Fraud Detection with the SQL Server Suite

Author: Dejan Sarka
Reviewer: Matija Lah
July 2013
# Table of Contents

Management Summary .................................................................................................................. 3
Abstract ......................................................................................................................................... 5
Introduction ...................................................................................................................................... 5
The SolidQ Approach to Projects .................................................................................................. 7
Data Preparation .............................................................................................................................. 9
Data Overview ............................................................................................................................... 12
Data Mining Models ........................................................................................................................ 13
The Continuous Learning Cycle ..................................................................................................... 14
The Results ...................................................................................................................................... 14
Conclusion ...................................................................................................................................... 15
References ...................................................................................................................................... 16
About the Author ............................................................................................................................ 17
About SolidQ ................................................................................................................................... 17
Management Summary
The most significant element of SolidQ’s approach to Fraud Detection is the continuous learning cycle. We are focusing on using Microsoft SQL Server Suite as the fraud detection toolset because it includes all of the software one needs to create an appropriate fraud detection infrastructure. The service is performed as mentoring: actual work and consulting, together with the transfer of knowledge onto the customer’s employees. Once the project is completed, or sometimes even as soon as the proof-of-concept (POC) project is completed, the customer can deploy the fraud detection system into production. The way the infrastructure is set up supports continuous learning, which means that the customer is able to improve the system after deployment and address the ever changing circumstances of the particular business, throughout its use.

Two principal techniques are employed: supervised or directed models, and unsupervised or undirected models. Supervised data mining algorithms try to explain the value of the flags, with which the existing fraudulent transactions have been marked. When the patterns and rules that lead to fraudulent behavior are identified, they can be used to predict fraudulent behaviors in new transactions. Unsupervised techniques analyze data without any prior knowledge. With unsupervised models we establish a way to control the supervised ones, which together with OLAP models or DW reports represents the foundation for continuous learning infrastructure is established.

SolidQ suggests to start with a proof-of-concept (POC) project. It takes between 5 and 10 working days. Besides SolidQ’s data mining mentor (expert) the team should include a subject matter expert (SME) and at least one information technology (IT) expert. We can always replace the IT part of the team with SolidQ people; however, we cannot conduct a project alone, without a SME.

Steps:
- Training (optional, but strongly recommended) (1-2 days)
- Data Preparation (2-3 days together with the Data Overview)
  - Selection of data; Building computed variables; Sampling; Handling of missing values and outliers; Categorization.
- Data Overview (2-3 days together with Data Preparation)
  - Checking the distribution of variables; Finding dependencies between variables; the amount of information in variables (entropy).
- Building and Evaluating Data Mining Models (2 days)
  - Decision Trees; Neural Networks; Naive Bayes; Clustering
- Initial preparation of the continuous learning infrastructure (1 day)
- Presentation of results (1 day)

POC Benefits:
- The customer learns how fraudulent behavior manifests itself in operational data
- The customer’s employees learn how to perform the entire maintenance cycle on their own, which means that additional engagements by SolidQ would only be required in the unlikely event that the complexity of the problem grew unexpectedly
  - Analysts with appropriate subject matter expertise can perform additional in-depth analyses
  - IT experts can perform data extraction and preparation much more efficiently
  - Both groups of employees learn how to employ creativity to further improve the process and the procedures
- Improved data quality
- Improved employee job satisfaction, as each one of them can see how they could proactively contribute to the central knowledge about fraud patterns in the company
- With a larger number of employees involved in the enterprise the more and better fraud detection patterns can be developed
Abstract
With the massive usage of credit cards and web applications for banking and payment processing, the number of fraudulent transactions is growing rapidly and on a global scale. Several fraud detection algorithms are available within a variety of different products. In this paper, we focus on using the Microsoft SQL Server suite for this purpose. In addition, we will explain the SolidQ approach to solving the problem by introducing a continuous learning procedure. Our preferred type of service is mentoring; it allows us to perform the work and consulting together with transferring the knowledge onto the customer, thus making it possible for a customer to continue to learn independently.

