Predicting the future, Part 2: Predictive modeling techniques

Alex Guazzelli

June 19, 2012

This is the second article of a four part series focusing on the most important aspects of predictive analytics. Part 1 offered a general overview of predictive analytics. This article focuses on predictive modeling techniques, the mathematical algorithms that make up the core of predictive analytics.

View more content in this series

Introduction

As a society, we are accumulating data on an exponential scale. IBM reports that 90 percent of the data available today was created just in the past two years. Fortunately, many predictive modeling techniques, including neural networks (NNs), clustering, support vector machines (SVMs), and association rules, exist to help translate this data into insight and value. They do that by learning patterns hidden in large volumes of historical data. When learning is completed, the result is a predictive model. After a model is validated, it is deemed able to generalize the knowledge it learned and apply that to a new situation. Given that predictive modeling techniques can learn from the past to predict the future, they are being applied to a myriad of problems such as recommender systems, fraud and abuse detection, and the prevention of diseases and accidents. The availability of "big data" and cost-efficient processing power is expanding the applicability of predictive data-driven techniques in different industries. In doing that, clever mathematics is helping more and more companies realize the true potential hidden in their data.

Predictive analytics is being used by companies and individuals all over the world to extract value from historical data obtained from people and sensors. People data includes structured customer transactions (for example, from online purchases) or unstructured data obtained from social media. Sensor data, on the other hand, comes from a barrage of devices used to monitor roads, bridges, buildings, machinery, the electric grid, and the atmosphere and climate. In this article, we focus on predictive modeling techniques. These are the mathematical algorithms, which are used to "learn" the patterns hidden on all this data.

After a predictive model is built and validated, it is deemed able to generalize the knowledge it learned from historical data to predict the future. In this way, for example, it can be used to predict the risk of customer churn or defection, in case of people data, or the risk of machinery breakdown, in case of sensor data. Models such as these compute a score or risk by implementing a regression function. Predictive models can also be used to implement a classification function, in which the result is a class or category.

No matter the type of model though, one thing is for certain: Predictive models are already shaping our experiences wherever we go and whatever we do. They recommend products and services based on our habits. They help healthcare providers design and implement preventive life saving measures given our susceptibility towards a particular disease.

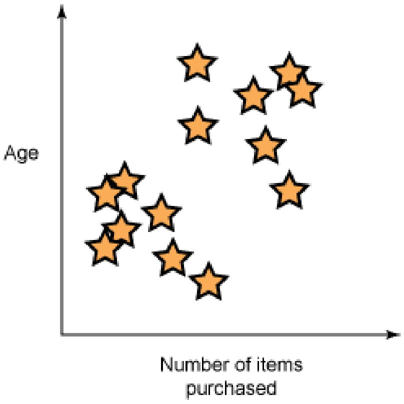
The birth of a predictive model

Predictive models are born whenever data is used to train a predictive modeling technique. To put it formally, data + predictive modeling technique = model.

A predictive model is then the result of combining data and mathematics, where learning can be translated into the creation of a mapping function between a set of input data fields and a response or target variable.

To build a predictive model, you first need to assemble the dataset that will be used for training. For that, a set of input fields representing a customer, for example, is assembled together into a record. This record may contain features such as age, gender, zip code, number of items purchased in the last six months, and number of items returned, combined with a target variable that may be used to inform us if this customer has churned or not in the past. A customer record can then be mathematically described as a vector in a multidimensional feature space, since multiple features are being used to define the object of type customer. When all customer records are assembled together, they become a dataset that may contain millions of records. Figure 1 shows a two-dimensional representation (using features "age" and "number of items purchased") of a few input vectors or customer objects.

Figure 1. Two-dimensional representation of input vectors in which each vector or customer object is represented by a yellow star



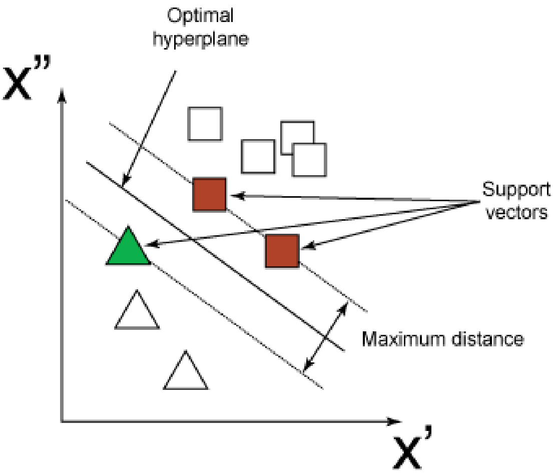
Predictive modeling techniques allow for the building of accurate predictive models, as long as enough data exists and data quality is not a concern. Bad data yields bad models, no matter how good the predictive technique is. And so the saying, garbage-in, garbage-out.

