relationships that describe the relationships between observations. And from information we can create knowledge: high-level descriptions of what and why, explaining and understanding the fundamental data observations. The mass of data available to us allows us to potentially discover important relationships between things, but the sheer volume dictates that we need to use the number-crunching power of computers to assist us with this process. But using computers alone is not sufficient. Computers are not endowed with insight, and have little knowledge of the outside world on which to gauge whether the concepts they are examining are worthwhile or useful. We have taken as a design principle that we will achieve more useful results if we are able to support the user in exploring and linking data, using visualisations and artificial intelligence algorithms to aid insight, rather than on trying to fully automate the process. We are aiming to provide assistive intelligence to augment the user's skills, not provide artificial intelligence to replace them. Our approach therefore aims for a synergistic relationship between the user and the computer, allowing each to use their abilities to best effect.

This approach is exemplified in Figure 1.

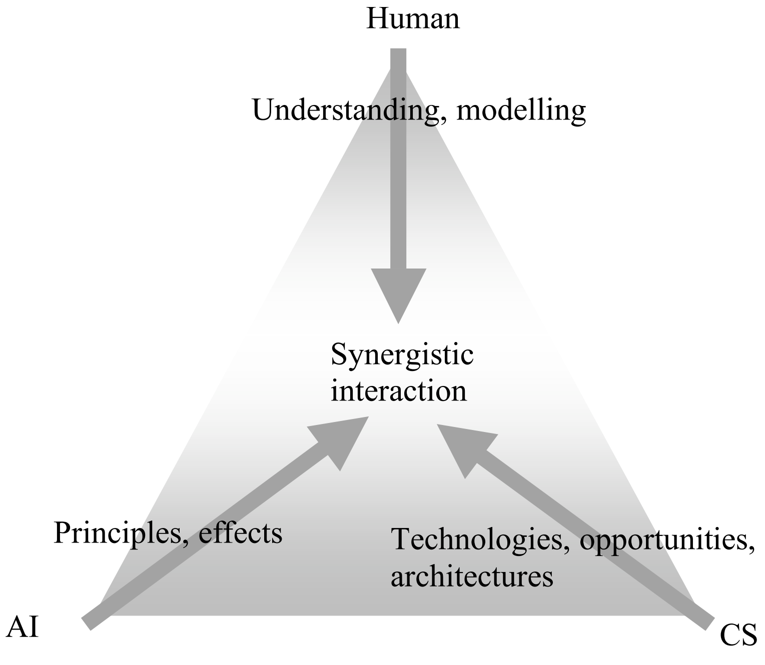


Figure 1: an understanding of artificial intelligence, computer science and users combine to give synergistic interaction

By understanding human capabilities, we can incorporate the better aspects of user skills into the systems we design, and work on supporting the things they are less capable of. Awareness of computational technologies and opportunities allows us to develop systems that utilise the best features of modern systems. These two elements are similar to the principles of socio-cognitive design (Sharples et al. 2002). We couple this with a detailed comprehension of the scope and limitations of artificial intelligence, which provide the techniques we will utilise to generate more effective approaches, and we can produce synergistic interaction. Essentially, we are using artificial intelligence approaches to reduce the distance between the user interface and the system (Abowd & Beale 1991; Norman 1988), making it more natural for the user to be able to achieve their goals.

This paper is structured around two systems, each designed according to these synergistic interaction principles. Both these examples are described and evaluated, and demonstrate the improvements that synergistic system design offers. Both systems are related in that they support the exploration and discovery of information: the first is a generic data mining system, and the second is a web browsing support system. Both these have components that support serendipitous discovery, and these are presented in detail; however, in each case it is the holistic system which offers the most significant benefits, rather than the individual advances themselves.

Interactive Data Mining

Design goals

In data mining, or knowledge discovery, we are essentially faced with a mass of data that we are trying to make sense of. We are looking for something "interesting". Quite what "interesting" means is hard to define, however - one day it is the general trend that most of the data follows that we are intrigued by - the next it is why there are a few outliers to that trend. "Interesting" is an essentially human construct, a perspective on relationships between data that is influenced by tasks, personal preferences, past experience and so on. Interest, like beauty, is in the eye of the beholder. For this reason, we cannot leave the search for knowledge to computers alone. We have to be able to guide them as to what it is we are looking for, which areas to focus their phenomenal computing power on. In order for a data mining system to be generically useful to us, it must therefore have some way in which we can indicate what is interesting and what is not, and for that to be dynamic and changeable (Ceglar, Roddick & Calder 2001). Many data mining systems do not offer this flexibility in approach: they are one-shot systems, using their inbuilt techniques to theorise and analyse data, but they address it blindly, unable to incorporate domain knowledge or insights into what is being looked for; they have only one perspective on what is interesting, and report only on data that fit such a view.

In order to provide an indication of interest, we need to provide the user with some representation of the data that they can interact with (Lee, Ong & Sodhi 1995; Nagel 2001; Shneiderman 2002). We use visualisation techniques to present an abstract representation of the data in order to achieve this (Zhang et al. 2003). The human visual system is exceptionally good at clustering, at recognising patterns and trends, even in the presence of noise and distortion (Wünsche 2004). By interacting with the raw data presented visually, the user can identify to the system the areas of interest, and focus the data mining onto exploring that part of the dataset.

Once we can ask the question appropriately, we then need to be able to understand the responses that the system gives us. The data mining system produces some information, be it classification of the data, association rules or other such information. Whilst complex statistical measures of the dataset may be accurate, if they not comprehensible to the users they do not offer insight, only description. It is desirable that a data mining system should be able to present comprehensible results, in an accessible manner (Hofmann, Siebes & Wilhelm 2002; Holmes, Donkin & Witten 1994).

An ideal data mining system should therefore, we would argue, offer the following characteristics; the ability to define what is interesting, using the abilities of the user and the computer in tasks to which they are best suited, and providing explanations of the data that are understandable and provide deep insights.

This leads us towards a system that will be interactive, in order to be flexible and iterate towards a solution. It should use visualization techniques to offer the user the opportunity to do both perceptual clustering and trend analysis, and to offer a mechanism for feeding back the results of machine-based data mining. It should have a data mining engine that is powerful, effective, and which can produce humanly comprehensible results. The Haiku system was developed with these principles in mind, and offers a synergistic system that couples interactive 3-d dynamic visualization technology with a novel genetic algorithm.

Visualization in Haiku

The visualization engine used in the Haiku system provides an abstract 3-d perspective of multi-dimensional data based on the Hyper system (Hendley et al. 1999; Wood et al. 1995) for force based visualization. The visualization consists of nodes and links (similar to a ball-and-stick model, only dynamic), whose properties are given by the parameters of the data. Data elements affect parameters such as node size, mass, link strength and elasticity, and so on. Multiple elements can affect one parameter, or a subset of parameters can be chosen.

Many forms of data can be visualised in Haiku. Typical data for data mining consists of a number of individual “items” (representing, for example, customers) each with the same number of numerical and/or nominal attributes. This is similar to standard dimension reduction methods used for solely numerical data such as Projection Pursuit (Friedman & Tukey 1974) and Multi Dimensional Scaling (Cox & Cox 1994), but applicable to data with a mix of nominal and numeric fields. What is required for Haiku visualization is that a similarity can be calculated between any two items. The similarity metric should match an intuitive view of the similarity of two items. In most cases, a simple and standard distance measure performs well.

Many forms of data can be visualisated in Haiku. Typical data for data mining consists of a number of individual "items" (representing, for example, customers) each with the same number of numerical and/or nominal attributes. What is required for Haiku visualisation is that a distance can be calculated between any two items.

The distance calculation should match an intuitive view of the differences between two items. In most cases, a simple and standard distance measure performs well: with data elements $\overline{x}_{a=}\left[x_1, x_2, \ldots x_n\right]$, distance *d* between elements $\overline{x_a}$ and $\overline{x_b}$ is

$$d = \left|\overline{x_a} - \overline{x_a}\right| = \sum_{i=1}^{n} \left|x_{ai} - x_{bi}\right|$$

An example of this is shown in Table 1:

Data item	Books	CDs	Fuel	Children	Age	sum distance
Customer 1	124.23	235.12	46.23	2	34	
Customer 2	34.56	281.46	123.09	0	29	
distance	89.67	46.34	76.86	2	5	219.87

Total distance *d* = 219.87

Table 1: Calculating distances between multidimensional data items

Clearly, many variations of this exist - a weighted sum can be used, and so on. One of the characteristics of the system is that the user can choose which parameters are used to create the distance metric, and which ones affect the other characteristics of the visualisation.