This paper is based on practical experience with different projects covering online banking and credit card usage.

Introduction
A fraud is a criminal or deceptive activity with the intention of achieving financial or some other gain. Fraud can appear in multiple business areas. You can find a detailed overview of the business domains where fraud can take place in (Sahin & Duman E., 2010). Dealing with frauds includes fraud prevention and fraud detection. Fraud prevention is a proactive mechanism, which tries to disable frauds by using previous knowledge. Fraud detection is a reactive mechanism with the goal of detecting suspicious behavior when a fraudster surpasses the fraud prevention mechanism. A fraud detection mechanism checks every transaction and assigns a weight in terms of probability between 0 and 1 that represents a score for evaluating whether a transaction is fraudulent or not. A fraud detection mechanism cannot detect frauds with a probability of 100%; therefore, manual transaction checking must also be available. With fraud detection, this manual part can focus on the most suspicious transactions. This way, an unchanged number of supervisors can detect significantly more frauds than could be achieved with traditional methods of selecting which transactions to check, for example with random sampling.

There are two principal data mining techniques available both in general data mining as well as in specific fraud detection techniques: supervised or directed and unsupervised or undirected. Supervised techniques or data mining models use previous knowledge. Typically, existing transactions are marked with a flag denoting whether a particular transaction is fraudulent or not. Customers at some point in time do report frauds, and the transactional system should be capable of accepting such a flag. Supervised data mining algorithms try to explain the value of this flag by using different input variables. When the patterns and rules that lead to frauds are learned through the model training process, they can be used for prediction of the fraud flag on new incoming transactions. Unsupervised techniques analyze data without prior knowledge, without the fraud flag; they try to find transactions which do not resemble other transactions, i.e. outliers. In both cases, there should be more frauds in the data set selected for checking by using the data mining knowledge compared to selecting the data set with simpler methods; this is known as the lift of a model. Typically, we compare the lift with random sampling. The supervised methods typically give a much better lift than the unsupervised ones. However, we must use the unsupervised ones when we do not have any previous knowledge. Furthermore, unsupervised methods are useful for controlling whether the supervised models are still efficient.

Accuracy of the predictions drops over time. Patterns of credit card usage, for example, change over time. In addition, fraudsters continuously learn as well. Therefore, it is important to check the efficiency of the predictive models with the undirected ones. When the difference between the lift of the supervised
models and the lift of the unsupervised models drops, it is time to refine the supervised models. However, the unsupervised models can become obsolete as well. It is also important to measure the overall efficiency of both, supervised and unsupervised models, over time. We can compare the number of predicted frauds with the total number of frauds that include predicted and reported occurrences. For measuring behavior across time, specific analytical databases called data warehouses (DW) and on-line analytical processing (OLAP) systems can be employed. By controlling the supervised models with unsupervised ones and by using an OLAP system or DW reports to control both, a continuous learning infrastructure can be established.

There are many difficulties in developing a fraud detection system. As has already been mentioned, fraudsters continuously learn, and the patterns change. The exchange of experiences and ideas can be very limited due to privacy concerns. In addition, both data sets and results might be censored, as the companies generally do not want to publically expose actual fraudulent behaviors. Therefore it can be quite difficult if not impossible to cross-evaluate the models using data from different companies and different business areas. This fact stresses the importance of continuous learning even more. Finally, the number of frauds in the total number of transactions is small, typically much less than 1% of transactions is fraudulent. Some predictive data mining algorithms do not give good results when the target state is represented with a very low frequency. Data preparation techniques like oversampling and undersampling can help overcome the shortcomings of many algorithms.

