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 no 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.

Data mining, or knowledge discovery as it is sometimes called, is the application of artificial intelligence and statistical analysis techniques to data in order to uncover information. Given a number of large datasets, we are fundamentally interested in finding and identifying interesting relationships between different items of data. This may be to identify purchasing patterns, which are then used for commercial gain through guiding effective promotions, or to identify links between environmental influences and medical problems, allowing better public health information and action. We may be trying to identify the effects of poverty, or to understand why radio-frequency observations of certain stars fluctuate regularly. Whatever the domain of the data, we are engaged in a search for knowledge, and are looking for interesting patterns in the data.

But what is “interesting”? One day, it may be that the data falls into a general trend - the next it may be the few outliers are the fascinating ones. 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 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. 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. Many such systems have been utilised effectively, but we believe that there is more to data mining than grabbing just the choicest, most obvious nuggets.

There are further issues with current approaches to data mining, in that the answers are often almost as incomprehensible as the raw data. It may be that rules can be found to classify data correctly into different categories, but if the rules to do so are pages long, then only the machine can do the classification and we haven’t gained knowledge: we may know how to do the classification but have no insight into why it may be like that. Information, but not knowledge. We believe that we should be able to understand the answers that the system gives us. In order to achieve this, it may be that we need broader, less accurate generalisations that are comprehensible to the human mind, but then feel confident in the main principles to allow the machine to do classification based on much more complex rules that are refinements of these basic principles. For example: “if it’s red and squishy, it’s a strawberry” is easy to understand. Even if that’s true only 80% of the time, it’s a useful rule, and easier to grasp than:

```
red, deforms 4mm under 2N pressure, >3cm diameter = strawberry &
red, deforms 1mm under 2N pressure, <6cm diameter = cherry &
red, deforms 3 mm under 4N pressure, >5cm diameter = plum
else rasberry
```
which may be 96% correct but is hardly memorable. For many data mining systems, the rules developed are far more complex than this, each having numerous terms, with no overall picture able to emerge. For statistical-based systems, the parameter sets are even harder to interpret.

Since “interesting” is essentially a human construct, we have argued that we need a human in the data mining loop, and if we are to develop an effective system we need to allow them to understand and interact with the system effectively. We should also take advantage of the capabilities of the user, many of which we have tried to emulate with AI systems for many years and are still a long way from reproducing effectively. A key example is the human visual system, which is very effective at picking out trends within a mist of data points, capable of dealing with occlusion, missing values and noise without conscious effort. On the other hand, processing vast numbers of points and deriving complex statistics is something much better suited to computers.

This leads us to conclude that a knowledge discovery system should be interactive, should utilise the best in artificial intelligence, evolutionary and statistical techniques in deriving results, but should be able to trade accuracy for understanding, and needs to provide a way of allowing the user to indicate what is interesting and to understand the suggestions that the computer makes. An ideal system should be symbiotic, each benefiting from the intrinsic abilities of the other, and holistic, producing results that are much more powerful than each could achieve on their own.

KNOWLEDGE DISCOVERY WITH HAIKU

The Haiku system was developed with these principles in mind, and offers a symbiotic system that couples interactive 3-d dynamic visualisation technology with a novel genetic algorithm. The system creates a visualisation of the data which the user can then interact with, defining which areas are of interest and which can be ignored. The system then takes this input and processes the data using a variety of techniques, presenting the results as explanations to the user. These are in both textual and visual form, allowing the user to gain a broader perspective on what has been achieved. Using this information, they can refine what the system should look at, and slowly focus in on developing knowledge about whatever it is they are interested in. As well as using conventional rule generation techniques, Haiku also has a specifically designed genetic algorithmic approach to producing explanations of data. Each of these components is described in more detail below.

VISUALISATION

The visualisation engine used in the Haiku system provides an abstract 3-d perspective of multi-dimensional data. The visualisation consists of nodes and links, 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. 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 \( x = [x_1, x_2, \ldots, x_n] \), distance \( d \) between elements \( x_a \) and \( x_b \) is

\[
  d = \sum_{i=1}^{n} (x_{ai} - x_{bi})^2
\]

An example of this is shown below.

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Phone bill</th>
<th>Shopping</th>
<th>Petrol</th>
<th>Children</th>
<th>Age</th>
<th>sum distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer 1</td>
<td>124.23</td>
<td>235.12</td>
<td>46.23</td>
<td>2</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Customer 2</td>
<td>34.56</td>
<td>281.46</td>
<td>123.09</td>
<td>0</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>distance</td>
<td>89.67</td>
<td>-46.34</td>
<td>-76.86</td>
<td>2</td>
<td>5</td>
<td>-26.53</td>
</tr>
</tbody>
</table>

Total distance \( d = -26.53 \).

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 an 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.