In the visualisation, a node is created that represents a data item. These nodes may be all equivalent, or may have characteristics inherited from the data (e.g. number of children may be used not in the standard distance measure, but in the mass of the node). Links are created between all the nodes, which act as springs and try to move the nodes about in the space.

To create the visualisation, nodes are initially scattered randomly into the 3d space, with their associated links. This 3d space has obeys a set of physical-type laws, which then affect this initial arrangement. Links tend to want to assume a particular length (directly related to the distance measure between the nodes), and tend to pull inwards until they reach that length, or push outwards if they are compressed, just as a spring does in the real world. Nodes tend to repel each other, based on their mass. This whole approach can be seen as a force directed graph visualisation. This initial state is allowed to evolve, and the links and nodes shuffle themselves around until they reach a local minimum, low energy steady state. The reasoning behind these choices of effects are that we want related things to be near to each other, and unrelated things to be far away. Therefore, by creating links that are attractive between data points with similar characteristics, we achieve this clumping effect. The data points themselves, the nodes in the visualisation, are made repulsive so that the system does not collapse to a point, but instead are individually distinguishable entities, slightly separated from their similar neighbours.

The physics of the space are adjustable, but are chosen so that a steady state solution can be reached that is static - this is unlike the real world, in which a steady state exists that involves motion, with one body orbiting another. This is achieved by working in a non-Newtonian space. In the real physical world with no friction (a Newtonian space) we have the following condition:

$$F = ma = m\,\frac{dv}{dt} \qquad (1)$$

where F is the force applied to a body, m the mass of that body, a the acceleration, and v is the velocity of the object.

Since there is a net energy in the system – a changing balance between kinetic and potential – the system will not settle into a position of minimum potential energy, and even if it does it is not likely to stay there.

In order to get it to settle into some form of stable configuration, we need some form of energy dissipation.

We can rewrite equation 1 as

$$F - \mu v = ma \qquad (2)$$

Where μ is a coefficient of friction, proportional to the velocity of the system. This is a dissipative equation, reducing the total energy in the system, and hence will find a local minimum.

When the mass is small and the friction is large, this approximates to

$$F = \mu v \qquad (3)$$

We use this equation to calculate the direction and amount to move each body. When we reach the steady state, we have (for non-zero masses)

$$0 = \mu v \Rightarrow v = 0 \qquad (4)$$

Thus, in our representations the steady state that the arrangement evolves to is static, since there is no movement of the elements.

This representation can then be explored at will by rotating it, zooming in and flying through and around it. It is a completely abstract representation of the data, and so has no preconceptions built in. Different data to attribute mappings will clearly give different structures, but the system can at least produce a view of more than 3 dimensions of the raw data at once.

A typical structure is shown in Figure 2.

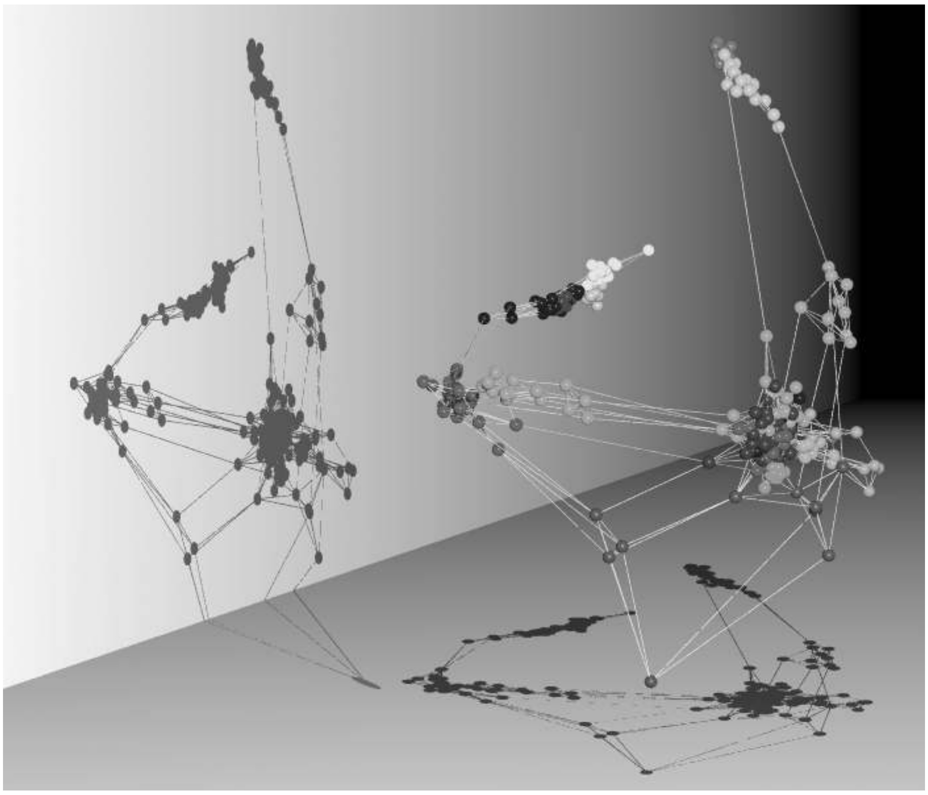


Figure 2: Nodes and links self-organized into a stable structure

To evolve the structure from the initial random state to the final static one, each node is checked for links to other nodes, and the forces of those links is added vectorially to give a net force, and the node is then moved according to that force using (3) above. Computationally, the process scales exponentially with the number of links, which is usually proportional to the number of data points, and so the evolution to the stable structure moves from being a real-time process that you can watch towards one that has to be allowed to run for a long period of time as the dataset increases in size. In general, this is not a problem, since the initial arrangement of data is random and the evolutionary process is not in itself informative (although it is interesting to observe). However, when the visualisation is used as a component in the data mining tool, this is designed to be an interactive process, and so we have taken a number of approaches to speeding up the relaxation to steady state. The first involves placing the nodes into the space in a non-random position initially; each node is placed 'near' a node it has a link to. This is marginally more computationally expensive initially, but reduces the numbers of nodes that have to move a large amount through the visualisation, and hence case large scale changes in other nodal positions. The most effective approach is to use predominantly local relaxation, however: instead of considering all the forces to act over infinite distance, we can limit nodal interactions to be very local, so that nodes a long way away do not exert any forces on the ones in question (much like assuming that the gravitational effects of all the stars except the sun are negligible). Once the system has undergone some initial relaxation, which provides some level of organisation, we can also focus on the local neighbourhood much more, and occasionally recompute the longer-range interactions. This is akin to organising a tight cluster properly, but then treating that as one structure for longer-range effects.

A combination of these approaches allows us to produce an effective steady state representation even with large datasets, in interactive time.

This approach achieves a number of things. It allows us to visualise high-dimensional data in a comprehensible and compact way. The visualisation in itself provides a lot of information about the dataset; it produces results that are similar to those achieved using approaches such as multidimensional scaling (Cox & Cox 1994; Inselberg 2002), but is somewhat more comprehensible because it tries to cluster 'similar' things with other 'similar' ones. It is certainly true that the choice of distance metric, and particularly which items to include and which to map to node characteristics, can affect the resulting visualisation, but we are searching for insight and meaning, not trying to come up with a single right solution. At different times, different features can be examined, and different results achieved - this is an inherent characteristic of searching for information, rather than an intrinsic problem with the approach. In any move from a high-dimensional space to a lower one, information will have to be lost - this approach at least preserves some of the main similarity characteristics of the original datasets.

The interface provides full 3D control of the structure, from zooming in and out, moving smoothly through the system (flyby), rotating it in 3D, and jumping to specific points, all controlled with the mouse.