SQL Server suite includes all of the software required to create, deploy any maintain a fraud detection infrastructure. The Database Engine is the relational database management system (RDBMS), which supports all activity needed for data preparation and for data warehouses. SQL Server Analysis Services (SSAS) supports OLAP and data mining (in version 2012, you need to install SSAS in multidimensional and data mining mode; this was the only mode in previous versions of SSAS, while SSAS 2012 also supports the tabular mode, which does not include data mining). Additional products from the suite can be useful as well. SQL Server Integration Services (SSIS) is a tool for developing extract transform–load (ETL) applications. SSIS is typically used for loading a DW, and in addition, it can use SSAS data mining models for building intelligent data flows. SQL Server Reporting Services (SSRS) is useful for presenting the results in a variety of reports. Data Quality Services (DQS) mitigate the occasional data cleansing process by maintaining a knowledge base. Master Data Services is an application that helps companies maintaining a central, authoritative source of their master data, i.e. the most important data to any organization. For an overview of the SQL Server business intelligence (BI) part of the suite that includes Database Engine, SSAS and SSRS, please refer to (Veerman E., Lachev T., & Sarka D., 2009). For an overview of the enterprise information management (EIM) part that includes SSIS, DQS and MDS, please refer to (Sarka D., Lah M., & Jerkić G., Training Kit (Exam 70-463): Implementing a Data Warehouse with Microsoft® SQL Server® 2012, 2012). For details about SSAS data mining, please refer to (MacLennan J., Tang Z., & Crivat B., 2009).

SQL Server Data Mining Add-ins for Office, a free download for Office versions 2007, 2010 and 2013, bring the power of data mining to Excel, enabling advanced analytics in Excel. Together with PowerPivot for Excel, which is also freely downloadable and can be used in Excel 2010, is already included in Excel 2013. It brings OLAP functionalities directly into Excel, making it possible for an advanced analyst to build a complete learning infrastructure using a familiar tool. This way, many more people, including employees in subsidiaries, can contribute to the learning process by examining local transactions and quickly identifying new patterns.
The SolidQ Approach to Projects

It is impossible to evaluate the time and money needed for a complete fraud detection infrastructure in advance. We, as SolidQ, do not know the customer’s data. We don’t know whether there is already an existing infrastructure, like a data warehouse, in place, or whether we would need to build one from scratch. Therefore, we always suggest to start with a proof-of-concept (POC) project. A POC takes something between 5 and 10 working days, and involves personnel from the customer’s site – either employees or outsourced consultants. The team should include a subject matter expert (SME) and at least one information technology (IT) expert. The SME must be familiar with both the domain in question as well as the meaning of data at hand, while the IT expert should be familiar with the structure of data, how to access it, and have some programming (preferably Transact-SQL) knowledge. With more than one IT expert the most time consuming work, namely data preparation and overview, can be completed sooner. We assume that the relevant data is already extracted and available at the very beginning of the POC project.

If a customer wants to have their people involved in the project directly and requests the transfer of knowledge, the project begins with training. We strongly advise this approach as it offers the establishment of a common background for all people involved, the understanding of how the algorithms work and the understanding of how the results should be interpreted, a way of becoming familiar with the SQL Server suite, and more.

Once the data has been extracted, the customer’s SME (i.e. the analyst), and the IT expert assigned to the project will learn how to prepare the data in an efficient manner. Our knowledge and expertise allow us to focus immediately on the most interesting attributes and identify any additional, calculated, ones soon after. By employing our programming knowledge, we can, for example, prepare tens of derived variables, detect outliers, identify the relationships between pairs of input variables, and more, in only two or three days, depending on the quantity and the quality of input data. We favor the customer’s decision of assigning additional personnel to the project. For example, we actually prefer to work with two teams simultaneously. We demonstrate and explain the subject matter by applying techniques directly on the data managed by each team, and then both teams continue to work on the data overview and data preparation under our supervision. We explain to the teams what kind of results we expect, the reasons why they are needed, and how to achieve them. Afterwards we review and explain the results, and continue with new instructions, until we either resolve all known problems.