This approach achieves a number of things. It allows us to visualise high-dimensional data in a comprehensible and compact way. It produces results that are similar to those achieved using approaches such as multidimensional scaling, 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 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 (a Newtonian space) we have the following condition:

\[ F = ma \]  

where \( F \) is the force applied to a body, \( m \) the mass of that body, and \( a \) the acceleration produced. This can be re-written as

\[ F = m \frac{dv}{dt} \]

where \( v \) is the velocity of the object.

When the visualisation is in a local minimum, there is no net force on any of the bodies (i.e. all the spring like forces from the links and repulsive nodal forces balance each other out) and so for each node, \( F = 0 \)

\[ 0 = m \frac{dv}{dt} \Rightarrow \frac{dv}{dt} = 0 \]

\[ \Rightarrow v = \text{constant} \]  

Therefore, in a steady state Newtonian space, each node may potentially have zero or a constant velocity. In other words, the steady state solution has dynamic properties, with bodies moving in orbit, for example.

In our space, we redefine (1) to be

\[ F = mv \]  

When we reach the steady state, we have (for non-zero masses)

\[ 0 = mv \Rightarrow v = 0 \]

Thus, in our representations the steady state that the arrangement evolves to is static.

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 1.
Figure 1. Nodes and links self-organised into stable structure.

To evolve the structure, 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 re-coding the system into OpenGL/DirectX, to take advantage of the power of modern graphics processors, especially for 3D work. The second places 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.

PERCEPTION-ORIENTED VISUALISATION

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 on the surface of a sphere centered on the main node. 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".

The visualisation in itself provides a lot of information about the dataset. We have used the visualisation in isolation for a number of tasks[7]. One of the more effective ones has been the visualisation of users internet browsing behaviour. 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[10]. 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.

The system has also been used to assist users comprehend their progress in information retrieval tasks. 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. 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. 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 [8,9]. 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 Visualisation
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
- Revisualisation
- Explanation

The simplest of these (Data identification) is to view the identifiers or details or items in the feature, or export this
information to a file for later use.
Another option is re-visualise the dataset without the selected data or indeed to focus in and only visualise the selected data. This can be used to exclude distorting outliers, or to concentrate on the interactions within an area of interest. Of course, we can data mine the whole dataset without doing this, the approach taken by many other systems. One of the features of the Haiku system is this interactive indication of the things that we are currently interested in, and the subsequent focusing of the knowledge discovery process on best describing that data only.

A key feature of the system is that this user selection process takes full advantage of the abilities of our visual system: humans are exceptionally 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 both 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 as well as reducing the work that the computational system has to do.

**Explanation**

The most striking feature of the system is its ability to 'explain' 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?". Answers to these types of question are generated by applying a machine learning component.

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 / characteristic) etc. The system returns a set of rules describing the features selected.

As an alternative, the classic machine learning system C4.5 [6] may be used to generate classification rules. Other data mining systems may be applied by saving the selected feature information to a comma separated value file.

**Rule Visualisation**

Rules generated using C4.5 or the GA based method can be visualised within the system to give extra insight into their relationships with the data. Rules are usually represented by massive nodes that don't move far in space, and are regularly spaced. Links show which rules apply to which data, and hence unclassified data and multiply classified data are shown well.

From this, the processing moves towards the computer, as the genetic algorithm-based process takes over.

**GENETIC ALGORITHMS FOR DATA MINING**

We use a genetic algorithm (GA) approach for a number of reasons. The first is that a GA is able to effectively explore a large search space, and modern computing power means we can take advantage of this within a reasonable timeframe. We use a special type of GA that evolves rules; these produce terms to describe the underlying data of the form:

\[
\text{IF } \text{term OP value|range (AND ... ) THEN term OP value|range (AND ... )}
\]

where \( \text{term} \) is a class from the dataset, \( \text{OP} \) is one of the standard comparison operators \( (<, >, =, \leq, \geq) \), \( \text{value} \) is a numeric or symbolic value, and \( \text{range} \) is a numeric range. A typical rule would therefore be:

IF colour = red AND consistency = soft 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 in its implementation, but with specialised mutation and crossover operators, in order to explore the space effectively. We start with a number of random rules, and evolve the population through subsequent generations based on how well they perform.

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. Crossover points are chosen to be in syntactically similar positions, in order to ensure that we are working with semantically meaningful chunks. Mutation is specialised: for ranges of values it can expand or contract that range, for numbers it can increase or decrease them, for operators it can substitute them with others.
Statistically principled comparisons showed that this technique is at least as good as conventional machine learning at classification [1], 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 GA 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 comprehension, we can go back into the system and allow the rules to become longer and hence more specific, and accuracy will then increase.

**FEEDBACK**

The results from the GA are fed back into the visualisation: identified clusters can be coloured, for example, or rules added and linked to the data that they classify, as in Figure 6.