Common predictive modeling techniques

Today, a myriad of predictive techniques exist for model building. Different techniques are supported by distinct systems and vendors, but a half-dozen or so techniques are pretty much supported by most commercial and open-source model building environments. Although some are specific to a single class of problem, a few are generic and can be used for a variety of applications. Support vector machines (SVMs), for example, fall into this category.

An SVM maps input data vectors into a higher dimensional space, where an "optimal hyperplane" that separates the data is constructed. Two parallel hyperplanes are constructed on each side of this hyperplane. Figure 2 shows an example in which an optimal hyperplane is shown separating two data categories (triangles and squares). The optimal separating hyperplane is the one that maximizes the distance between the two parallel hyperplanes. The larger the distance between the two hyperplanes, the more accurate the model is assumed to be. The data points that lie on one of the two parallel hyperplanes that define the largest distance are known as the support vectors.

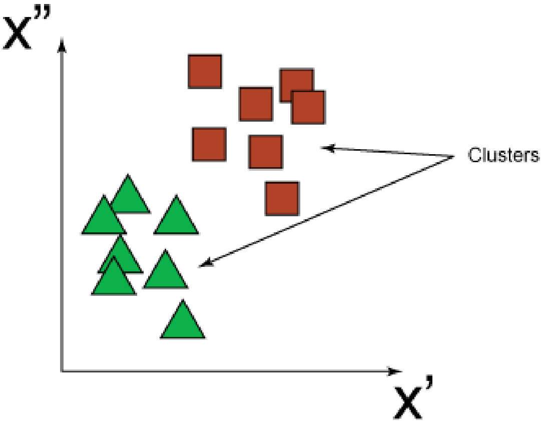
Figure 2. Two-dimensional view of an optimal hyperplane separating data and support vectors



SVMs, as well as NNs and logistic regression models, are powerful generic techniques that although mathematically different, generate somewhat comparable results. Decision trees represent yet another generic predictive modeling technique that stands out for its ability to explain the rationale behind the produced output. Because they are easy to use and understand, decision trees are the most commonly used predictive modeling technique.

Clustering techniques, on the other hand, are very popular whenever the target or response variable is not important, or not available. As the name suggests, clustering techniques are able to cluster input data depending on similarity. Figure 3 shows an example in which input data has been divided into two clusters. While the data in the first cluster is depicted with the use of green triangles, the data in the second cluster is depicted with the use of red squares.

Figure 3. Two-dimensional view of the result of clustering a set of input data into two clusters: green triangles and red squares

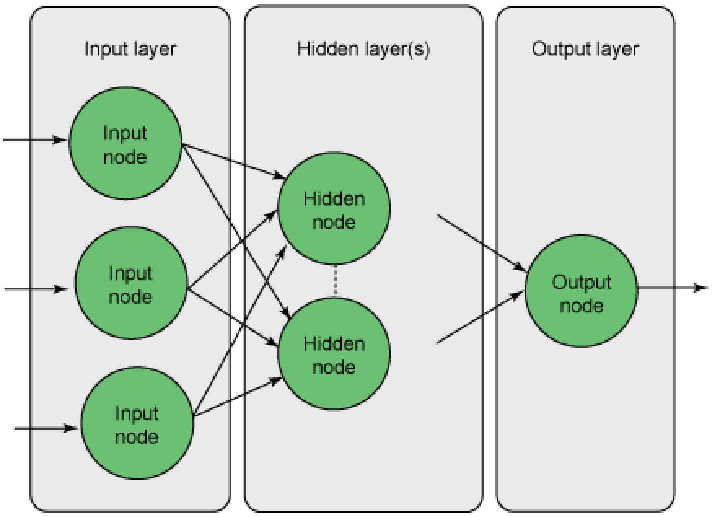


When a target variable or similarity measure is not important, but the associations between input items are, a technique known as association rules can be used to find them. For example, association rules can be used to discover that people who buy diapers and milk, also buy beer.