Simultaneously with the data preparation the data overview is performed. The logic behind this task is the same – again we show to the teams involved the expected results, how to achieve them and what they mean. This is also done in multiple cycles as is the case with data preparation, because, quite frankly, both tasks are completely interleaved. A specific objective of the data overview is of principal importance – it is represented by a simple star schema and a simple OLAP cube that will first of all simplify data discovery and interpretation of the results, and will also prove useful in the following tasks. The presence of the customer’s SME is the key to resolving possible issues with the actual meaning of the data. We can always replace the IT part of the team with an appropriate SolidQ member; however, we cannot conduct this kind of a project without the customer’s SME.

After the data preparation and when the data overview is available, we begin the scientific part of the project. We assist the team in developing a variety of models, and in interpreting the results. The results are presented graphically, in an intuitive way. While it is possible to interpret the results on the fly, a much
more appropriate alternative is possible if the initial training was also performed, because it allows the customer’s personnel to interpret the results by themselves, with only some guidance from our mentor. The models are evaluated immediately by using several different techniques. One of the techniques includes evaluation over time, where we use an OLAP cube.

After evaluating the models, we select the most appropriate model to be deployed for a production test; this allows the team to understand the deployment process. There are many possibilities of deploying data mining models into production; at the POC stage, we select the one that can be completed quickly. Typically, this means that we add the mining model as an additional dimension to an existing DW or OLAP cube, or to the OLAP cube developed during the data overview phase. Finally, we spend some time presenting the results of the POC project to the stakeholders and managers.

Even from a POC, the customer will receive lots of benefits, all at the sole risk of spending money and time for a single 5 to 10 day project:

- The customer learns the basic patterns of frauds and fraud detection
- The customer learns how to do the entire cycle with their own people, only relying on SolidQ for the most complex problems
  - The customer’s analysts learn how to perform much more in-depth analyses than they ever thought possible
  - The customer’s IT experts learn how to perform data extraction and preparation much more efficiently than they did before
  - All of the attendees of this training learn how to use their own creativity to implement further improvements of the process and procedures, even after the solution has been deployed to production
- The POC output for a smaller company or for a subsidiary of a larger company can actually be considered a finished, production-ready solution
  - It is possible to utilize the results of the POC project at subsidiary level, as a finished POC project for the entire enterprise
- Typically, the project results in several important “side effects”
  - Improved data quality
  - Improved employee job satisfaction, as they are able to proactively contribute to the central knowledge about fraud patterns in the organization
  - Because eventually more minds get to be involved in the enterprise, the company should expect more and better fraud detection patterns

After the POC project is completed as described above, the actual project would not need months of engagement from our side. This is possible due to our preference to transfer the knowledge onto the customer’s employees: typically, the customer will use the results of the POC project for some time, and only engage us again to complete the project, or to ask for additional expertise if the complexity of the problem increases significantly. We usually expect to perform the following tasks:

- Establish the final infrastructure to measure the efficiency of the deployed models
- Deploy the models in additional scenarios
  - Through reports
Fraud Detection with the SQL Server Suite

- By including Data Mining Extensions (DMX) queries in OLTP applications to support real-time early warnings
- Include data mining models as dimensions in OLAP cubes, if this was not done already during the POC project
- Create smart ETL applications that divert suspicious data for immediate or later inspection
  - We would also offer to investigate how the outcome could be transferred automatically to the central system; for instance, if the POC project was performed in a subsidiary whereas a central system is available as well
  - Of course, for the actual project, we would repeat the data and model preparation

It is virtually impossible to tell in advance how much time the deployment would take, before we decide together with customer what exactly the deployment process should cover. Without considering the deployment part, and with the POC project conducted as suggested above (including the transfer of knowledge), the actual project should still only take additional 5 to 10 days.

The approximate timeline for the POC project is, as follows:

- 1-2 days of training
- 2-3 days for data preparation and data overview
- 2 days for creating and evaluating the models
- 1 day for initial preparation of the continuous learning infrastructure
- 1 day for presentation of the results and discussion of further actions

Quite frequently we receive the following question: are we going to find the best possible model during the POC project, or during the actual project? Our answer is always quite simple: we do not know. Maybe, if we would spend just one hour more for data preparation, or create just one more model, we could get better patterns and predictions. However, we simply must stop somewhere, and the best possible way to do this, according to our experience, is to restrict the time spent on the project in advance, after an agreement with the customer. You must also never forget that, because we build the complete learning infrastructure and transfer the knowledge, the customer will be capable of doing further investigations independently and improve the models and predictions over time without the need for a constant engagement with us.

