

SCIENTIO

Beginner's guide to data mining, text mining, business rules and Scientio technologies

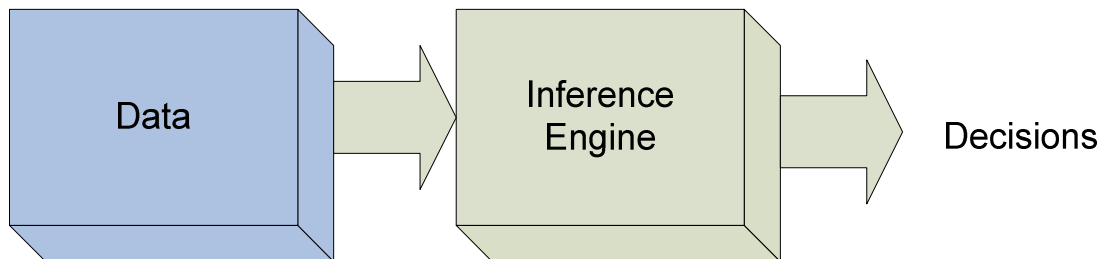
Andy Edmonds PhD, Scientio LLC

The business world is full of data. It comes in the form of emails, web pages, bank statements, RSS feeds, spreadsheets and many other forms.

Data mining is about making use of that data to automatically do something useful.

To be more particular, it's often assumed that data mining is about filtering and sorting data so that you, for instance, see only your most recent bank statement transactions, or just yesterday's emails.

Our definition is that data mining is about inferring or deciding something from data. So when you look at your bank statement and decide you're in trouble, or decide that a particular email message is spam, you're data – or text - mining.



So, data mining relies on a source of data, and some device that can make inferences from the data.

Now, computer programs make inferences about data all the time, and yet not all these programs can be described as data mining programs. What's the difference?

There are two main things that separate data mining type data analysis from general data analysis. The first is that data mining systems generally *learn* to make inferences from the data itself. We'll look at the forms of learning later. The other is that data mining systems are very sensitive to the uncertainty in making a decision, and are used where some uncertainty exists. Conventional programming systems can't cope with uncertainty.

So when your bank's computer decides you're overdrawn, there's no uncertainty present, and normal programming techniques can be used. If your bank decides that they ought to offer you a loan, however, things are much more uncertain, and they ought to be using data mining type analysis.

So, to put another slant on it, much of the data that is flowing around is precise and accurate, and existing well-understood algorithms exist for processing it. This is the world of normal programming.

Other data is imprecise, has elements missing, or comes from unreliable sources. There may also be no defined algorithm for processing it to get the results we want, even if the data is exact. This is the world where Scientio programs rule!

Data and text mining

Much of the data that businesses, engineers and scientists use is numeric, or consists of simple choices from a narrow range, like male/female, married/single/divorced/separated.

Data mining is concerned with this kind of data. Often there are longer pieces of text available that might be meaningful and useful, and text mining is concerned with making sense of these, like working out what the main topic of a piece of text is, or finding other documents containing similar text. While computers can't yet understand a piece of text the way that we do, they can do some surprising things using the frequency of particular words in a piece of text, and latterly by analysing the concepts within a piece of text using a model of the language. This kind of concept based text mining is also called concept mining.

Learning from data

There are three main forms of learning that data and text mining systems use.

Learning by example

This is also called supervised learning. It is where you already have a set of examples of the thing you want the system to learn, and can specify what is to be learned from the data, and what is to be used to learn from. An example of this might be credit checking. If you have a set of examples of loan applications, along with whether or not the customer repaid the loan, you would teach the system to recognise good risks.

To do this the system has to generate a model of the relationships between the loan application data and the outcome, and when you use the system to credit check new applicants you re-use this model.

Learning by inspection

This is also called unsupervised learning. It is where you have a set of data and you merely want to find interesting relationships within it, but have little or no preconceived ideas as to what the relationships might be. This is speculative, but often useful, data mining. One example is shopping basket analysis, where a store manager might discover that particular things turn up together in supermarket shopping baskets frequently. He/she might then sell more if they

moved them together in the shop, or linked them in their website. Again, the learning process creates a model that can be re-used on new customers or data.

Learning with a critic

This is also known as reinforcement learning. Sometimes instead of data you have a system that you want to improve, which has some elements that are variable. If you can model this system with a computer model and if you have some way of measuring whether one set of variables is better than another you can optimise the system with this range of techniques. The variables modified in this process can be simple, as in numeric values, or really complex like complete algorithms. In both cases, once again, a model is generated that can be reused later.

