Being proactive – analytics for predicting customer actions

D D Nauck, D Ruta, M Spott and B Azvine

Customers should be at the heart of most businesses, and in particular service providers such as BT. In order to serve our customers better, we regularly introduce reliable processes and procedures to improve interaction with our customers, which is known as customer relationship management. Typically, organisations collect and keep large volumes of customer data as part of their processes. Analysis of this data by business users often leads to discovery of valuable patterns and trends that otherwise would go unnoticed and that can lead to prioritisation of decisions on future investments. Current tools available to business users are limited to visualisation and reporting of data. What is needed is modelling customer behaviours to be able to build future scenarios. More advanced tools and techniques have been available for a number of years but have not been developed for the business community due to the level of expertise required to use them. In this paper we present a number of tools and techniques developed for business users to perform advanced analysis on customer data. The tools can be used to perform sensitivity analysis, what-if analysis and impact analysis, all of which are aimed at prediction and simulation of future customer actions. The paper also covers application of the tools to real customer data and reports on some of the results obtained.

1. Introduction

Customers are at the heart of most businesses and this is especially true for service providers like BT. In order to serve their customers better, businesses have to introduce reliable processes and procedures for the interaction with their customers, which is known as customer relationship management (CRM). Typically, organisations collect and keep customer data as part of their processes. Therefore, data forms the core information source for CRM. In addition to process data, other forms of data are used, such as results from market research, demographic information, or surveys that provide customer feedback.

Process data reveals what customers do and survey data reveals what customers say they do. There is frequently quite a difference. For example, when analysing churn, service providers often find that a large percentage of customers who leave have previously said that they are very satisfied with the service.

Customer analytics can be divided into three main areas.

- Customer segmentation

  A limited number of customers are surveyed and this data is combined with account information and demographic data. By using methods like cluster analysis a number of segments are created and interpreted. The interpretation of the analysis results leads to a number of customer segments which are labelled with intuitive descriptions such as, for example, ‘traditionalists’ or ‘technophiles’. Then each account is mapped to one of the segments based on available data. This mapping obviously has to rely on less information than the segmentation exercise, because the vast majority of customers are not surveyed. Customer segmentation therefore is always imperfect.

- Predicting customer actions

  Based on historic information about actions by individual customers, we try to predict likely customer actions in the future. This kind of analysis is mainly based on process data that records events and interactions with customers. This kind of predictive analytics can only be done if a sufficiently long customer history is available and there is relevant interaction with the customer. This can be difficult for businesses, like, for example, satellite TV providers where the only interaction is typically payment of a monthly bill, assuming there is no automatic surveying of viewing behaviour. Telecommunications providers, in contrast, have better insights because they can see how customers are using the service and their operation requires a lot of effort in providing and maintaining services, and that leads to regular customer interaction on a large scale.

- Understanding customer views

  Most large businesses run some form of customer surveys. Businesses want to understand how their brand image is perceived, whether customers are satisfied with certain products or the company in general, or whether customers are happy to recommend the products and services they use. Businesses also want to understand what potentially drives satisfaction or loyalty and how such drivers can be influenced. Survey data has to be treated with caution because responses can be influenced by factors not covered by the survey, e.g. personal situation of the interviewee, interview situation and interaction with the interviewer, competitor activities. It is also important to keep in mind that surveys reflect what customers say and not necessarily what they actually do or are about to do.

In this paper we will be concerned with the latter two aspects of customer analytics — predicting customer actions and understanding customer views.

2. Data analysis issues

Customer analytics is essentially concerned with analysing data and requires standard techniques from areas like statistics [1], data mining [2], machine learning [3] and intelligent data analysis [4, 5]. In our work in customer analytics we particularly encountered issues around data quality and the selection of appropriate analysis methods.

2.1 Data quality

Most large established businesses run a huge number of legacy systems collecting data in different formats. This data is frequently not collected with analysis in mind and therefore important attributes can be missing. Data fusion across different legacy systems can be extremely difficult and often requires a lot of manual intervention and data cleansing. If customer survey data is involved we often see missing values because customers may refuse to respond to questions or questions are not asked in every survey. From our experience, it is not rare to have up to 75% missing values in such a data set. Poor data quality requires a big effort in data cleansing and pre-processing. It is important, for example, not simply to discard data records with missing values, but to check whether the fact that values are missing carries some hidden meaning that could be relevant to the analysis. Systematically missing data can introduce spurious relationships into the data. For example, discovery algorithms might detect a relationship between attributes that are missing in the same records and thus lead to the identification of wrong influence factors.

Another data quality problem can be introduced by aggregation functions. If summarised or averaged data is used where individual records would actually be required, relationships in the data can be lost or spurious relationships can be introduced. Basing an analysis on, for example, weekly summaries or averages is often done for one of two reasons — to reduce the amount of data or because some parts of the data are only available as summaries.

Table 1 gives a simple example of how averaging can hide an existing relationship. Assume we have $y = x^2$ and that we measure x and y in two consecutive weeks and compute weekly averages. While from the individual (x, y) records we can clearly see the functional dependency of y on x, this relationship is not visible in the weekly averages of x and y. It is as easy to construct examples, where x and y are unrelated but the averages show a spurious relationship. This can also be done for other aggregates like the median.