Data Preparation

The problem of credit card fraud detection is not trivial. With every transaction processed, only a limited amount of data is available, making it difficult if not impossible to distinguish between a “good” transaction and a (potentially) fraudulent one. In addition, there are literally millions of points of sales and web sites where a single credit card can be used. Even additional properties that could be available in the card owner’s profile, such as demographical data, probably will not make things much clearer. Do we focus on the customer or on the credit card? From experience, it seems that the customers who use multiple credit cards typically use each card for a specific purpose. Although this means that we might start by profiling the card, it might also be worth checking the customer’s profile, as this might reveal different habits in different cultures.

It is also quite hard to request specific card properties and/or customer profile properties in advance. Different companies collect, maintain, and have access to different data sets. On the other hand, some
common data, like geographical location, time of usage, type of a product purchased, type of a transaction (purchase, cash advance), and similar, can be available to us. A good overview of data, useful for credit card and online banking fraud detection, can be found in (Hand D.J. & Blunt G., 2001).

In addition to the source variables, many calculated variables can be extremely handy. If geographic data is not available, it can often be extracted from IP addresses, ZIP codes, and similar source data. Web addresses also might contain country of origin, or, alternatively, business type. Universal product codes (UPCs) contain country of origin and the company that produces it. Many variables can be calculated from the time of the transaction and other data, for example:

- A flag designating whether multiple transactions have been issued from different IPs and the same person in a particular time frame
- A flag designating whether transactions from multiple persons and the same IP have been issued in a particular time frame
- Whether there are multiple persons using the same credit card or user account
- Whether the total amount of a transaction is near the maximum amount allowed for a particular type of transaction, or whether it is nearer the minimum amount
- The time of day could be significant: is the day a holiday, a weekday, or was the transaction issued on a weekend, or a particular day during the month
- The frequency of transactions in a moving time frame
- The number of distinct transactions in a moving time frame (often, the same kind of a transaction could be repeated regularly)
- The quantity of deviations from a moving average for the type of the transaction

We could address the problem with number of detected frauds that is too low in all of the transactions by oversampling, by repeating, or copying, known fraudulent transactions, or by undersampling, by lowering the number of non-fraudulent transactions in the sample used for model training. From experience we have learned to prefer the approach of undersampling. We select non-fraudulent transactions either with simple random sampling, or when we have clearly determined groups of transactions, with stratified sampling. For example, it might be obvious that there are significant differences in the patterns across different countries; by using countries or regions we can define different strata and then randomly select appropriate numbers of cases from each stratum separately. You can read more about sampling at (Wikipedia, n.d.). Different data mining algorithms are more or less prone to giving erroneous predictions when the target state is presented with a low enough frequency. With SSAS algorithms, we usually observe the following:

- The Microsoft Neural Networks algorithm works best when you have about 50% of frauds in the total sample data set
- The Microsoft Naïve Bayes algorithm already works well with 10% frauds
- The Microsoft Decision Trees algorithm even works well with only 1% of frauds
During the data preparation, we also have to take care of missing values and outliers. Missing values can have a seriously negative effect on a data mining project. However, if the number of missing values is small enough, they can be handled by using a variety of methods:

- **Do nothing** (a simple, but rarely a valid approach)
- **Filter out** the rows containing the missing data (note that we could also filter out too many rows and lose a pattern)
- **Ignore** the column (note that we could also ignore too many columns)
- **Predict** the missing values with data mining algorithms, like Decision Trees (note that we could lose variability)
- **Build separate models**, for example, one model for all the data (including missing values if the algorithm can handle them), one model for known data, and one model for rows with missing values (this does represent quite a lot of additional work)
- **Modify** the operational systems so that the missing values can be collected later (this represents the best alternative, but unfortunately it is also the most difficult to achieve)
- **Replace** the missing data with a mean (this is a very popular technique, although we could lose variability again)

