



"turning data into dollars"

Tom's Ten Data Tips – January 2008

Data Mining Algorithms

Data mining algorithms come in many shapes and forms. Because the profession is so young, there is no agreed upon comprehensive algorithm taxonomy, yet. One distinction everybody agrees on, however, is supervised versus unsupervised algorithms (see tip #1). Most *new* data mining algorithms are developed in the Machine Learning community.

Another distinction that can be made is a sliding scale running from transparent to opaque algorithms. An algorithm that produces a "transparent" prediction is one where the relation between input variables (predictors) and the output variable is obvious and understandable. An "opaque" prediction is one where the prediction function can *not* be understood by humans. Examples of transparent algorithms are decision trees or regression. Examples of opaque predictions are neural networks or support vector machines.

1. Supervised Versus Unsupervised Algorithms

Supervised and unsupervised algorithms are sometimes also referred to as directed and undirected algorithms. These terms may be used interchangeably.

Examples of unsupervised algorithms are market basket analysis, clustering, Kohonen maps, association analysis, etc. What these have in common, is that they group sets of records to maximize *at the same time* within group similarity as well as between group disparity.

Examples of supervised algorithms are regression, neural networks, decision trees, support vector machines (SVM's), rule generators, etc. All these algorithms use a battery (array, set) of input variables, that are jointly related to an output variable. Although rarely used in practice, this may also be more than one output variable concurrently.

In summary, unsupervised algorithms serve to discover and describe associations, supervised algorithms are used to make predictions.

2. Unsupervised Algorithms Do *Not* Have An Optimal Solution

It holds for just about *any* unsupervised algorithm that there is *no* (response) function that can be minimized to objectively determine an optimal solution. What this implies is that you always need some *external* criterion to settle on 'the best' solution.

This distinction is not directly attributable to the fact that these algorithms are unsupervised; rather, these algorithms typically just don't have an unequivocal function to minimize.

3. Information Theory Is Underutilized In Algorithms

At the moment, pretty much the only use of information theory for data mining algorithms is in building decision trees (ID3, C4.5, C5.0 by Quinlan Ross). Information theory is a sub branch of probability theory that really got off the ground by Claude Shannon's seminal work in 1948. Another sub branch of probability theory (in itself relatively young) is statistics.

For some reason, almost all data mining algorithms are based on statistics, and *not* on information theory. Information theory has a lot to offer, like programming efficiency, evaluation of a data set's information content, and in particular when it comes to optimizing a model *across* data sets. This can make such models hold up much better in light of population drift, for example. Note that this particular feature is not available in Quinlan Ross' algorithms.

4. Data Mining (Still) Only Works On Flat Files

The way the current 'generation' of data mining algorithms operate, they *always* require a file structure with exactly *one* row per element in the universe, and all available columns appended to this (a "flat file"). For a few years now, academic researchers have been trying to develop what is called "relational data mining", aimed at mining relational database schemes.

This has so far not led to any substantial practical successes, or workable algorithms for practitioners. The principles look very promising but until a commercial application becomes available, the business relevance is negligible.

5. Decision Trees Have Many Advantages

Decision trees are an algorithm for recursive partitioning. What this means is that the training set is split from a parent node into child leafs. Then, for each subset arriving at each leaf this splitting process is repeated, hence "recursive". It's a *partitioning* algorithm in that each node in the tree splits the (sub)set of records arriving at that particular node among child leafs.

Decision trees have a number of advantages. One of them is that they are remarkably resilient to the curse of dimensionality. Another advantage is that decision trees can work with missing values in the data without the need for missing value replacement (although this still may be desirable). Most tree tools have little performance problems dealing with a data set with a very large number of columns. Decision trees are easy to understand, and can readily be explained to business people.

Since decision trees are made up of interactions, the discovered interactions can very well be used as input variables for a regression tool, greatly limiting the number of potential interactions to be considered, and quickly pointing to potentially *relevant* interactions.

The one drawback decision trees tend to have, is that predictive accuracy is usually not the highest attainable.

6. Neural Networks Are Probably The Easiest Way To Your Best Model

Neural Networks are often seen as the "ultimate" data mining algorithm. The metaphorical association with the human brain, and the opaque nature of the prediction add to the mysticism. It used to require considerable expertise from the miner to "tune" the network in order to come up with a good model. Much of the engineering required like preprocessing of nominal variables, determining the best topology for the hidden layer(s), and how to deal with missings has been largely automated in most tools nowadays. You don't need a PhD in stats anymore to come up with a good Neural Network model.

Because the predictive accuracy of a *very well* tuned Neural Network is often the highest available, this is certainly a technique to consider if an explanation for the prediction is not per se required, and getting the maximum accuracy is important. Given the current generation of user friendly Neural Network tools, even one based more or less on the default settings is likely to give a remarkably strong prediction.

7. Genetic Algorithms Are *Not* A Data Mining Algorithm

Contrary to many beliefs, genetic algorithms (also referred to as GA's) are *not* a data mining algorithm. Rather, GA's are an optimization procedure. Analogous to Darwinian selection, through breeding, cross-over and mutation an optimal solution can be searched in very large problem spaces. GA's are particularly effective at avoiding local maxima, and can be tuned to converge to a best solution within a predetermined time span.

In the past, GA's have been used, for example, to find an optimal set of regression coefficients. This can be useful because the number of interaction terms to consider can be daunting (approximately the squared number of variables divided by two for first order interactions *only*). A GA can control the search through that vast problem space.

8. OLAP is *Not* Data Mining, But It Is Related

Many Business Intelligence professionals equate OLAP (On-Line Analytical Processing) with data mining. This probably has to do with its exploratory nature. There's a key distinction, however. In OLAP you search for noteworthy patterns *within* predetermined dimensions. In data mining, you search for patterns across *all available variables*.

It is in this sense, that in OLAP you 'test' hypotheses within previously chosen dimensions that are likely to reveal interesting patterns. In data mining, you search for *unexpected* relations and you therefore make no assumptions which variables will be relevant for this.

One could say that data mining more or less *generates* hypotheses, and OLAP has its function to *test* hypotheses.

9. No Algorithm Performs Universally Best

There is often heated debate in the data mining community about which algorithm performs best. The fact of the matter is that whatever algorithm the miner is most familiar with is likely to give the best results, or at least, so it will appear.

For every algorithm, it is possible (and this has been demonstrated repeatedly) to produce a synthetic data set on which it will excel, and one that appears diabolically hard to predict. Since the nature of the data sets and the accompanying structure of problem space *for real life problems* are fundamentally unknowable, the best there is to do in practice is try a few different algorithms, and find out which one

performs best. This also shows why a good data miner needs to be versatile and technique agnostic.

10. Predictive Data Mining Models Require *Both A Prediction As Well As An Explanation*

Even in the 'extreme' case where explicitly only a predictive model is requested, it is nonetheless good practice to always accompany this with some form of explanation. This might be an explanation of the kinds of records that are targeted by the model, or maybe only the role that variables play in the prediction.

From a negative angle this serves as a sort of sanity check, to make sure the prediction actually makes sense. But from a positive vantage point, this will aid adoption and trust in the model, as well as foster insight in market dynamics.