Table 1 An example where averages hide a relationship present in the data (see text).

	first week		second week	
Number	x	y	x	y
1	1.00	1.00	1.00	1.00
2	2.00	4.00	4.00	16.00
3	3.00	9.00	1.00	1.00
average	2.00	4.67	2.00	6.00

2.2 Choosing appropriate analysis methods

Due to a lack of expertise and tools, data analysis is often done in a too simple or even naive way. Linear models are the most frequently used analysis methods, because they are easy to understand. A linear model assumes that one or more dependent variables are determined by a linear combination of mutually independent variables. Additionally, in many methods from linear statistics there is an implicit assumption about normally distributed values. Both assumptions — mutual independency and normal distribution — are frequently not valid for real-world problems. Linear models cannot take compensatory or reinforcing effects into account. Especially in customer analytics, we observe different types of dependencies between all variables and these dependencies can be nonlinear in nature. Assuming a linear relationship without checking for nonlinear dependencies can obviously lead to wrong conclusions.

For example, consider the simplified scenario in Fig 1. We assume that by increasing the effort we can drive down dissatisfaction. Let's further assume the real model is the nonlinear saturation curve. If we instead use a linear function to represent the dependency between dissatisfaction and effort, we might end up with the depicted linear function. This would give us the wrong assumption that we could reduce dissatisfaction down to zero by putting in enough effort instead of the more realistic saturation that would settle on a certain level of effort. It also could — in this example — lead us to assume that in order to reduce dissatisfaction to a target level of t% we would have to input the effort e_2 instead of the smaller effort e_1 required by the nonlinear model.

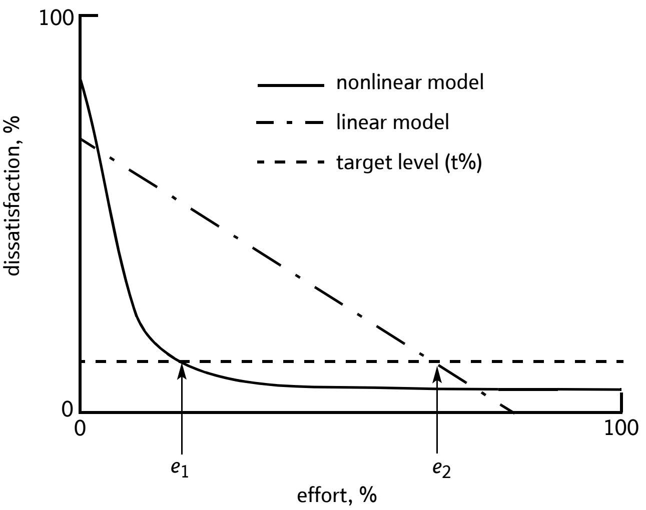

Fig 1 A linear and a nonlinear model.

In a typical customer analytics scenario we usually see some of the following methods applied depending on the skills of the analyst. These methods can all reveal useful information, but they also have limits that the analyst must consider.

A typical analysis would involve looking at the frequency distributions of all variables. From that we can learn in which areas there may be a problem, for example, in customer satisfaction. Given the right software, it is possible to drill down and look at the replies of certain customer groups based on demographical information or the replies to particular questions. From that we may be able to learn, for example, that users who complain about the layout of our Web portal are also more likely to complain about not finding the hyper-links than customers who are happy with the layout. The problem with that approach is that the analyst will only discover what he or she is looking for and that multidimensional dependencies will be overlooked, because they cannot be visually represented.

Another typical analysis is to look at correlations between the different variables. For example, if we look at customer satisfaction surveys, we may find that many questions in a survey are highly correlated, meaning the higher the satisfaction or dissatisfaction in one area, the higher the satisfaction or dissatisfaction in another area is likely to be. A correlation analysis, however, assumes a linear relationship, and its result is therefore only relevant if the assumed linear dependence actually exists. This type of analysis can overlook nonlinear relationships and cannot detect multidimensional dependencies.

In order to understand the quantitative influence of the result for one question on the overall satisfaction, we can use a functional model. In statistical analysis we typically see linear regression being used for this purpose. However, linear regression assumes that the individual variables are independent of each other and that a linear dependency between the independent variables and the target variable actually exists. For each value of an independent variable, the distribution of the dependent variable must be normal. These constraints are very strong and they are usually never fulfilled.

Especially the independence assumption is typically not realistic at all. One way to alleviate this problem is to run a principal component analysis and using the uncorrelated principal components as inputs to the regression analysis. However, it should be obvious that a linear regression model is unsuitable for modelling nonlinear relationships and it completely ignores mutual relationships between the inputs and thus cannot take compensatory or reinforcing effects into account.

In order to be more flexible and less constrained with the choice of a functional model, we can look at non-parametric nonlinear methods. They are not based on implicit distribution or independence assumptions and can model high-dimensional nonlinear relationships. Methods like decision trees, neural networks, fuzzy systems, neuro-fuzzy systems, support vector machines, etc, are known from areas like intelligent data analysis and machine learning. Some of these methods, such as decision trees and fuzzy systems, are rule-based and can be used to obtain information about the nature of the modelled relationships. Other systems, e.g. neural networks or support vector machines, are only suitable for making predictions because the way they represent relationships is not easily interpretable.