Although all predictive techniques have different strengths and weaknesses, model accuracy is very much dependent on the raw input data and the features used to train a predictive model. As mentioned above, model building involves a great deal of data analysis and massaging. Usually, from hundreds of available raw data fields, a subset is selected and fields are pre-processed before being presented to a predictive modeling technique. In this way, the secret behind a good predictive model often times depends on good massaging and more so than the technique used to train the model. That is not to say the predictive technique is not important. If the wrong technique is used, or the wrong set of input parameters is chosen, good data is not going to help.

NNs, for example, come in all shapes and forms. Selecting an appropriate network structure is important for building a good predictive model. As shown in Figure 4, feed-forward NNs are composed of an input layer, with as many nodes as the number of input fields and features being considered, and an output layer, which in case of a regression function is made up of a single node representing the predicted field. In between input and output layers though, the neural network may be configured with any number of hidden-layers and nodes. The problem here is that if you choose to give a NN too few hidden nodes, it may not learn the mapping function between the input fields and the target. Too many nodes and it will over fit, that is, it will learn the input data completely, but will not be able to predict future events.

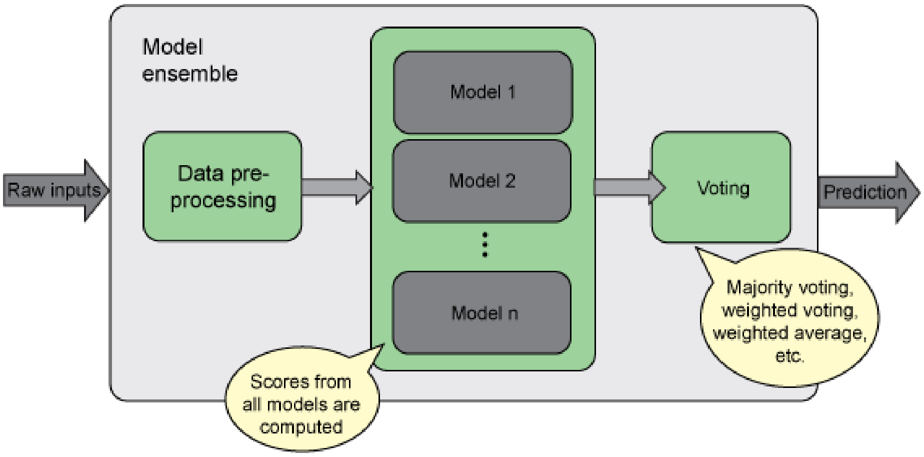
Figure 4. Feed-forward neural network with input, hidden, and output layers



Clustering techniques require that the number of clusters be provided before training. In this case, if the number of clusters is too small, the model may lose important differences in the input data, since it is being forced to bucket different data together. On the other hand, if the number of clusters is too big, it may miss important similarities. In the example shown in Figure 3, had the number of clusters been set to three instead of two, an extra cluster would have been created, which would probably have clouded up the true nature of the data (yellow triangles or purple squares?).

Predictive models can also benefit from different modeling techniques at the same time. This is because many models can be combined together in what is called a model ensemble (Figure 5). In this way, the output of the ensemble is designed to leverage the different set of strengths inherent to different models and techniques.

Figure 5. Diagrammatic representation of a model ensemble in which scores from all models are computed and the final prediction is determined by a voting mechanism or the average


Model ensemble
Raw inputs
Data pre-processing
Model 1
Model 2
Model n
Voting
Prediction
Majority voting, weighted voting, weighted average, etc.
Scores from all models are computed

Supervised vs. unsupervised learning

SVMs, decision trees, NNs and regression models use supervised learning to create the mapping function between a set of input data fields and a target variable. The known outcome is then used as a teacher who supervises the learning of her pupil. Whenever the pupil makes a mistake, the teacher provides her with the right answer in the hopes that the pupil will eventually get it right. For instance, when presented with a specific set of inputs, her output will match the target.