Whenever we make any changes to the data, we are influencing the analysis. Before making any changes, we should determine whether there are any patterns in the missing data. We use data mining for this analysis. For instance, we add a flag with a value of 1 for a row that includes missing values or the value of 0 for rows where all of the variable values for the case are known and present. Then we use a predictive algorithm like Decision Trees to explain this new flag variable with other input variables. The resulting tree should be very shallow, without any strong patterns; otherwise, we have identified a pattern in the missing values. If a pattern has been found, it should be explained, and then we should use an appropriate missing value handling technique that does not alter the data (e.g., build separate models). Otherwise, we prefer to filter the rows with missing values.

Outliers are rare and far out-of-bound values. They are so far out of bound that they can influence the results of the analyses. Similarly to handling missing values, we determine whether there is any pattern in the outliers before handling them. We can do one of the following to address outliers:

- Check if the outlier is an erroneous value, and if is, correct it (the best possibility)
- Do nothing (a simple, but rarely a valid approach)
- Filter out the rows with the outliers (note that we could end up filtering out too many rows and lose a pattern)
- Ignore the column (note that we could ignore too many columns)
- Replace outliers with common (mean) values (note that, of course, we are losing variability)
- **Bin** values into equal height ranges (this is a good approach, especially for algorithms that use discrete input variables)
- **Normalize** the data values in predefined limited ranges

Some well-known methods of normalization include:

- Range normalization.
- Z-score normalization
• The logistic (sigmoid) function normalization
• The hyperbolic tangent function normalization

Discretization (or binning, or categorization, or recoding) is also useful for other purposes, not only for dealing with outliers. For example, some algorithms, for instance the Microsoft Naïve Bayes algorithm, can accept only discrete input variables. Discretization is performed on a single column. Examples of discretization include:

• Age
• Income
• Transaction amount

Note that with proper discretization we can compare otherwise incomparable data. For example, if one country income per capita is significantly higher than in another country, then it is difficult to compare the amounts of the transactions directly. However, if we discretize the income into three groups, like low, average and high, and appropriately tailor the group boundaries per country, we get comparable data. Please refer to (Pyle D., 1999) for further reading about data preparation for data mining.

Data Overview
As already mentioned, data overview activities interleave with the data preparation. In order to find outliers, we must get the idea of the distribution of a variable. We can use Microsoft Office Excel Pivot Tables and Pivot Graphs for this task. However, many times it is faster to use statistical computations and interpret the results. With Transact-SQL queries, we can calculate a lot of useful statistical information.

For a quick overview of discrete variables, we can use frequency tables. In a frequency table, we can show values, the absolute frequency of those values, absolute percentages, cumulative frequency, cumulative percent, and a histogram of the absolute percentage. OLAP cubes can be used to establish an overview of the frequency distribution for tens, if not hundreds of variables, very quickly.

For continuous variables, we can use descriptive statistics and calculate the first four population moments: Mean, Standard Deviation, Skewness, and Kurtosis. This gives us a quick impression of the distribution of values of those variables.

It is also worth checking linear dependencies between pairs of variables. Some algorithms, like the Microsoft Decision Trees algorithm, tend to exclude one variable from the dependent pair in the analysis, while other algorithms, like the Microsoft Clustering algorithm, might find too good a clusters if they use pairs of dependent variables. There are multiple methods for calculating these dependencies:

• Chi-Squared test for pairs of discrete variables
• Correlation Coefficient for pairs of continuous variables
• Analysis of variance (ANOVA) for pairs where one variable is continuous per one discrete variable

Based on our experience in the field of fraud detection, we developed a much faster method to test all possible linear dependencies between multiple variables, not just pairs. We use the Microsoft Naïve Bayes data mining algorithm, where we declare all variables as input and predictable at the same time; this way, we can determine all important dependencies with a single analysis. Of course, the Naïve Bayes algorithm expects discrete inputs only; however, SSAS can discretize variables on the fly, using different
discretization methods. We usually use the *Equal Heights* method, even though it is typically not useful for the final analysis, because it changes the shape of the distribution; however, it is extremely useful for the overview of linear dependencies, because it retains the maximum amount of *information* in the variables.