These kinds of nonlinear model are very powerful, but they are unidirectional. That means, they can only be used to compute the impact on one or more previously selected target variables when some independent variables or drivers change.

However, we also want to understand the effects on all the other drivers. To compute the impact that any variable has on any other variable we can use a multi-dimensional probabilistic model. Such a model is not restricted by linear dependencies or global independence assumptions. A suitable probabilistic model is a Bayesian network that can represent arbitrary probabilistic relationships between any number of variables.

In the next section we look at a software tool that is based on Bayesian networks and that has been used by us to analyse customer data.

3. iCSat

iCSat (intelligent customer satisfaction analysis tool) is a Java-based client/server platform for analysing customer data. Its initial focus was the survey data about customer satisfaction, but in fact iCSat is a generic tool and has subsequently been applied in other domains as well.

3.1 Bayesian networks

iCSat uses Bayesian networks [6, 7] to model dependencies between all variables available in a data set. A Bayesian network is represented as a graph where each node represents a variable and connections between the nodes represent direct conditional dependencies. Each node displays a probability distribution over the possible values of the variable represented by the node given the current state of the whole network. A Bayesian network is a convenient way of representing a high-dimensional probability space by exploiting conditional independence between variables. Nodes that have no direct connections are conditionally independent. We only need to represent conditional probabilities between connected nodes.

A Bayesian network exploits the fact that a joint probability distribution $p(x_1, \ldots, x_n)$ can be rewritten by applying the chain rule of probability as:

$$p(x_{i1}, \ldots, x_{in}) = p(x_{i1} | x_{i2}, \ldots, x_{in}) \cdot p(x_{i2} | x_{i3}, \ldots, x_{in}) \cdot \ldots p(x_{in})$$

where $\langle i1, i2, \ldots, in \rangle$ is an arbitrary permutation of $\langle 1, 2, \ldots, n \rangle$. In addition, we will often find the distribution of a variable x_{jk} can be described conditional on a set of parents Π_{ik} that is substantially smaller than $(x_{i(k+1)},\ldots,x_{in})$ and that renders x_{ik} independent from $(x_{i(k+1)},\ldots,x_{in})$, that is:

$$p(x_{ik} | x_{i(k+1)}, \ldots, x_{in}) = p\ (x_{ik} | \Pi_{ik}).$$

For example, if we have a data set of 10 variables where each variable can assume two possible values, then we have 2^{10} = 1024 possible combinations of values. In a probabilistic model we would have to maintain one value for each combination resulting in 1024 probabilities. If we manage, for example, to represent the 10 variables in a Bayesian network such that no variable has more than two parents and one variable has no parents at all (root node), then we would have to maintain only $2 + (9 \times 2^3) = 74$ probabilities. By computing within the Bayesian network we can still calculate the probabilities for all 1024 possible value combinations, but we are not required to compute all of them up-front and store them.

In order to obtain a Bayesian network we must first define its structure, i.e. determine which nodes are connected to each other, and then we must provide the conditional probabilities that describe the dependencies between the connected nodes. For small problems it would be possible that an expert does this manually. However, for larger problems and for non-experts, it is usually impossible to specify a Bayesian network from scratch. We have therefore implemented a powerful structure-learning algorithm into iCSat that can learn the connections within a Bayesian network automatically from data [8]. iCSat then uses a commercial Bayesian library (Netica) to represent the network and to learn the probabilities from data [9].

Bayesian networks are used by inputting observations or assumptions into some of the nodes and then studying the changes in all other nodes. As an example, assume we are looking at a set of customers who ordered a certain service in the last three months and who have subsequently been surveyed. Assume further that we are interested in the relationship between speed of provision and overall satisfaction with the service. Say, the node for provision time displays the options 'same day', 'next day' and '2 or more days'. By setting the value of that node to 'same day' we can immediately see the satisfaction distribution for all customers who experience a same day provision.

Because a Bayesian network can reason in all directions we can also run a scenario where we set the satisfaction level, say, to 'extremely satisfied' and then look at the distribution of provision times. Since all the nodes in the network change their distributions when we enter some value in any other node, we can also see the impact of changes in satisfaction or provision times on all other variables at the same time.

Large Bayesian networks can be very complex and difficult to work with in a what-if analysis (see Fig 2). User interfaces of standard software tools typically focus on the network structure and expect users to navigate through it. Users are also expected to create a network manually, which is basically impossible for domains where they have little or no knowledge about mutual

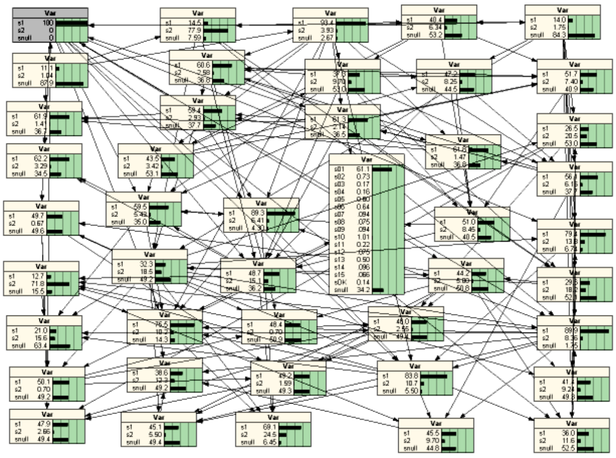

Fig 2 A Bayesian network.

dependencies between variables and are actually looking for a model helping to detect relevant relationships in the first place.