Modelling the solution

There are hundreds of techniques for data and text mining. Many PhDs (including the author's) have been granted for finding some new algorithm or variation that is particularly effective for particular data types. As a consumer of data mining you don't want to worry about this. And yet, to this day, there are still companies extolling their technology – like the fact they use Neural Networks, or Bayesian belief Networks, rather than the ability of the systems to solve the user's problems.

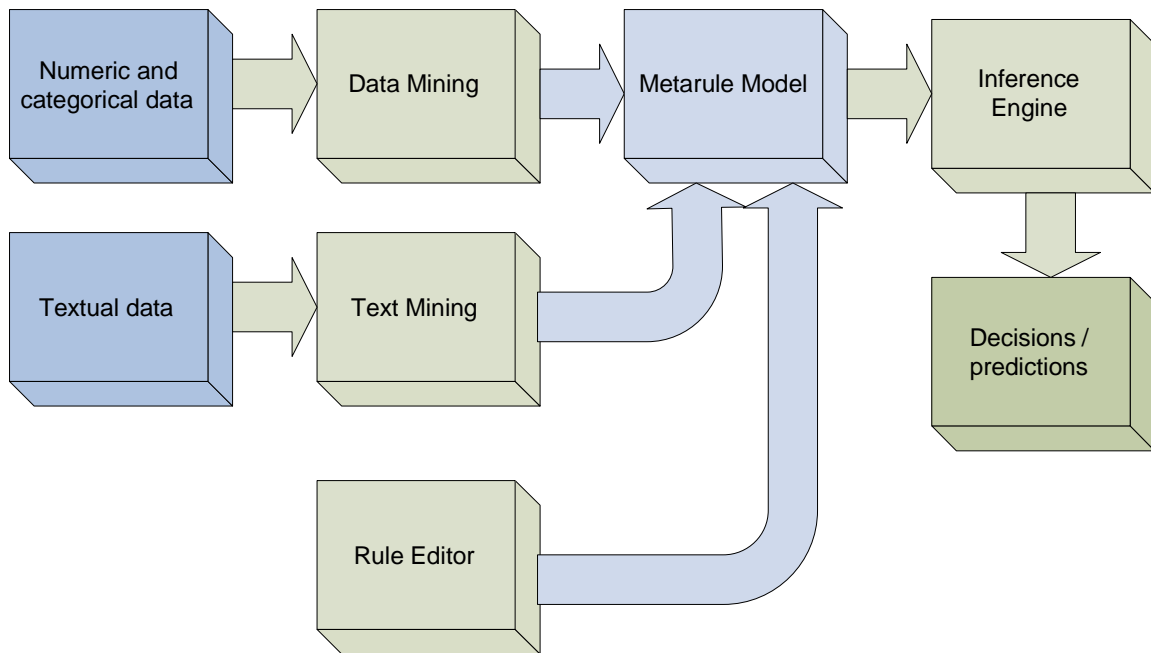
At Scientio we decided that in many cases the algorithms had developed to the point where they were mostly all good, and that the key thing was to develop a system that could be used for all the kinds of learning described above. The second and maybe more important consideration was that the model generated by the learning could easily be understood. Many of the learning systems in existence produce models that are incomprehensible to mere mortals. They may have learned something, but they can't tell you what it is.

If you want to describe to someone else how to make a decision or inference you are likely to use rules, unless the problem is mathematical, in which case you might use an equation.

Scientio therefore concentrated on producing learning algorithms that produced results in this format, and made sure that each could use the same inference engine.

We developed a language called Metarule which can contain both rules and equations. One of the really nice aspects of Metarule is that although our systems understand it, it's also really easy to convert it into English language if...then rules and equations.

As a spin-off from this we have created an editor so that you can create your own rules. The editor is so arranged that you can't make a syntax error. The system only lets you change things so that the resulting rules are syntactically correct. This is ideal for non-programmers who wish to rapidly modify the internal logic of a system. This kind of product is often called a business rules engine.



Another interesting and unique aspect of Metarule and the Metarule engine that reads and applies it to new data, is that it has special technology for handling, quantifying and representing the various uncertainties that crop up in the real world. Oh – and before I forget, Scientio’s data mining package can data mine and text mine simultaneously from different elements in the same data source, so the rules created can depend on textual and numeric data.

Forms of uncertainty

It’s interesting that, although most of the issues of our lives are bound up with uncertainty, (We spend a great deal of our time living for the future, and the future is always uncertain), we don’t learn very much about it, in high school, or even at university. Here are a few examples of forms of uncertainty:

Numerical uncertainty: if you want to buy a jacket for your partner – you may be both uncertain about their exact size and if the jacket is really the size it says it is.

Classification uncertainty: You’re interviewing someone for a job. Are they telling the truth on their CV? Is their experience relevant to the job?

Algorithmic uncertainty: You’re doing the first accounts for your new company. Should they be cash based or accruals based?