Some typical structures emerge, recognisable from dataset to dataset. For example, a common one is the "dandelion head": a single central node connected to a number of other nodes with the same strength links. The links pull the attached nodes towards the central one, but each node repels the others, and so they spread out evenly around the central point. This looks much like a dandelion head. Another typical structure occurs when a number of dandelion heads are loosely linked together. The effect of the other heads in the chain forces the outer nodes away from being equidistantly spaced on the sphere and makes them cluster together somewhat on the side away from the link, and a series of "florets" are created, all linked together. It is because of this that some users have termed the visualisation "cauliflower space".

Visualisation to support serendipitous discovery

The visualisation approach itself supports serendipitous discovery. We have used the visualisation in isolation for a number of tasks (Hendley et al. 1999). One of the more effective ones has been the visualisation of users internet browsing behaviour (Wood et al. 1995). Each page visited is represented by a node, and their page transitions are represented by the links. Typically, users start on a home or an index page, and move out and back a number of times before moving off down a promising thread: this behaviour, when visualised in real time, produces a dandelion head with increasing numbers of 'seeds' (the outer nodes) and then switches towards a floret as the thread is followed. A new index-type page is reached (sometimes after one hop, sometimes after many, and another floret is created. Often, there are links back to the originally explored pages, and when the user follows these the visualisation pulls itself into a ring, representing a notion of closure and returning that has an exact analogy in the real world. This representation, linking related items, allows the user to explore the space freely but keep track of their navigation, and to see how different explorations can lead to the same results. This supports the serendipitous discovery of related material. The effect of this work prompted us to explore further ways of supporting web browsing, a theme we will return to later.

A different representation is formed if we visualise the structure of web pages: pages themselves are nodes again, but hyperlinks map to visualisation links. A web site has a fairly typical cauliflower image, caused by closely interrelated and interlinked sections, tied back to a common home or index page, with links off to other cauliflowers where the site links externally to other sites.

<<FIGURE 3a-d to insert here>> Figure 3a: Visualising the result of one query	Figure 3b: Adding a second query
Figure 3c: Adding a third, unrelated query	Figure 3d: A sequence of four queries, showing the inter-relationships

The system has also been used to assist users comprehend their progress in information retrieval tasks (Beale, McNab & Witten 1997). Using a digital library as our domain, for each query a representation of the results was returned. A large node represented the query, and was fixed in the 3D space. Each document that matched the query was a mobile node, with a link attaching it to the query, with the link strength being how relevant the document was to that query. An initial query would return a number of documents, and so a distorted dandelion head would appear, as in Figure 3a. However, a second query that returned some of the same documents would show links from those documents to both fixed nodes, and hence the degree of overlap could be easily seen, as shown in Figure 3b. Such an approach allowed the user, in real time, to see how effectively they were exploring the space of documents and how those were interrelated to the queries made, as in Figures 3c and 3d. This is important as subsequent searches are often dependent on the results of the previous ones, and so having a representation of the history and its relationships to the present search matches more closely what the user is doing internally.

Interaction with the Data Visualization

When features of interest are seen in the visual representation of the data they can be selected using the mouse. This opens up a number of possibilities - data identification, re-visualization, and explanation. The simplest of these (data identification) is to view the identity or details of items in the feature, or export this information to a file for later use.

The second option is to re-visualise just the selected data, or the rest of the dataset without the selected parts. This can be used to exclude distorting outliers, or to concentrate on the interactions within an area of interest. One of the features of the Haiku system is this interactive indication of the things that we are currently interested in, and the subsequent focussing of the knowledge discovery process on categorizing or distinguishing that data. Of course, we can data mine the whole dataset without doing this, the approach taken by many other systems.

A key feature of the system is that this user selection process takes full advantage of the abilities of our visual system: users are generally good at picking up gross features of visual representations. Our abilities have evolved to work well in the presence of noise, of missing or obscured data, and we are able to pick out simple lines and curves as well as more complex features such as spirals and undulating waves or planes. By allowing user input into the knowledge discovery process, we can effectively use a highly efficient system very quickly to direct the knowledge discovery algorithms.

The third option asks the machine to process the selected data, using the data mining components. This allows the system to generate explanations of why features of interest exist. Typical questions when looking at a visual representation of data are: “Why are these items out on their own?”, “What are the characteristics of this cluster?”, “How do these two groups of items differ?”. The data mining has to produce answers to these sorts of questions, and present them in a way that the user can understand.

Evolving Rules with Symbolic Genetic Algorithms

The data mining component within the Haiku system is based on a genetic algorithm, with a decision tree approach (C4.5 (Quinlan 1992)) used for comparative purposes. We use a genetic approach for a number of reasons: see, for example – see Freitas (Freitas 2003) and Fayyad *et al* (Fayyad, Piatetsky-Shapiro & Smyth 1996) for a general review. For our purposes, however, there are two particular reasons that they are appropriate. Firstly, genetic algorithms are able to effectively explore a large search space, and modern computing power means we can take advantage of this within a reasonable timeframe. The mutation and crossover operations that are used to investigate this space can lead to unexpected, surprising new rules that reflect the serendipitous nature of the approach. Secondly, the genetic algorithm aims to discover rules and rulesets which optimise an objective function (termed “fitness”), and manipulation of this allows us to explore different areas of the search space. For example, we can strongly penalise rules that give false positives in order to obtain rules that can be used to determine the class of new data examples. Alternatively, we can bias the system towards rules that indicate the typical characteristics of items in a group, whether these characteristics are shared with another group or not. Short rules with few terms in are going to be easier to comprehend than longer ones, but longer rules reveal more information - again, we can allow the user to choose which they would prefer by controlling the fitness function. Initially we might prefer short rules, in order to get an overview, iterating towards greater precision in subsequent evaluations.

We use a symbolic genetic algorithm to evolve the rules; this produces terms to describe the underlying data of the form:

IF `term` OP `value|range` (AND …) THEN `term` OP `value|range` (AND …) (5)

where `term` is a class from the dataset, OP is one of the standard comparison operators (<, >, =, ≤, ≥), `value` is a numeric or symbolic value, and `range` is a numeric range. A typical rule would therefore be:
IF *colour* = red & *texture*= soft & *size* < 3.2 THEN *fruit* = strawberry

A set of these rules can, in principle, describe any arbitrary situation. There are two situations that are of interest to us; classification, when the left hand side of the equation tries to predict a single class (usually known) on the right hand side, and association, or clustering, when the system tries to find rules that characterise portions of the dataset.

The algorithm follows fairly typical genetic algorithmic approaches (Mitchell & Melanie 1996) in its implementation, but since it is a symbolic representation, we use specialised mutation and crossover operators, in order to explore the space effectively and to ensure that the components of the rules are not split at inappropriate points or combined in infeasible ways. We start with a number of random rules, and evolve the population through subsequent generations based on how well each rule performs on the dataset: how accurate it is, and how many false positives and negatives it produces, and its coverage of the data. The genetic algorithm aims to optimise an objective function, and manipulation of this function allows us to explore different areas of the search space. For example, we can strongly penalise rules that give false positive results, and achieve a different type of description than rules that may be more general and have greater coverage, but make a few more mistakes. Each rule is analysed in terms of the objective function and given a score, its fitness. The fittest rules are then taken as the basis for the next population, and new rules created.

Rules are created by “breeding” good rules together, which combine their attributes. A new rule is created by cutting two parent rules and joining the different halves; it then inherits characteristics from both parents. The cut points are known as crossover points, and are chosen to be in syntactically similar positions, in order to ensure that we are working with semantically meaningful chunks. As in biological systems, we also model mutation – a slight random change in one of the parameters. Mutation is specialised: for ranges of values it can expand or contract that range, for numbers it can increase or decrease them, and for operators it can substitute them with others.

There are three situations that are of particular interest to us:

- **classification**, when the left hand side of the equation tries to predict a single class (usually known) on the right hand side
- **characterisation** when the system tries to find rules that describe portions of the dataset
- **association** which detects correlations in attribute values within a portion of the dataset.