Business users require an intuitive and highly automated interface to Bayesian networks to benefit from the advantages that these models provide. In the following section we describe the iCSat platform that was implemented to support the analysis of customer satisfaction data.

3.2 Analysing customer data with iCSat

iCSat has been implemented to allow business users to easily analyse customer data, build models, run what-if scenarios, identify drivers, and set targets for them.

In order to build a new Bayesian network model, the user first loads a data description and a data set. The data description describes the meaning of each variable and its values and represents typically a survey, where a variable is a question and a value is a possible response. The data set is a table or spreadsheet where variables are organised in columns and each row or record represents data of one customer. The values in the data must represent mutually exclusive categories. Numerical data would have to be discretised first.

iCSat stores data sets and descriptions in an associated database. Once data is available, the user can start the modelling process by merely selecting the variables to be included in the new model, providing a name for the new model and starting the automatic model generation process.

iCSat first learns the structure of the Bayesian network. This is done by using the CB algorithm suggested by Singh and Valtorta [8]. This algorithm starts with a fully connected undirected graph and uses χ^2 tests to determine conditional independence of connected nodes. If two nodes are conditionally independent the connection between them is removed. Then several rules and heuristics are used to direct the edges such that a directed acyclic graph (DAG) is created. The DAG is used to create a topological order of the nodes. This order is used as an input to the K2 algorithm by Cooper and Herskovitz [10] that computes for each node the list of parent nodes and thus creates the structure of a Bayesian network. After that we use the Netica library [9] to compute the conditional probability tables of the network. Netica is also used to represent the Bayesian network and run all the computations within the network.

The learning algorithm iterates by starting with tests for zero order conditional independence, and then continues with 1st order tests and so on until an upper limit is reached, no more connections can be deleted, or a newly generated network is not better than the previous one. The complexity of the algorithm depends on the order of the independence tests and increases exponentially with the number of nodes. Therefore the algorithm is stopped after a maximum of 3 iterations, i.e. after 2nd order tests have been completed (an nth order test has to look at all subsets of cardinality n of the set of nodes connected to the two nodes whose independence is to be tested).

After that automatic learning process has been completed, a model can be loaded for analysis. iCSat provides a very intuitive GUI (Fig 3) where the user sees two columns of bar charts. In the left column the charts can be manipulated to represent input data. In the right column the predictions of the model are displayed. The charts in the prediction column contain two sets of bars to compare the predictions to the original distribution found in the data. Both columns can be fully configured to contain only the variables in which the user is interested. The following operations can be carried out on a model.

- Sensitivity analysis

  This automatically identifies drivers and their degree of impact on a selected variable (Fig 4). This information allows users to concentrate experiments on variables that actually impact on particular target variables.

- What-if analysis

  This understands how changes in the distributions of some variables affect all other variables. There are two modes — impact analysis and target setting. In impact analysis users can set some variables to particular values to, for example, simulate particular customer groups and see predictions for value distributions in other variables. In target-setting mode, users can specify a new frequency distribution for some variables and see the impact on the distributions of other variables.