Variables with an insignificant amount of information are useless in analysis. We measure the amount of information in a variable by calculating the *Entropy*. As this calculation is quite slow, we do it only for variables, which we suspect contain a low amount of information.

For more details about the statistics mentioned, please refer to (Wonnacott T.H. & Wonnacott R.J., 1990). For more information on the information theory, please refer to (Kullback S., 1997).

**Data Mining Models**

We create multiple mining models by using different algorithms, different input data sets, and different algorithm parameters. Then we evaluate the models in order to find the most appropriate candidates for the actual deployment to production.

Many different algorithms can be used for fraud detection; it is difficult to say which one would generally yield the best result. In a project, the available algorithms are typically chosen, based on experience and the knowledge about the given domain. Because we use the Microsoft SQL Server suite, we use Microsoft Decision Trees, Microsoft Neural Network, and Microsoft Naïve Bayes directed algorithms, and Microsoft Clustering for the undirected one. In recent years, the Support Vector Machines methods are becoming more and more popular. SSAS does not bring this algorithm out of the box. However, it can be downloaded as a free *plug-in* algorithm for SSAS from the Microsoft CodePlex site at (Valkonet, 2008). Of course, if there are time and software policy constraints that prevent us from using this download, we simply skip it. We do not lose much, because according to (Sahin Y. & Duman E., 2011), the Decision Tress algorithm usually yields better results in fraud detection analysis than Support Vector Machines. For details on specific data mining algorithms, please refer to (Han J., Kamber M., & Pei J., 2011), or to the SolidQ course (Sarka D., Data Mining with SQL Server 2012, 2012).

We evaluate the efficiency of different supervised models by using standard techniques, namely the *Lift Chart*, the *Classification Matrix*, and *Cross Validation*. All of these techniques are built into the SSAS data mining feature and are described in more detail in (MacLennan J., Tang Z., & Crivat B., 2009). To evaluate the Clustering models, we have developed a technique of our own, implementing entropy. If the individual clusters are homogenous, the entropy in any given cluster must be low. We calculate the average entropy and the standard deviation of the entropy across all clusters. In a SSAS Clustering model that has been *trained* (or *processed*), it is possible to read the model data with DMX queries. In the *cluster notes* we can identify the distribution of the input variables, and then use it to can calculate the entropy.

From experience, we have learned that not all algorithms are equally useful for all data sets. The Microsoft Neural Network algorithm works best when the frequency of the target state (i.e. fraud) is about 50%. Microsoft Naïve Bayes can work well when the target state is represented by approximately 10% of the population or more. However, Microsoft Decision Trees work well even if the target state frequency is only about 1%, and is thus a very suitable algorithm for small data sets and low frequency of the target state as well.
The Continuous Learning Cycle

The continuous learning cycle is shown graphically in Figure 1.

![Diagram of the continuous learning cycle]

Figure 1: The continuous learning cycle

We start by creating the directed models, assuming that the customer has already flagged frauds in the existing data. We evaluate the directed models and then use the best one to predict the frauds in the new data. We also create the undirected models, evaluate them, and use the best one for selection of potential frauds. We do this over time and check the difference between the number or percentage of frauds caught with the directed and the undirected model deployed. When this difference drops, it is time to refine the directed model. In addition, we store the predictions of both models and the actual, confirmed or reported frauds in a data warehouse. When the percentage of the predicted frauds in the total number of frauds drops, it is time to refine both models. We use an OLAP cube on the top of the DW to measure the efficiency of the models over time.