![Figure 6: Rules and classified data](image)

In this figure, rules are the large purple, fuschia and green 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.

A number of things are immediately apparent from this visualisation, much more easily than would the case from a textual description. On the very left of the figure, one rule, the fuschia sphere, covers exactly the same data as the other fushia sphere, except it also misclassifies one green data point. But the rightmost fushia rule, whilst correctly classifying all the fushia data also misclassifies much of the other data as well. On the right hand side, the purple rule clearly does very well; it covers all its data and doesn't misclassify anything. The green rule at the top has mixed results.

The system is fully interactive, in that the user can now identify different characteristics and instruct the GA to describe them, and so the process continues.

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.

**CLASSIC CASE STUDY 1: WELL KNOWN DATASETS**

Several machine learning datasets from the UCI Machine Learning Repository[2] were used to benchmark the performance of data mining and classification. It should be noted that it focuses on quantitative performance, whereas the qualitative experience and use of perception-based mining techniques is not assessed. However, good results on these datasets in quantitative terms will give us confidence when analysing new datasets.
The GA-based approach gave perfectly acceptable results, with statistically analysis showing it performed better than C4.5[6] on the “Australian Credit Data” (p=0.0018). No significant difference in performance were found for the other two datasets. These results are summarised below

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Genetic algorithm</th>
<th>C4.5 []</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>% Correct</td>
<td>% Correct</td>
</tr>
<tr>
<td>Australian Credit [3]</td>
<td>86%</td>
<td>82%</td>
</tr>
<tr>
<td>Boston Housing [4]</td>
<td>64%</td>
<td>65%</td>
</tr>
<tr>
<td>Pima Indians Diabetes [5]</td>
<td>73%</td>
<td>73%</td>
</tr>
</tbody>
</table>

**CASE STUDY 2: INTERACTIVE DATA MINING OF HOUSING DATA**

The figure below shows a 2D view of the systems visual clustering of the Boston Housing data. Two used selected groups have been indicated.

GA based data mining was then applied to these user identified groups. The fitness function was chosen so as to bias the system towards the discovery of rules which are short and accurate. The rules are shown below:

<table>
<thead>
<tr>
<th>Rule 1.</th>
<th>CHAS = 0 &amp; 4 &lt;= RAD &lt;= 8 -&gt; User = grp2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LHS: 200 IHS: 347 both: 201</td>
<td></td>
</tr>
</tbody>
</table>

Rule 2. CHAS = 0 \& 364.0 <= TAX <= 400.0 -> User = grp2

<table>
<thead>
<tr>
<th>Rule 3.</th>
<th>CHAS = 1 -&gt; User = grp1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LHS: 35 IHS: 150 both: 35</td>
<td></td>
</tr>
</tbody>
</table>

Rule 4. 2.92 <= INUS <= 3.11 -> User = grp2

<table>
<thead>
<tr>
<th>Rule 5.</th>
<th>CHAS = 1 -&gt; User = grp1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LHS: 49 IHS: 347 both: 47</td>
<td></td>
</tr>
</tbody>
</table>

Rule 6. 5.68 <= LSTAT <= 6.06 -> User = grp2

<table>
<thead>
<tr>
<th>Rule 7.</th>
<th>CHAS = 0 -&gt; User = grp2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LHS: 100 IHS: 120 both: 132</td>
<td></td>
</tr>
</tbody>
</table>

This case study illustrates the following. The interactive visual discovery approach has revealed new structure in the data by visual clustering. Application of a data mining algorithm has generated concrete information about these “soft” discoveries. Together, interactive data mining has delivered increased knowledge about a well known dataset.

Having proven its worth on known datasets, we have used the system to try and discover new phenomena.
CASE STUDY 3: APPLYING HAIKU TO TELECOMS DATA

Justification
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.

Data
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.

Call patterns of high usage companies

Visualisation
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 (TBC) shows a clustering of calls from a single company. The most immediately obvious feature is the "blue 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"

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 "blue 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 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.

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.

CONCLUSION
The Haiku system for information visualisation and explanation provides a useful interface for interactive data mining. By interacting with a virtual data space created dynamically from the data properties, greater insight can be gained than by using standard machine learning based data mining. It allows users to explore features visually, to direct the computer to generate explanations and to evaluate the results of their exploration, again in the visual domain. By using a novel genetic algorithmic approach, we can bias rules generated to give us first a general overview and then progressively refine their accuracy and coverage as our understanding increases. This combination of intuitive and knowledge driven exploration with the mechanical power of the learning algorithms provides a much richer environment and can lead to a deeper understanding of the domain.
ACKNOWLEDGEMENTS
This work was partially supported by grants from British Telecom, Integral Solutions Ltd and British Maritime Technology. Thanks to Nick Drew and Bob Hendley for their work on the visualisation parts of the system, and to colleagues for their comments and help.

REFERENCES