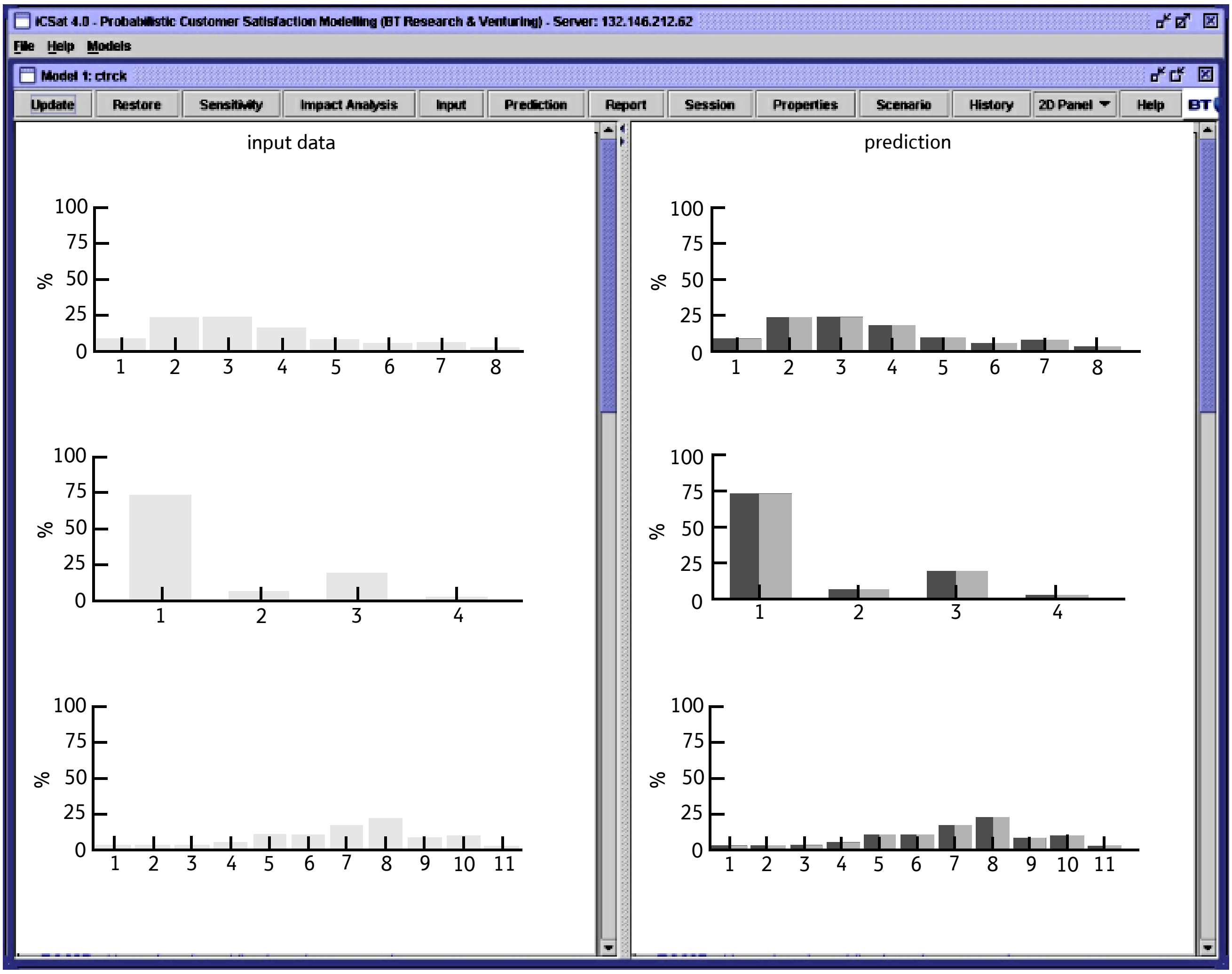

Fig 3 The graphical user interface of iCSat (variable names have been removed for confidentiality reasons).

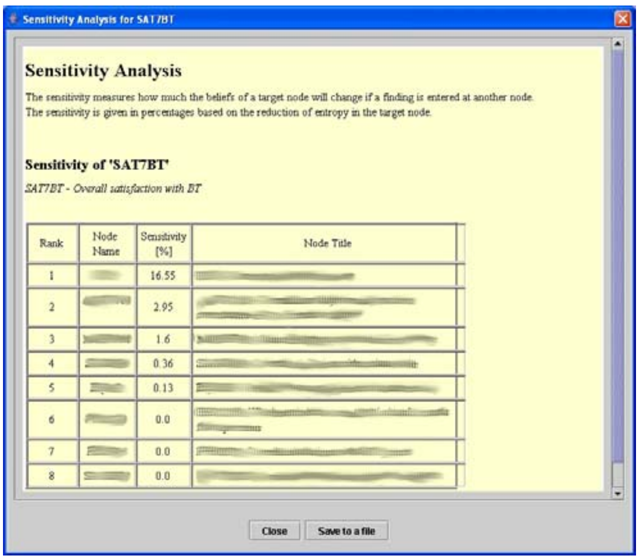

Fig 4 Result of a sensitivity analysis (variable names are obscured for confidentiality reasons).

- Report

  This creates a table of the current settings for import into office documents.

- History

  This allows saving what-if scenarios, reloading them and comparing them side-by-side.

- Data aggregation

  This reduces the complexity of a data set by combining groups of values to new values.

- 3-D View

  This operation represents the otherwise separate input and output charts in one 3-D chart (Fig 5). All analyses can also be carried out in the 3-D view.

4. Application of iCSat

iCSat has been used by us in different data analysis scenarios and it is also used by several business users