Statistically principled comparisons showed that this technique is at least as good as conventional machine learning at classification (Pryke 1998), but has advantages over the more conventional approaches in that it can perform clustering operations too. One of the key design features is to produce a system that has humanly comprehensible results. Rules of the form in (5) are inherently much more understandable than decision trees or probabilistic or statistical descriptions. It is also true that short rules are going to be easier to comprehend than longer ones. Since the genetic algorithm is trying to minimise an objective function, we can manipulate this function to achieve different results. If we insist that the rules produced must be short (and hence easier to understand) then the system will trade off accuracy and/or coverage but will give us short rules, because they are 'fitter', which provide a general overview that is appropriate for much of the data. Because the Haiku system is interactive and iterative, when we have this higher level of user comprehension, we can go back into the system and allow the rules to become longer and hence more specific, and accuracy will increase. For example, we can firstly get a general impression of the dataset, generating rules of the form:

```
If fruit is red then it's a strawberry
```

This may be only 80% accurate, but it gives us an overall perspective on what was originally a mass of numbers. This can be refined to produce rules such as:

```
If fruit is red and soft and diameter > 2.5cm then strawberry
      else if diameter <= 2.5 cm then raspberry
If fruit is red and hard then it's an apple
```

each with a known accuracy. This allows the user to move their understanding from a broad overview through to comprehending the detail.

The interaction works as follows: first, a group or number of groups is selected. Then the option to explain the groups is selected. The user answers a small number of questions about their preferences for the explanation (short/long; highly accurate/ general characteristics etc.) The system then returns a set of rules describing the features selected, and ensures that the rules conform to the level of detail that the user requires.

Knowledge visualization and feedback

The results from the data mining can be fed back into the visualization to give extra insight into their relationships with the data (Consens, Cruz & Mendelzon 1992). One of the aims of the Haiku system is to bridge the human-computer communications gap, by producing textual rules that are simple to understand and then showing the effects of those visually. Identified clusters can be coloured, for example, or rules added and linked to the data that they classify, as in Figure 4.

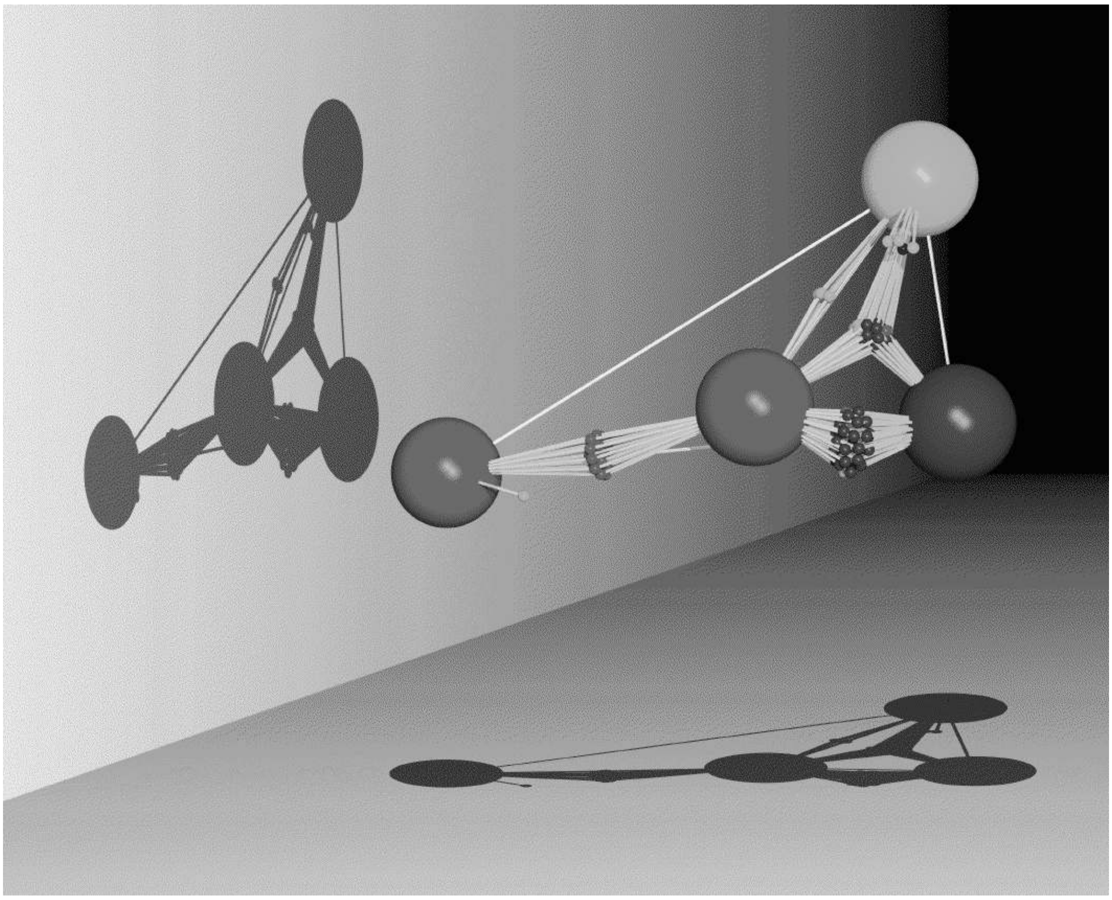


Figure 4: Rules and associated data

In this figure, rules are the four large spheres, with the data being the smaller spheres. Links are formed between the rules and the data that is covered by the rule, and the visualisation has reorganised itself to show this clearly. We have additionally coloured the data according to its correct classification (though this is less visible in the greyscale image).

A number of things are apparent from this visualisation, much more easily than would the case from a textual or statistical description. Much easier to see in colour than in the greyscale illustration here, the left two spheres are the same colour (fuchsia). The bottom right sphere is blue, and the top right one is green. In an ideal classification, there would be one sphere (one rule) per class, with all the data of that class being linked to only that sphere. If data is linked to more than one sphere, then it is classified by more than one rule. If it is linked to just one different coloured sphere, then it is misclassified: if it is linked to both its correct colour and an incorrect colour, then the system classifies it twice, once correctly. Conversely, if it is not linked to any sphere, it is not classified at all.

In this example, the data between the two fuchsia spheres is also fuchsia: it has been correctly classified by two rules. This data is collected in a central annulus between the two rules. The left rule also misclassifies one (green) data point, which is just to the right of the sphere – this is not linked to any other sphere and so is not classified by any other rule. The right fuchsia rule, whilst correctly classifying all the fuchsia data, also misclassifies much of the other data as well – shown by the linked points to the right of it – the bottom set is almost entirely blue (the colour of the right sphere), whilst that data with three links is predominately blue but with a few fuchsia and greens. On the far right hand side, the blue rule clearly does very well; it covers all its data and only misclassifies a few. The green rule at the top has mixed results, classifying most of the green points but also contributing to the misclassification of the points with three links. The visualisation allows us to assess the coverage and accuracy of the rules and understand more about how they interact with the data. In this case, we can see that if we deleted the right of the fuchsia rules, we would not lose much in terms of coverage of the data (only the few data points to its right) and would remove a source of confusion, reducing the misclassifications of the blue and green data. The system is fully interactive, in that the user can do this, or can now identify different characteristics and instruct the genetic algorithm to describe them, and so the process continues. For example, we may choose to focus on reclassifying the data that currently is linked to three rules, to improve our accuracy on this part of the data set.

It is interesting to note that as this visualization depends solely only on the relationship between knowledge (e.g. classification rule) and data, it can be applied to a very wide range of discoveries, including those made by non-symbolic systems such as neural networks.

Since the system can work in real time, new data could be constantly added into the system - users are not constrained to working with a fixed data set. The plasticity of both the visualisation approach and the genetic algorithm-based knowledge discovery system ensures that if new parameters or features of the current data set are discovered these can be added in as well. Haiku could therefore utilise new contextual and environmental information in real time, tapping in to both machine and human capabilities for adaptively processing changing data. One of the characteristics of ambient systems is that they utilise a much larger proportion of the information and context inherent in data in order to produce results, and the Haiku system works in sympathy with these goals. Since the approach is iterative, as facts and knowledge about the data are discovered, these can be fed back into the system to guide further discoveries and results, allowing the system to build upon the knowledge it has created.

This synergy of abilities between the rapid, parallel exploration of the structure space by the computer and the user's innate pattern recognition abilities and interest in different aspects of the data produces a very powerful and flexible system.