As an example, consider training an NN (shown in Figure 4) for predicting customer churn or defection due to attrition. We start by piecing together a set of input data fields that represent a particular customer who has churned in the past. It may consist of age, gender, as well as satisfaction-related features such as number of complaints. This customer, now represented by a set of data fields and the defection outcome, is then presented to the NN for learning. It may be presented multiple times until the NN is able to learn the relationship between input and target. However, this customer is not isolated. It is just one of many. The same process needs to be repeated for all customers, churners and non-churners. To learn to differentiate between the two possible outcomes, the NN will need to create an abstract representation for customers that did and did not churn.

A well-known learning algorithm used for feed-forward NNs is called back-propagation. It allows for the error, or the difference between target and output, to be propagated back through the network, which is then used to adjust the synapse weights linking the network nodes. In this way, the network eventually learns the task at hand, even if little by little. Without a target though, such process would not be feasible.

Unsupervised learning requires no teacher or target. Clustering techniques fall into this category. As shown in Figure 3, data points are simply grouped together based on their similarity. In case of customer churn, a clustering technique could potentially assign different clusters to churners and non-churners even though the outcome is not available during model training.

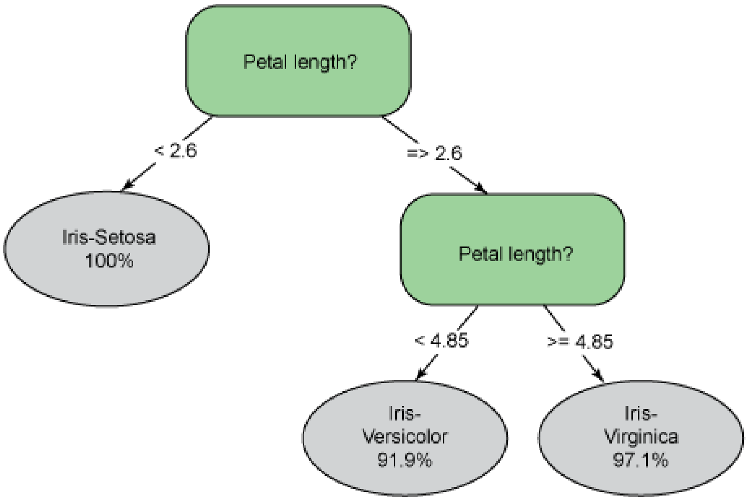
Black-box analytics

Black-box is a term used to identify certain predictive modeling techniques that are not capable of explaining their reasoning. Although extremely powerful, techniques such as NNs and SVMs fall into this category. Consider our highly accurate NN model, which was trained to differentiate churners from non-churners. If it outputs a high risk of churn for a particular customer, it will not be able to tell us why. This raises an important question: should a predictive model be able to explain its reasoning? Well, the answer might well be "it depends." In cases for which the risk generated by a predictive model is used to trigger an adverse action, an explanation is often desired and in some cases even required. For example, when a risk score is used to decline a loan application or a credit card transaction.

Whenever explaining is a must, you need to consider using a predictive modeling technique that clearly pinpoints the reasons for its decisions. Scorecards fit such a criteria very well. Based on regression models, scorecards are a popular technique used by financial institutions to assess risk. With scorecards, all data fields in an input record are associated with specific reason codes. During processing, data fields are weighted against a baseline risk score. After the fields with the highest influence on the final output are identified, their associated reason codes are then returned together with the output.

As with scorecards, decision trees are easy to explain and understand. In a decision tree, the whole decision process is represented by a set of human-readable conditions, that is, a set of rules. A leaf node in a decision tree is reached after a set of conditions evaluates to true. Figure 6 shows the graphical representation of a decision tree used to classify the Iris plant into three distinct classes based on petal length. Target classes are: Iris-Setosa, Iris-Virginica, and Iris-Versicolor. For more information on the Iris dataset, please refer to Asuncion, A. & Newman, D.J. (2007). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science (see Related topics). Note that the tree can be represented by a set of rules. For example, to identify Iris-Setosa plants, the rule would simply state: "If petal length less than 2.6 then plant is Iris-Setosa with probability of 1."