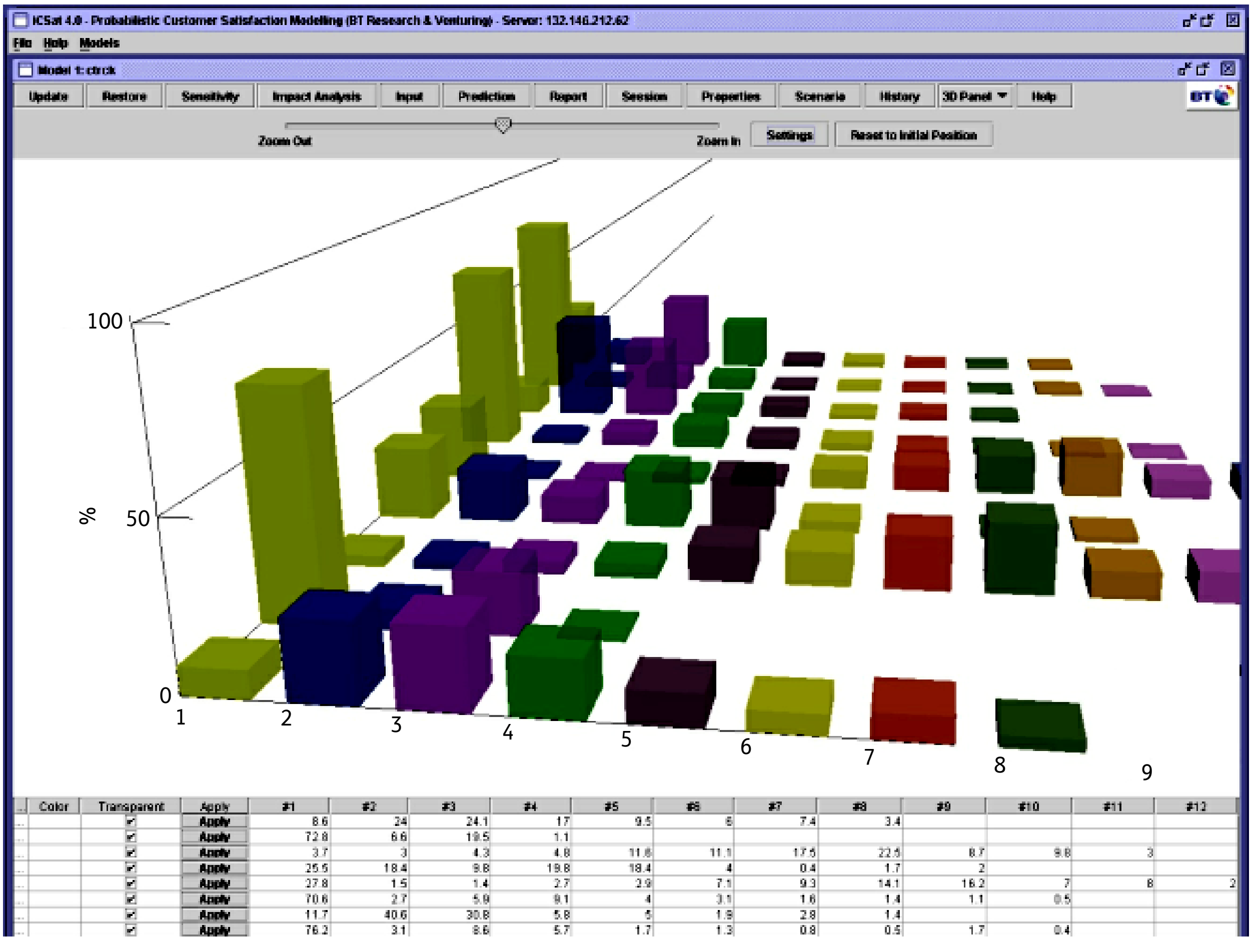

Fig 5 The 3D interface of iCSat (variable names are hidden for confidentiality reasons).

across BT. Below we give four examples of where the tool has been applied. Because of confidentiality reasons we cannot provide any details of the analysis results.

- Customer satisfaction analysis

  iCSat has been initially conceived for this purpose. By using data from customer surveys we have identified the main drivers of satisfaction and analysed their impact.

- Identifying customers in jeopardy

  We used iCSat to identify which circumstances in a repair process can lead to complaints by customers. Finding relevant process parameters allowed us to provide a simplified model for operational systems that allows customer services to obtain an early warning if a process for a customer is about to go wrong. A trial revealed that by intervening at the right time complaints could be substantially reduced.

- Target setting

  Combining process data with customer survey data allowed business users to understand the impact of certain process parameters on customer satisfaction. By setting a satisfaction target, target distributions for process parameters could be computed and used for setting internal performance goals.

- Field force performance

  By using data from field force operations we looked at the impact of certain regional factors on job performance.

iCSat is a generic data analysis tool that can be applied to any domain if the provided data is categorical. We are in the process of adding new functionality to the tool and in the near future it will be integrated into iCAN — a platform for intelligent customer analytics.

5. Intelligent customer analytics

iCAN is a Java-based client/server software package for the purpose of intelligent monitoring, analysis and prediction of customer behaviour, and its changes, during a complete customer life cycle with a service provider. iCAN is connected to a data warehouse with customer and process data and can work with both live and off-line data to provide analytics, build and apply various predictive models, and display the results. iCAN is designed in an open form which allows it to incorporate new analytical models at any time. All models covered by the tool provide full visualisation of their internal structure and offer user-friendly interfaces to carry out predictive experiments on customer behaviour.

The current research prototype includes two analytical methods — hidden Markov models (HMMs) [11] and decision trees. Decision trees [3] are capable of making static predictions of events with no related timing information, while HMMs can make time-stamped event predictions. Both models can use either live data stored in remote databases or off-line data from a data warehouse. The models are supported by some standard statistical data analytics and visualisation capabilities. Models can be stored in a database and can be applied to different data sets at any time.

Markov models represent a family of stochastic methods focused on the analysis of temporal sequences of discrete states. In traditional first-order HMMs, the states are hidden from observation, but each of them emits a number of observable variables which could take either discrete or continuous values. As in all Markov models the current state of an HMM depends only on its previous state which means no prior history of sequence evolution has any effect on the current state. An HMM is fully described by its parameters, which are the transition probabilities between hidden states and the probabilities or probability densities of the emission of observables. Given a coarse model structure there are three central issues in hidden Markov models:

- the learning problem — concerns calculation of model parameters based on the training observations of visible symbols,
- the evaluation problem — assumes that the HMM model is fully defined and concerns the evaluation of the probability that a particular sequence of visible states was generated by the model,
- the decoding problem — assumes the HMM and a set of visible observations while asking for the most likely sequence of hidden states that led to these observations.