Case studies

Several machine learning datasets from the UCI Machine Learning Repository (Blake & Merz 1998) were used to benchmark the performance of data mining and classification from a quantitative perspective. The qualitative experience and support of serendipitous discoveries is not evaluated directly in this first study - instead we are testing the performance of the genetic algorithm approach with no user guidance. Good results on these datasets in quantitative terms will give us confidence when analysing new datasets, as we would expect that the qualitative experience of seeing, interacting with, guiding and refining the approach will lead to more insights.

We compared the approach on three varied examples from the repository: the Australian Credit dataset (Quinlan 1987), the Boston Housing dataset (Quinlan 1993), and the Pima Indians Diabetes dataset (Smith et al. 1988). We compared the genetic algorithm approach with C4.5 (Quinlan 1992), the definitive benchmark decision tree approach. Our experimental results are summarised below in Table 2.

Dataset	Genetic algorithm % Correct	C4.5 % Correct
Australian Credit	86%	82%
Boston Housing	64%	65%
Pima Indians Diabetes	73%	73%

Table 2: Summary of experimental results on three datasets

The genetic algorithm gave similar or better results, with statistically analysis showing it performed better than C4.5 on the "Australian Credit Data" (p=0.0018). No significant difference in performance was found for the other two datasets. We therefore conclude that the approach is at least comparable to the benchmark approach, even when used *without* any user input. Therefore, the default action of the system, with no guidance from the user, is no worse than the benchmark approach. We now investigate whether user participation in the discovery process provides us with any benefits or new knowledge. With user input, the system is essentially doing something different than a conventional data mining system would do and so there is not direct comparison we could make. However, by demonstrating that the unguided results are no worse than the benchmark, we can have confidence that the rules generated are fundamentally sound.

Case Study 1: Interactive Data Mining of Housing Data

We further investigated the Boston Housing dataset. This dataset contains information about properties, locality and people's economic status. Haiku was used to visualise the data and the complex clustering shown in Figure 5 was revealed.

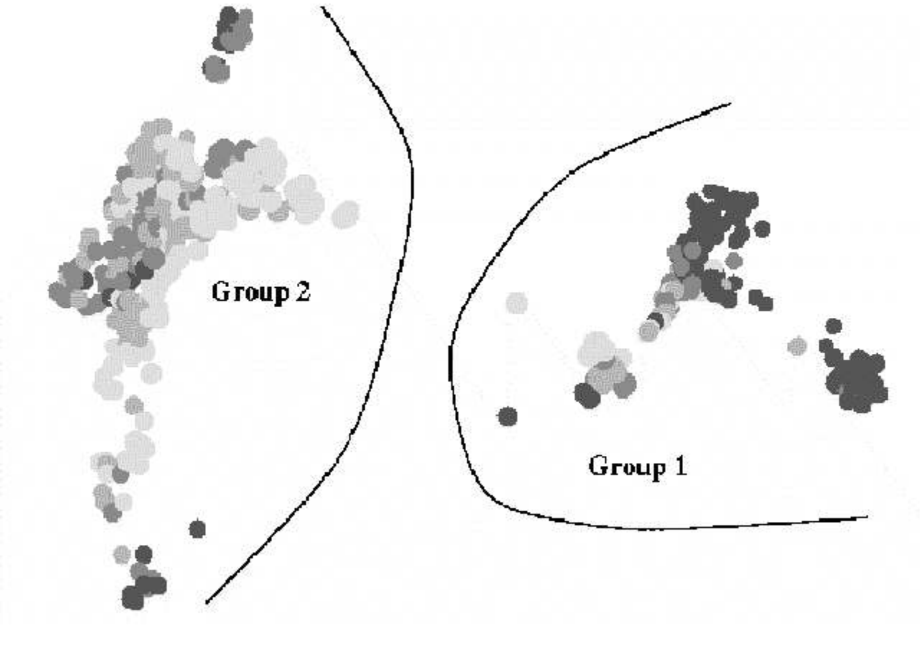


Figure 5: Clustering of Boston Housing Data

Two fairly distinct groups of data are visible, which show smaller internal features such as sub-groups. The two main groups were selected using the mouse, and short, accurate, classification rules were requested from the data mining system. These rules are shown below:

Bounds_river=true ⇒ GROUP_1
Accuracy: 100% Coverage: 43%

PropLargeDevelop = 0.0 AND 9.9 <= older_properties_percent <= 100.0 AND Pupil_teacher_ratio = 20.2 ⇒ GROUP_1
Accuracy: 94% Coverage: 83%

Bounds_river=false AND 4 <= Highway_access <= 8 ⇒ GROUP_2
Accuracy: 100% Coverage: 77%

Bounds_river=false AND 264 <= Tax_rate <= 403 ⇒ GROUP_2
Accuracy: 100% Coverage:69%

2.02 < Industry_proportion <= 3.41 ⇒ GROUP_2
Accuracy: 98% Coverage: 13%

5.68 <= Lower_status_percent <= 6.56 ⇒ GROUP_2
Accuracy: 96% Coverage: 75%

Bounds_river=false ⇒ GROUP_2
Accuracy: 73% Coverage: 100%

This case study illustrates the following characteristics:

- The interactive visual discovery approach has revealed new structure in the data by visual clustering.
- We have used human visual perception to determine features of interest, and application of the data mining algorithm has generated concrete information about these “soft” discoveries.
- Together, interactive data mining has delivered increased knowledge about a well known dataset.

The synergy between visualisation, interactivity and artificial intelligence therefore produces a system that supports the development of new knowledge, even when applied to well-examined reference datasets. Using ambient intelligence, we gain an awareness of the data, the users interests, and can therefore produce new and interesting explanations.

Case Study 2: Applying HAIKU to telecoms data

Massive amounts of data are generated from monitoring telecommunications switching. Even a small company may make many thousands of phone calls during a year. Telecommunications companies have a mountain of data originally collected for billing purposes. Telecoms data reflects business behaviour, so is likely to contain complex patterns. For this reason, Haiku was applied to mine this data mountain.

The data considered detailed the calling number, recipient number and duration of phone calls to and from businesses in a medium sized town. Other information available included business sector and sales channels. All identity data was anonymized.

Visualising call patterns

A number of companies with particularly high numbers of calls were identified. These were visualised separately to identify patterns within the calls of individual company. Figure 6 shows a clustering of calls from a single company. The most immediately obvious feature is the "wave" to the right of the image. This has been labelled A. Also visible are various other structures, including the two cluster labelled B and C.

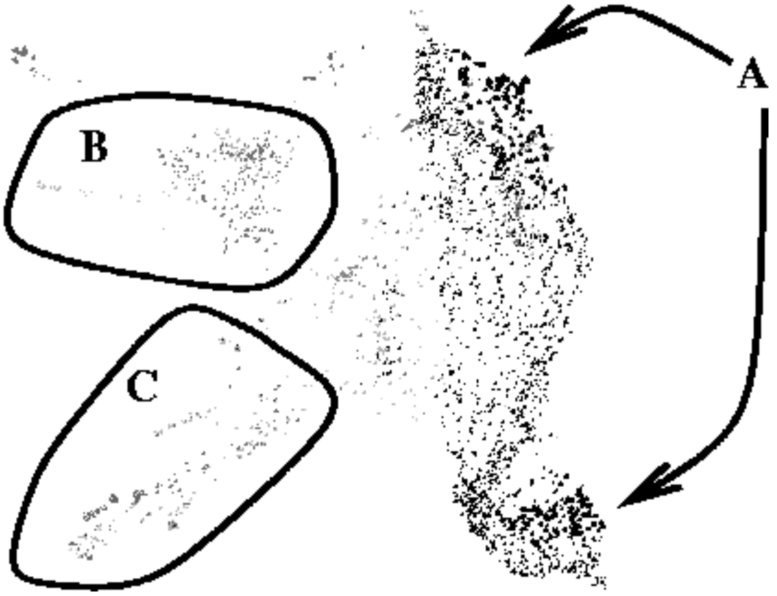


Figure 6: Visualization of Telephone Calls from one Site – User Selected Groups are Marked

Discoveries

After identifying these features, we then asked the system to explain their characteristics. The following rules were discovered by the system, and translated into sentence form for clarity.

- All calls in group A are to directory enquiries.