Ways of representing uncertainty

Our culture has fallen into using one main model of uncertainty. It's taught in schools, it's used on the news, the weather, doctors love it, and researchers in the social sciences are wedded to it. It's Probability Theory.

This is based, ultimately, on the idea that in order to predict the likelihood of something, we count the number of times it happened in the past in some small sample, and assume that it will occur in the same proportion in the future. If we get really sophisticated, we might try to put some bounds on the prediction based on the size of the sample. I.e. if the sample is small, it's logical to assume that the prediction will be very vague. Unfortunately, as soon as the statistics get used by our journalists, doctors, politicians, public planners and so on, those bounds get forgotten; but that's another story...

We'll look at the pitfalls in making predictions later. It's interesting to note that there is another definition of uncertainty in common use, which has its own mathematics, but which generally doesn't get taught at school.

A great deal of the time we reason using *plausibility* and *possibility*. So for instance in a court of law, many of the circumstances that crop up are very unusual, so probabilities are no help, and it would be unjust to convict or find for someone just because of them. Jurors and judges have to examine the explanations of the two sides for possibility, and then weigh them for plausibility. For instance in the recent Michael Jackson trial the jurors constructed their own timeline of the alleged events and decided that many of the assertions of the prosecution were just not physically possible. Again in law, in drawing up a contract a lawyer has to consider every plausible scenario and cover it. In designing a program a programmer has to consider all the plausible inputs and outputs a program might see, rather than just the most probable ones. Although in language we often talk of plausibility and possibility as absolute things, they are in fact relative – we can compare possibilities with other possibilities and plausibilities with other plausibilities.

Since the late 60's a great body of work has been assembled on Possibility Theory which is theory underpinning fuzzy logic, and it is this work that gives Scientio products their power.

Both these theories have their uses. It would be a bad idea to place a bet based on possibility theory – although people do this with lotteries all the time!

Similarly it's a bad idea to design a nuclear power station based on avoiding just the most probable things that might go wrong.

Scientio's products make use of both possibility theory and probability theory in order to learn, reason and infer from your data.

Pitfalls of data and text mining

I mentioned earlier that statistical inferences are often misused. It is often all too easy to find out that two things occur together, and to assume that they are related. For many years a graph of the numbers of storks in Holland kept beautifully in step with the UK birth rate, but this of course doesn't mean that storks bring babies!

These 'spurious correlations' crop up all the time. We seem to be continually besieged with health scares where, say, users of a particular well known drug seem to have more than the expected rates of some disease or other. Frequently, it seems, some other study shows them not to be correct. In many cases these reports are intended only for a medical audience and are taken out of context by journalists, but they illustrate the dangers of data mining. There should always be a plausible cause and effect link between the data and the inference.

Getting data mining wrong can be very expensive for a business. If you, for instance, optimise your website based on bad data mining, you may lose customers, or at least not make the gains you hope for. One way of guarding against such mistakes is to test your results on fresh data. The idea is to use two sets of data one of which you supply to the data mining system, and the other you hold back, the test data. You can then test the performance of the model you've created on the test data and find out if the performance holds up. Scientio data and text mining products automatically separate data into two randomly selected sets and learn from one followed by testing on the other. The user need only determine what percentage of the data goes for each use. Statistics on the performance on both the training and test set are provided.

The other major pitfall of data mining is to create a model, and then forget the fact that all the predictions of that model are likely to have some degree of uncertainty. The Metarule language holds the uncertainty data generated in the learning process, and every prediction generated by the Metarule inference engine has associated with it an uncertainty figure, calculated for that particular prediction, as well as bounds on numeric predictions, or alternatives when classifying. You can even do this with new input data. If a particular data source is noisy or unreliable you can specify this as part of the model re-use process.

Data sources

A few years ago there were more data formats than there were programs generating data. XML was invented to simplify all that. XML is a language for transferring data between computers, databases and programs. Almost all programs now support exporting data in XML. As well as carrying tabular data, XML can also carry very complex tree shaped data. Scientio's products are designed to data and text mine data in XML, and also to mine the structure of xml documents.

Conclusion

Data and text mining are concerned with finding useful relationships in data that can be re-used to do things better or avoid unwanted outcomes. There are three main kinds of learning employed to find these relationships, and a host of algorithms have been developed that might be used. Scientio have selected and developed algorithms that enable the user to produce results in the same form, no matter which kind of learning is employed. This form, encapsulated in our language Metarule, is based on rules and equations so that it can be easily understood by the system's users. It is also very simple to generate your own rules. However these rules are generated, they can all be re-used on new data with our Metarule inference engine. Scientio products are specifically designed to avoid generating unreliable predictions and to present the user with all the facts about the reliability of any prediction or inference generated.