Once trained, an HMM can be used for a variety of applications, wherever sequential data and its future evolution is in question. HMMs have started to be appreciated in business analytics. Recent cross-sale models include HMM as one of the most successful approaches for recommending products to those customers who are most likely to buy them given their recent purchases [12]. In the iCAN software platform we use HMMs to solve the much more general problem of customer life-cycle modelling where the distinctions between hidden states are less sharp and the observables are available mostly in a continuous form that is difficult to process. It assumes that a customer develops a variable behaviour path which starts from subscribing to a service offered by a business and ends when he decides to cancel the service.

The HMM has been implemented in the standard discrete form with a pair of Viterbi and Baum-Welch algorithms [11] used to find the unknown parameters of the hidden states of the network. The model assumes that customer events or experiences represent observable customer variables generated from unknown (hidden) behavioural states that the customers find themselves in. The model assumption is that the customers, who behave similarly, i.e. are in the same hidden behavioural state, that is they have the same distribution of the likely events/experiences, have the same distribution of emission probabilities in the visible states.

In addition to this general analysis of the population of customers, an HMM can also be applied to a single customer, for example, to analyse the churn risk. iCAN provides the latest historical data for the selected customer and the time evolution of the cumulative churn risk is presented graphically and numerically as shown in Fig 6.

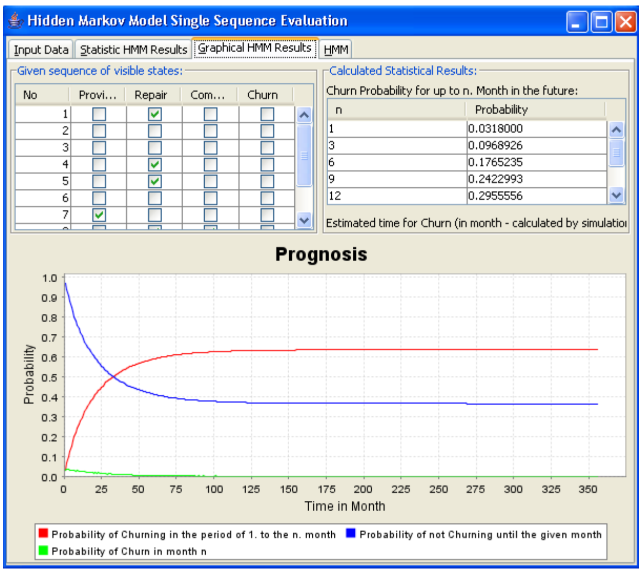

Fig 6 HMM churn analysis for a single customer.

Looking beyond just a churn prediction, the iCAN HMM can also be used for a comprehensive what-if scenario covering all the events encoded in the data (Fig 7). Given the trained HMM the user can make next step predictions based on a configuration of events and

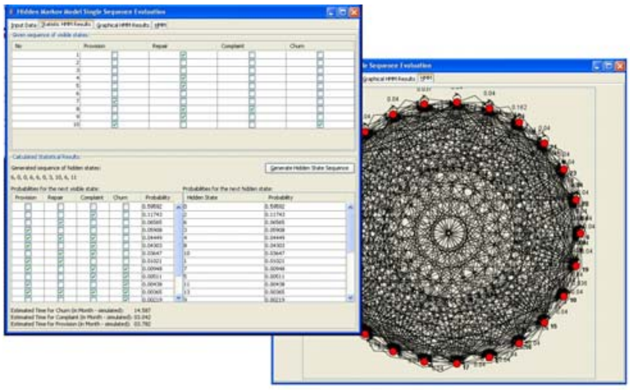

Fig 7 The iCAN HMM provides comprehensive what-if analysis.

subject to user-specified simulated past-event sequences. In all cases, predictions are delivered in the form of probabilities of event configurations, immediately calculated using learned model parameters. Additionally, the model also returns the estimated average time to the next event in all categories provided by the data. In case of churn, this statistic constitutes the estimated remaining lifetime with the service provider.

Because iCAN provides automatic model building and can generate predictions on customer level automatically, it can easily serve as an analytical platform or add-on for modern CRM systems like Siebel.

6. Conclusions

Customer analytics is an extremely important area for large businesses. Barriers to customer analytics are, typically, bad data quality and lack of expertise in analytics. Bad data quality can arise from inconsistent legacy systems and the fact that data gathering is usually done without a subsequent analysis in mind. One way of addressing data quality is to move from outdated legacy systems to a central corporate data model and repository on top of which modern CRM solutions can operate. A lack in analytics expertise can be addressed by using highly automated intelligent tools that provide advanced analytics but with an intuitive interface. Business users are domain experts not data analysis experts. Therefore we need tools that support them and allow them to focus on their job.

We have developed a suite of intelligent customer analytic tools that are automated to an extent that business users do not need to worry about the analytical methods being used but can easily run scenarios, test assumptions and discover relevant information. Software like iCSat and iCAN is based on our research results in automated intelligent data analysis [4]. While iCSat has already been rolled out for business use, we are continuing to develop new analytical methods to capture the complete customer life cycle in iCAN.

References

1 Sheskin D: 'Handbook of Parametric and Nonparametric Statistical Procedures', Third Edition, Chapman and Hall/CRC Press, Boca Raton, Florida (2003).

2 Fayyad U M, Piatetsky-Shapiro G, Smyth P and Uthurusamy R: 'Advances in Knowledge Discovery and Data Mining', AAAI Press and MIT Press, Menlo Park, CA and Cambridge, MA (1996).