- Further investigation, selecting parts of the “wave” showed that the wave structure was arranged by hour of day in one dimension and day of week in the other.
- Within group B, about 70% of calls are to two numbers. 90% of all calls to these numbers fall into the group B. Almost all of the remaining 30% of calls in group B are to another two numbers. Most long distance ISDN calls are in group B. All but one call in the group has these properties. Most calls in the group are also charged at the same rate.
- About 80% of Group C calls are ISDN calls, and about 10% are from Payphones. About one third occur between 21:00 and 22:59, and about one half start at 15 minutes past the hour. Most are long distance calls. About 50% of the calls are very long, lasting between 8 and 15.5 hours.

We can see that, for this dataset, Haiku discovers some very interesting facts about the calling patterns of a company. Notice that we can produce short, comprehensible rules that cover a significant portion of the dataset, which are intrinsically much more usable than detailed descriptions of 100% of the data. These insights can then be used by the company to optimise their phone usage, or, as for this study, to feed back to the telecoms company some concepts for marketing and billing strategies.

Supporting Internet Browsing

In this section of the paper, we turn our attention to the second system developed to support serendipitous discoveries.

We have focussed on supporting internet browsing. Users are relatively well supported for searching, which is the quest for something specific, with tools such as Google, A9 and so on. The other common form of internet behaviour is monitoring, the repeated return to a location in order to look at new information (e.g. news site, stock quote page) and this is increasingly supported with RSS feeds or page scraping algorithms. Whilst the ease of use of the web browser can claim credit for the explosion of internet usage amongst the general public, it has not evolved significantly to support the undirected, passing interest-driven wanderings of users. Browsing the internet can be seen as a loosely directed traverse of a series of disconnected tree structures, with backtracking. Pages are nodes, with links as branches. Which link is taken is governed by whether the user finds it interesting enough to follow; if taken, it leads to a new page and potential new links being followed - alternatively, the path may be retraced back to a more interesting point, or that tree abandoned and a new one started.

We employ the same principles as before within the system. We aim to put the user at the centre of the interaction, leading the process, and use artificial intelligence approaches to augment their skills to produce a synergistic system. We do not want the user to have to train or interact directly with the artificial intelligence components; these need to acquire their information from the ambient environment and the history of interaction, rather than directly through explicit training. We use intelligent modelling to determine the context of the internet interaction and hence provide some guidance through the multiplicity of options. Our aim is to give the user some guidance as to which parts of the tree are likely to be of interest to them, without cutting off any options.

The system (termed 'Mitsikeru', Japanese for “to find out, locate”) uses an agent-based system to capture and model the user's behaviour and determine the context of their interaction, and then looks ahead at the web pages linked to from the current page to determine their relevance to this particular interaction. Related systems include Leitza (Lieberman 1995) and (Balabanovic 1997).

Mitsikeru design

The system can be broken down into three parts; determining the current browsing context, determining the relevance of future pages based on the current context, and communicating this to the user. The system operates seamlessly as a proxy between the browser and the internet - all user interaction is still via the browser.

Mitsikeru incrementally builds a 'master' table for each browsing session based on word frequencies found in pages. This table consists of words and their corresponding frequencies, and is pruned by removing very common words. This table therefore represents the current browsing context for this session. Within any one browsing session, users may be following different threads of interest, all of which are recorded within the table. It therefore does not represent a single theme or topic, but mixes all the topics that the user is currently interested in.

The system uses a proxy to look ahead to the pages linked to the current page which it then examines on behalf of the user. The proxy performs look-ahead and also parses the HTML to return the text of the page. The look-ahead therefore works for dynamically produced pages, but fails for those that are image-based (without Alt tags). Mitsikeru produces a 'page' table for each of these pages, in which the page is represented as a table of words and their frequencies. It uses these page tables to update a 'history' table, which is a list of all the words ever seen. We then use these tables to determine the relevance of the potential pages, based on what is of interest in the current session. The fundamental concept is that pages that have content similar to that we have been looking at in this session are more likely to be of interest to us than pages that are on different things altogether. In order to do this we use a Bayesian approach to determine a measure of relevance. There are many alternative approaches to determining relevance: we could have used entropy methods, or Chi-squared tests, or other latent semantic indexing approaches. The intention was to provide an algorithm that was relatively simple and fast to compute. Its benefits come from supporting the task effectively and presenting information efficiently. Different users will have different perceptions of relevance in any case, so any more complex measure is still doomed to psychological failure in terms of being more accurate. We are aiming to support and guide, not dictate and remove thought.

The principles of the algorithm are as follows. Words that are highly common (as defined by the history table) are likely to occur in the next page in any case, and are not that informative. Words that are not common which do occur on the next page are of more interest. Of even greater interest are uncommon words that exist in both the current browsing context and in the next page, as the chances of these occurring by chance are low. This approach to calculating relevance allows us more leeway in calculating the current browsing context, as we need not model each separate browsing task but can collate many tasks into a single representation.

Defining Interesting
In the data mining system, we provided a visualisation of the complex data and enabled the user to indicate their interest interactively. In this context, we use a more indirect approach, and use the recent history of viewed pages as a measure of what the user currently finds interesting.

With n_{PT} as the number of words in the page table, and n_{MT} as the number of words in the master table, with $P(h)$ as the historic probability of a word, we can write the probability that the word occurs in the page table as

$$P_{PT} = 1 - (1 - P(h))^{n_{PT}}$$

The probability that the word occurs in the master table is

$$P_{MT} = 1 - (1 - P(h))^{n_{MT}}$$

The probability that the word exists in both the current and the master table is therefore

$$P(both) = P_{PT} \times P_{MT}$$

We now have probabilities that describe how common or not our word is in the both the current context and in the linked page under consideration, compared to the medium overall. We can write the *surprise factor* as:

$$\textit{P(one or other or both)} = P_{PT} + P_{MT} - (P_{PT} \times P_{MT})$$

Our definition of an interesting word is one that is neither a chance occurrence, nor very common, which we can write as follows:

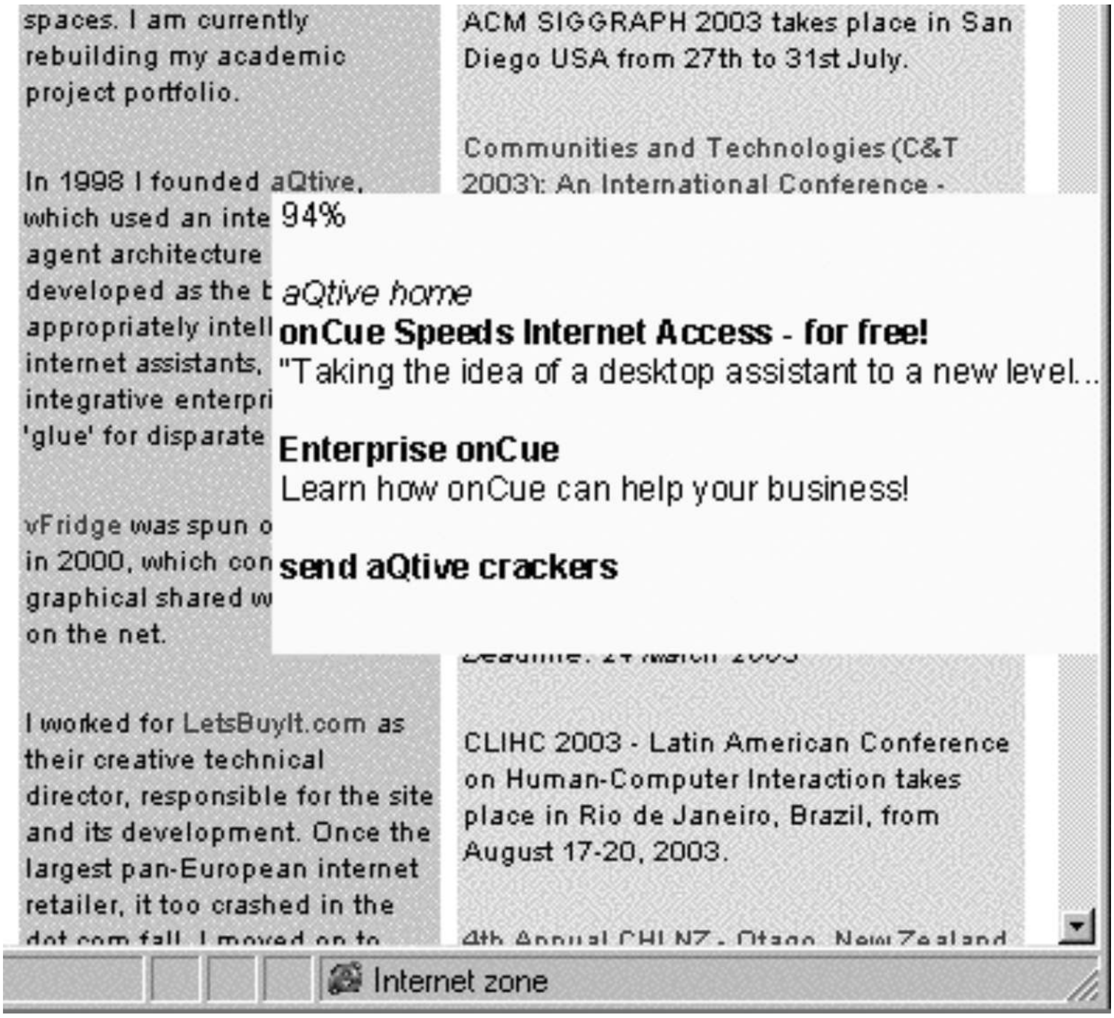
$$\textit{P (keyword|surprise factor)} = \frac{P_{PT} \times P_{MT}}{P_{PT} + P_{MT} - (P_{PT} \times P_{MT})}$$