The Results

Most of the financial companies already have some transaction checking and some constraints in place. However, even a small increase in the number of frauds detected using the same amount of checking can lead to impressive results. In one example, we managed to achieve a lift of 100 times the number of frauds found by selecting the rows to check with a directed model comparing to randomly selecting the rows to check. There was around 0.7% of frauds in all transactions. By selecting 10,000 transactions randomly, around 70 frauds were detected. By selecting 10,000 transactions with a directed model, 7,000 frauds were detected. Even with a controlling undirected model the lift was still 20 times – we caught 2,000 frauds in 10,000 transactions checked.

The result previously mentioned is quite impressive. However, this was so good that for some time we actually wondered whether there was an error in our system. Nevertheless, even if we lowered the lift to be closer to the average lift in our projects, we could still quickly calculate huge savings. In one example, with simple constraints in place, a credit card issuer managed to detect that something is wrong when checking transactions on line after an average of 8 fraudulent transactions per credit card with
approximately € 3,000 loss per credit card. We can suppose that 0.7% of the transactions were fraudulent. With half of the lift mentioned in the previous paragraph, with a lift of 50 times, we could find frauds on line after 2 fraudulent transactions with 70% probability. This would mean that we could prevent more than 4 frauds per credit card (6 frauds prevented with 70% probability = 6 * 0.7 = 4.2), thus saving approximately € 1,500 per credit card. For example, if the overall number of credit cards abused (stolen or any other way) would be only 10 per day, this would already mean € 15,000 of savings per day! Of course, the actual lift depends on the quantity and the quality of input variables, data quality in general, and the business knowledge available and used during the project.

Conclusion

Fraud detection is a very popular, albeit very complex, data mining task. In SolidQ, we have developed our own approach to fraud detection. The most significant element of this approach is the continuous learning cycle.

Although Microsoft SQL Server is not the most popular tool for data mining, we are using it. The SQL Server suite gives us all of the tools we need, and because all of the tools come from a single suite, they work perfectly together, thus substantially lowering the time needed to bring a project from the initial meeting to a production-ready deployment.

Another advantage of our approach is the mentoring with the knowledge transfer. It is not our intention to get permanent consulting contracts; we want to progress together with our customers. Once we finish the project, or sometimes even as soon as we finish the POC project, the customer can begin using and continue improving the fraud detection system constantly with the help of the continuous learning infrastructure.

Finally, due to Microsoft’s licensing policies, the customers that already possess Microsoft SQL Server Standard Edition or higher and Microsoft Excel, do not need to purchase any additional licenses.
References


About the Author

Dejan Sarka focuses on development of database & business intelligence applications with a strong focus on data mining. Besides projects, he spends about half of the time on training and mentoring. He is the founder of the Slovenian SQL Server and .NET Users Group. Dejan Sarka is the main author or coauthor of more than ten books about databases and SQL Server (among these Training Kits for Microsoft Exams 70-461 and 70-463). Furthermore he also developed courses for Solid Quality Mentors among them "Data Modeling Essentials" and "Data Mining with SQL Server 2012". As an MCT, Dejan Sarka speaks on many local and international events. Some of the international events include conferences such as PASS, TechEd, and DevWeek. He is also indispensable on regional MS events. In addition, he is a co-organizer of a technically top-level conference named Bleeding Edge.

About SolidQ

SolidQ sets the standard for information management services by providing the world’s most trusted and reliable data management services, business intelligence, collaboration and advanced technology solutions for Microsoft’s cloud and on premise platforms. As a Microsoft Gold Certified Partner with 10 Competencies (3 Gold, 7 Silver), the company employs many of the world’s top technical experts, including over 50 recipients of Microsoft’s Most Valuable Professionals (MVP) award or Regional Director recognition. These professionals serve more than 800 clients in 22 countries by helping them take advantage of the latest features for database and application deployments within Microsoft SQL Server, Business Intelligence, and .NET Development platforms. As key advisor and content provider to Microsoft, SolidQ experts have published more than thirty books and delivered countless speaking engagements at leading technical conferences around the world.