3 Mitchell T M: 'Machine Learning', McGraw-Hill, New York (1997).

4 Nauck D, Spott M and Azvine B: 'SPIDA — a novel data analysis tool', BT Technol J, 21, No 4, pp 104—112 (October 2003).

5 Berthold M and Hand D J (Eds): 'Intelligent Data Analysis: An Introduction', Springer-Verlag, Berlin (1999).

6 Pearl J: 'Probabilistic reasoning in intelligent systems: networks of plausible inference', Morgan Kaufmann Publishers Inc, San Francisco, CA (1988).

7 Heckerman D and Wellman M P: 'Bayesian Networks', Communications of the ACM, 38, No 3 (March 1995).

8 Singh M and Valtorta M: 'Construction of Bayesian Network Structures from Data: a Brief Survey and an Efficient Algorithm', International Journal of Approximate Reasoning, 12, pp 111—131, Elsevier Science, New York (1995).

9 Netica User Manual V1.05, Norsys Software Corp (1997) — http://www.norsys.com/

10 Cooper G F and Herskovitz E H: 'A Bayesian method for the induction of probabilistic networks from data', Machine Learning, 9, pp 309—347 (1992).

11 Rabiner L R and Juang B H: 'An introduction to hidden Markov models', IEEE Signal Processing Magazine, 3, Issue 1, Part 1, pp 4—16 (January 1986).

12 Li S, Sun B and Wilcox R T: 'Cross-selling Sequentially Ordered Products: An Application to Consumer Banking Services', Journal of Marketing Research, 42, No 2, pp 233—240 (2005).

Detlef Nauck is a Chief Research Scientist in BT's Intelligent Systems Research Centre in Adastral Park where he is currently leading an international team of researchers in Intelligent Data Analysis. He holds a Masters degree in Computer Science (1990) and a PhD in Computer Science (1994), both from the University of Braunschweig. He also holds a Benia Legendi (Habilitation) in Computer Science from the Otto-von-Guericke University of Magdeburg (2000), where he is a Visiting Senior Lecturer. He has published seven books and more than 80 papers and is a regular member of programme committees for conferences on computational intelligence. He has won two BCS gold medals and is an Associate Editor of IEEE Transactions on Systems, Men and Cybernetics — Part B. His current research interests include the application of Computational Intelligence, Intelligent Data Analysis and Machine Learning in different areas of Business Intelligence. His current projects include research in proactive intelligent data exploration and platforms for real-time business intelligence and predictive analytics in customer relationship management.

Dymitr Ruta received an MSc degree in Computer Science from the Silesian Technical University, Poland (1999) and subsequently a PhD in Pattern Recognition from the University of Paisley, Scotland (2003). After having completed a 5-month research studentship at Nottingham Trent University, where he developed an intelligent system for traffic flow prediction, he took up a PhD studentship challenge of developing high performance classifier fusion systems, which he successfully accomplished in 2003. During this time he held a Lecturer post at the University of Paisley where he was delivering mathematical and computing modules to BSc and MSc students. He joined BT in 2004 where he holds the post of a research scientist at the Intelligence Systems Research Centre in Adastral Park. His research interests include data mining, knowledge discovery, supervised and unsupervised learning, information fusion, evolutionary computation, physics of information, probabilistic and fuzzy reasoning. He has published more than 20 papers, and he is a regular member of programme committees for conferences in the areas of artificial intelligence and pattern recognition. He reviews publications for the reputable journals in these domains and actively participates in various research initiatives organised with BCS and NISIS.

Martin Spott received a Diploma (MSc) in Mathematics in 1995 from the University of Karlsruhe, Germany. He continued working in Karlsruhe until 2000 as a research assistant in the Innovative Computing Group of Prof G Goos. He completed his PhD in Computer Science in November 2000 with a dissertation on 'Reasoning with Fuzzy Terms' and joined BT in January 2001 where he works as a Principal Researcher in the computational intelligence research group. He has published numerous papers in his research area, and is also a regular member of programme committees for related conferences and a reviewer for scientific journals. His current research interests include soft computing, machine learning and data mining. Since joining BT, he has worked on several intelligent data analysis projects like travel time prediction, real time business intelligence tools and a platform for automating data analysis, for which he received a BCS medal in the category 'IT Developer of the Year — Infrastructure' in 2005.

Ben Azvine holds a BSc in mechanical engineeering, an MSc in control engineering, a PhD in intelligent control systems from Manchester University and an MBA from Imperial College, London.

Having held research fellowship and lectureship posts in several universities, he joined BT in 1995 to set up a research programme to develop and exploit soft computing and computational intelligence techniques within BT.

Since then he has held senior, principal and chief research scientist posts at Adastral Park where he currently leads the computational intelligence research group.

He has edited two books and published more than 100 scientific articles.

He is an inventor on 30 patents, has won two BCS gold medals, holds a visiting professorship at Bournemouth University, and visiting fellowships at Bristol and Cranfield Universities.

His current research interests include the application of soft computing to business intelligence, customer relationship management, intelligent data analysis and intelligent information management.

His current projects include building a soft computing platform for intelligent data analysis, developing a methodology for customer satisfaction modelling, developing decision support tools for universal service management, building intelligent information retrieval capability for future contact centres, and research into automatic identification of abnormal patterns from sensor data for health care.