Only interesting words are added to the master table. Interesting words lie towards the zero end of the spectrum. (In the implemented system, we chose words in the range 0.0006 to 0.1010). Note that this approach does not require us to filter the pages to remove HTML formatting, as this is treated as common and uninteresting. It is also not necessary to use a stemming algorithm (though it would have increased the generalisation of the system in early usage without compromising its overall performance).

We add the number of interesting words found on a page, and scale this by the number of words on the page, returning the result as a percentage, to arrive at our final relevance value. This metric favours shorter pages with more interesting words over longer pages, which is intuitively correct, though we recognise many alternatives are possible.

Information Presentation
Having calculated the relevance of the linked pages we have to present this information to the user. It is imperative that this is done in a non-intrusive manner

(Dix et al. 2003), allowing the user to maintain their conventional browser usage, and not stopping them from switching behaviours. We achieve this though the proxy adding a layer of DHTML to the current page. Links that are dead are removed, whilst others are colour-coded according to their relevance. Strongly coloured links are directly relevant, whilst irrelevant links are closer to the standard text colour. These codings give an immediate guide to the links that are most likely to be profitably followed, without cutting out any options. When the user hovers over a link, a summary of the page appears (title, initial sentence, main headings) and a relevance score is displayed in a small post-it style popup, achieved using DHTML, as shown in Figure 7. This allows the user to gain more information about the content of the page before actually deciding to visit it, and allows them to assess which of the likely candidates they should explore next much more rapidly.



spaces. I am currently rebuilding my academic project portfolio.

In 1998 I founded aQtive, which used an inte agent architecture developed as the b appropriately intell internet assistants, integrative enterpri 'glue' for disparate

vFridge was spun o in 2000, which con graphical shared w on the net.

I worked for LetsBuyIt.com as their creative technical director, responsible for the site and its development. Once the largest pan-European internet retailer, it too crashed in the dot com fall. I moved on to

ACM SIGGRAPH 2003 takes place in San Diego USA from 27th to 31st July.

Communities and Technologies (C&T 2003): An International Conference -

Deadline: 24 March 2003

CLIHC 2003 - Latin American Conference on Human-Computer Interaction takes place in Rio de Janeiro, Brazil, from August 17-20, 2003.

4th Annual CHI NZ - Otago, New Zealand

94%

aQtive home

onCue Speeds Internet Access - for free!

"Taking the idea of a desktop assistant to a new level...

Enterprise onCue

Learn how onCue can help your business!

send aQtive crackers

Figure 7: Hovering over a link ("aQtive", in this image) brings up a summary of that linked page as well as a score that relates to its likely relevance to the current task

The system assists in search behaviour as well. Since browsing is done via a proxy server that acts as a cache for pages, we can bias any search towards pages that have been recently browsed. This means that pages that are an equally good keyword match but have been recently looked at are ranked much higher in the returned results. This allows us to more easily return to briefly seen information that we want to subsequently study in further detail.

Evaluation

The system has been designed to primarily address the browsing needs of users. As a task with flexible or ill-defined goals, it does not seem appropriate to provide quantitative measures showing, for example, reduced time to find a particular item.

Instead, we have undertaken informal studies in which users have worked with the software and reported back their qualitative views. Users were drawn from a mixed population of science, engineering and humanities students, academics and the wider public. They ranged across the spectrum from hourly internet users to those who had only recently started using the internet. Their qualitative views can be summarised as follows:

- Mitsikeru is easy to learn; it has the same browser interface and interacts with web pages in the same manner as they are used to.
- The colouring of links is generally successful, with people tending to follow the strongly advised links most of the time. However, some users reported often wanting to see what was behind the other links to see how correct the system was.
- Presenting summary information about the next page was generally very well received, though the content and location of that information was sometimes criticised.
- Transitions between different browsing threads were not always carried out effectively by the system.
- The ability to inform the system as to its success or failure in recommending pages was requested by a number of users. This is an interesting point since it highlights that users are prepared to put in effort in helping the system become more accurate, and means that we do not have to solely rely on improved implicit techniques.
- Higher levels of satisfaction when browsing were reported.

Having a metaphor to understand how the system works was also mentioned by a number of people: they wanted to try and see why the recommendations were being made. The system as described is constructed from a number of independent but communicating modules: the pre-fetching proxy, the DHTML additive editor, the context modelling, the calculation of relevance, the page summary creation. Many of these modules act autonomously (without input from the user) and semi-intelligently (adapting their behaviour depending on the pages seen, providing enhanced information effectively 'hidden' from the user). We have found, through discussions with users, that they are best described as “agents”. Without getting in arguments as to definitions, users find that considering these processes as agents allows them to build up a more complete understanding of the system, and as a metaphor for user understanding, agents offer a positive contribution.

Summary

We have presented examples of systems that use ambient intelligence to collect information from their environment, from the data and from the user, in order to produce a more effective interaction. In particular, the synergy between artificial intelligence components (whether they be genetic algorithms, force-directed visualisations, or Bayesian statistics) and the user’s natural abilities and interests have allowed us to develop systems that support the free exploration of data and information, supporting the development of relationships and insights between disparate items. Representing the insights gained from the machine learning techniques has been critical to the successes of the systems, whether it be data visualisation, rule visualisation, or browsing behaviours and recommendations. We have developed systems that show that, by using appropriate technologies, we can keep the user at the centre of an interaction and still support them in making new discoveries in different ways, making for a more serendipitous environment.

Acknowledgments

This work was partially supported by grants from British Telecom, Integral Solutions Ltd and British Maritime Technology. Particular thanks to Alan Dix for insightful input, and to Andy Pryke who developed much of the Haiku system and did a lot of work on the UCI datasets. Thanks to Nick Drew and Bob Hendley for their work on the visualization parts of the system, and to Carl Burton for work on Mitsikeru. The comments of anonymous referees allowed me to clarify the paper further.

References

Abowd, G. & Beale, R. 1991, 'Users, systems and interfaces: a unifying framework for interaction', HCI'91: People and Computers VI, eds D. Diaper & N. Hammond, Cambridge University Press, pp. 73-87.

Balabanovic, M. 1997, 'An adaptive Web page recommendation service', in, Proceedings of the first international conference on Autonomous agents, ACM Press, Marina del Rey, California, United States, pp. 378-385.

Beale, R., McNab, R.J. & Witten, I.H. 1997, 'Visualising sequences of queries: a new tool for information retrieval', IEEE Conf on Information Visualisation, IEEE, London, England, pp. 57-62.

Blake, C.L. & Merz, C.J. 1998, UCI Repository of machine learning databases, http://www.ics.uci.edu/~mlearn/MLRepository.html, Irvine, CA: University of California, Department of Information and Computer Science.

Ceglar, A., Roddick, J.F. & Calder, P. 2001, Guiding Knowledge Discovery through Interactive Data Mining., Technical Report KDM-01-002. KDM Laboratory, Flinders University, Adelaide, South Australia.