Figure 6. A simple decision tree used to classify the Iris plant. Possible classes are: Iris-Setosa, Iris-Versicolor, and Iris-Virginica



Although the reasoning behind the decisions generated by black-box modeling techniques are hard to explain, the models themselves should not be. Fortunately, representing data pre-processing as well as predictive models is now straightforward with PMML, the Predictive Model Markup Language. PMML is the de facto standard used by all the top analytic companies to produce and consume predictive solutions. As such, it allows for all predictive techniques mentioned in this article to be represented in a single, standard format. Once represented as a PMML file, a predictive model can be moved right away from the scientist's desktop, where it was developed, to the operational environment, where it is put to work. In this way, new models or any updates to existing models can be operationally deployed right away. As an open-standard that can be understood by all, PMML is used as a bridge not only between model development and deployment systems, but also between all the people involved in the analytical process within a company. In this way, it ensures transparency and disseminates knowledge and best practices. For more information about PMML, see Related topics.

Conclusion

An ever-expanding sea of data surrounds us and analytics allows us to navigate it safely. Historical data gathered from people and sensors is transforming our world, since it allows for the building of models that can literally use the past to predict the future. These so-called predictive models are, in fact, a product of clever mathematical techniques applied to data.

NNs, SVMs, decision trees, linear and logistic regression, clustering, association rules, and scorecards are the most popular predictive modeling techniques used by data scientists today to learn patterns hidden in the data. Although capable of learning and generalizing, these techniques are not only data hungry, but also tend to consume a lot of processing-power. Because of that, predictive solutions are only now experiencing a boom in all industries, due to the advent of: 1) big data derived from people and sensors; 2) cost-efficient processing platforms such as Cloud- and Hadoop-based; and 3) PMML, a refined and mature open-standard used to represent the entirety of a predictive solution. Combined, these three factors yield powerful models that can start making decisions right away, no matter the company size.

In fact, data scientists are hard at work building predictive solutions with the data we as a society are gathering in an ever-expanding pace. When combined with clever analytical techniques, this data gives us the potential to transform the world into a smarter world, where the prevention of crime, disease or accidents becomes a true reality, not just a prediction.

Related topics

- Follow your Rules, but listen to your Data: Watch Alex Guazzelli's presentation at the Rules Fest 2010 Conference which focuses on the differences between data-driven and expert knowledge as well as the benefits of bringing the two together.
- Predictive analytics in healthcare (Alex Guazzelli, developerWorks, November 2011): Read this article on the challenges and applications of predictive analytics in healthcare.
- The Heritage Heath Prize: Find out more about the highly publicized contest that aims to identify who will be admitted to a hospital within the next year, using historical claims data.
- What is PMML? (Alex Guazzelli, developerWorks, September 2010): Read this article on the PMML standard used by analytics companies to represent and move predictive solutions between systems.
- UCI Machine Learning Repository: Find the Iris dataset mentioned in this article.
- Predictive Analytics: Read the Wikipedia page on predictive analytics for an overview of common applications and techniques used to make predictions about the future.
- PMML in Action (2nd Edition): Unleashing the Power of Open Standards for Data Mining and Predictive Analytics (Alex Guazzelli, Wen-Ching Lin, Tridivesh Jena; CreateSpace, Jan 2012): Learn to represent your predictive models as you take a practical look at PMML.
- The Data Mining Group (DMG) is an independent, vendor led consortium that develops data mining standards, such as the Predictive Model Markup Language (PMML).
- Zementis PMML Resources page: Explore complete PMML examples.
- Data Mining: Find more about this topic in Wikipedia.
- PMML discussion group: Join this LinkedIn group.
- IBM ILOG: Learn more about this recognized industry leader in Business Rule Management Systems (BRMS), visualization components, optimization and supply chain solutions that enriches the IBM software portfolio and fortifies the IBM Smarter Planet initiative.
- developerWorks podcasts: Listen to interesting interviews and discussions for software developers.
- IBM SPSS Statistics 20 puts the power of advanced statistical analysis in your hands. Whether you are a beginner or an experienced statistician, its comprehensive set of tools will meet your needs.
- Try the IBM ILOG CPLEX Optimization Studio 90-day trial: Rapidly develop optimization-based decision support applications.
- Evaluate IBM WebSphere Application Server: Build, deploy, and manage robust, agile and reusable SOA business applications and services of all types while reducing application infrastructure costs with IBM WebSphere Application Server.