Consens, M.P., Cruz, I.F. & Mendelzon, A.O. 1992, 'Visualizing Queries and Querying Visualizations', SIGMOD Record, vol. 21, no. 1, pp. 39-46.

Cox, T.F. & Cox, M.A.A. 1994, Multidimensional Scaling, Chapman & Hall, London.

Dix, A., Finlay, J., Abowd, G. & Beale, R. 2003, Human-Computer Interaction, 3rd edn, Prentice-Hall.

Fayyad, U.M., Piatetsky-Shapiro, G. & Smyth, P. 1996, 'From data mining to knowledge discovery: an overview', in, Advances in knowledge discovery and data mining, American Association for Artificial Intelligence, pp. 1-34.

Freitas, A.A. 2003, 'A survey of evolutionary algorithms for data mining and knowledge discovery', in, Advances in evolutionary computing: theory and applications, Springer-Verlag New York, Inc., pp. 819-845.

Friedman, J.H. & Tukey, J.W. 1974, 'A projection pursuit algorithm for exploratory data analysis', IEEE Trans. Computers, vol. 9, no. c-23, p. 881.

Hendley, R.J., Drew, N.S., Wood, A. & Beale, R. 1999, 'Narcissus: visualising information.' in S. Card, J. Mackinlay & B. Schneiderman (eds), Readings in information visualization: using vision to think., Morgan Kaufmann Publishers Inc., pp. 503-511.

Hofmann, H., Siebes, A.P.J.M. & Wilhelm, A.F.X. 2002, 'Visualizing association rules with interactive mosaic plots', Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining.

Holmes, G., Donkin, A. & Witten, I.H. 1994, 'WEKA: a machine learning workbench', Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems, Brisbane, Queensland, Australia, pp. 357-361.

Inselberg, A. 2002, 'Visualization and Data Mining of High Dimensional Data', Chemometrics, vol. 60, no. 1-2, pp. 147-159.

Lee, H.-Y., Ong, H.-L. & Sodhi, K.S. 1995, 'Visual Data Exploration Using WinViz', International Symposium on Intelligent Data Analysis, Conf. on Systems Research, Informatics and Cybernetics, IIAS Press, Baden-Baden, Germany.

Lieberman, H. 1995, 'Letizia: An Agent That Assists Web Browsing', International Joint Conference on Artificial Intelligence, vol. 14, Lawrence Erlbaum Associates, pp. 924-929.

Mitchell, M. & Melanie, X. 1996, An Introduction to Genetic Algorithms, Ann Arbor: MIT Press.

Nagel, H.R., Granum, E., and Musaeus, P. 2001, 'Methods for visual mining of data in virtual reality. In', PKDD 2001 International Workshop on Visual Data Mining.

Norman, D.A. 1988, The Design of Everyday Things, MIT Press Edition edn, MIT Press, London.

Pryke, A. 1998, 'Data Mining using Genetic Algorithms and Interactive Visualization', PhD thesis, University of Birmingham.

Quinlan, R. 1992, C4.5: Programs for Machine Learning, Morgan Kaufmann.

Quinlan, R.J. 1987, 'Simplifying decision trees', Int. J Man-Machine Studies, vol. 27, pp. 221-234.

Quinlan, R.J. 1993, 'Combining Instance-Based and Model-Based Learning', Proceedings of the Tenth International Conference of Machine Learning, Morgan Kaufmann, University of Massachusetts, Amherst, pp. 236-243.

Sharples, M., Jeffery, N., du Boulay, J.B.H., Teather, D., Teather, B. & du Boulay, G.H. 2002, 'Socio-Cognitive Engineering: A Methodology for the Design of Human-Centred Technology', European Journal of Operational Research, vol. 132, no. 2, pp. 310-323.

Shneiderman, B. 2002, 'Inventing Discovery Tools: Combining Information Visualization with Data Mining', Information Visualization, vol. 1, no. 1, pp. 5-12.

Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C. & Johannes, R.S. 1988, 'Using the ADAP learning algorithm to forecast the onset of diabetes mellitus', Proceedings of the Symposium on Computer Applications and Medical Care, IEEE Computer Society Press., pp. 261-265.

Wood, A., Drew, N., Beale, R. & Hendley, B. 1995, 'HyperSpace: Web Browing with Visualization', Third International World-Wide Web Conference, Darmstadt, Germany, pp. 21-25.

Wünsche, B. 2004, 'A Survey, Classification and Analysis of Perceptual Concepts and their Application for the Effective Visualisation of Complex Information', Australasian Symposium on Information Visualisation, vol. 35, eds N. Churcher & C. Churcher, Australian Computer Society, Christchurch, New Zealand, pp. 17-24.

Zhang, C., Leigh, J., DeFanti, T.A., Mazzucco, M. & Grossman, R. 2003, 'TeraScope: Distributed Visual Data Mining of Terascale Data Sets Over Photonic Networks', Journal of Future Generation Computer Systems (FGCS) Elsevier Science Press, vol. 19, no. 6, pp. 935-944.

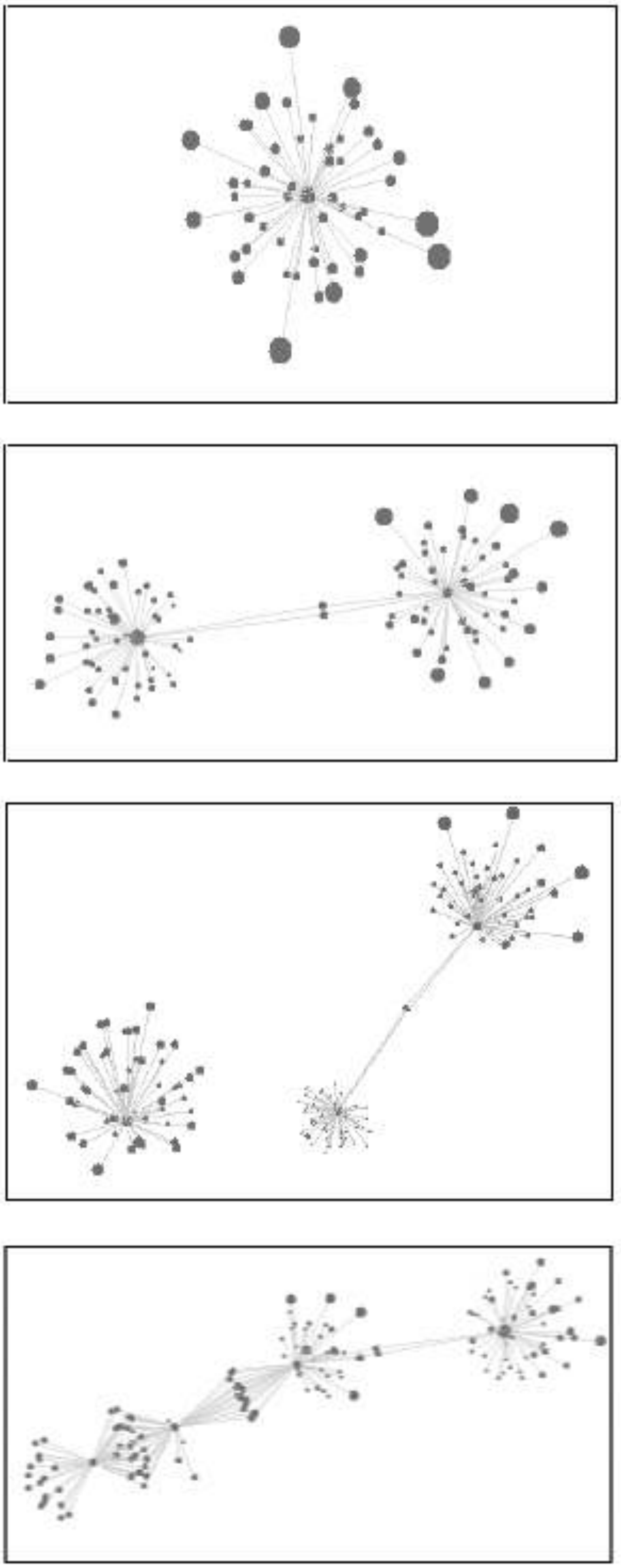


Figure 3, a-d