



Data Quality Control

Why you'd want a "novelty detector" in your ETL

Tom Breur
May 2009

Introduction

When a Data Warehouse (DWH) goes in production mode, the initial load has been reviewed and tested thoroughly. From here on you'll try to maintain a high level of quality moving forward.

Rather than waiting for end-users to question the content of your DWH "the hard way", we advocate a pro-active stance. Our recommendation is to constantly monitor data quality. Not only compliance with formatting, adherence to delivery processes, and referential integrity, but *also* data content. This way you can truly remain "in control" of your data.

In this paper we describe how to design test programs along with your ETL that provide monitoring of data content quality. This application that assesses whether new data are in line with historical feeds was previously labeled a "novelty detector" (Pyle, 1999), a term we will adopt here too.

Standard quality controls

Best practice for ETL is to have transparent SLA's with all data providers. You describe when files are to be expected, naming conventions for these files, file format, and record layout. Valid value ranges, referential constraints, and business rules that data should conform to (like an order amount can not be negative, for instance), are preferably part of these specifications. These agreements should also cover what to do in case some of the data aren't up to specifications. All these conventions support automating the ETL (and quality control!) process as much as possible.

The very act of writing these agreements down raises awareness that downstream processes (DWH or BI) are depending on timely and

accurate delivery of data. BI is increasingly a “mission critical” process in many companies, which needs to be supported by appropriate controls.

SLA's go a long way towards pre-empting unannounced changes in source system formats that might upset the flow of ETL processes. Are new columns added, existing columns dropped? Might the use or definition of certain fields be subject to change? Hopefully source system owners will realize that notifying their DWH partners will tremendously help them dealing with such changes.

Olson (2003, Data Quality – the Accuracy Dimension) has identified three categories of errors:

- Errors at the column level (invalid values)
- Errors due to inconsistencies *across* columns
- Business rule violations

But even if all of these quality controls have passed, the content may still be suspect. Some feeds may be expected to always grow and never shrink. This could be a dimension table of parts in use, or customers, etc. If it is a small dimension, a complete reload (instead of deltas) is not uncommon. If the count drops from one day on to the next, you know something is wrong. Monitoring for adherence to these rules happens during staging.

When all is said and done, sometimes, somehow, unannounced changes *will* flow through to the DWH. There are two scenario's now. Maybe the new data can not be read “automatically” by existing ETL programs. In this case you learn about these changes the hard way because an ETL batch job crashes. The alternative scenario is much more sneaky: although the data still adheres to the agreed upon format, the *content* is no longer valid.

In particular this last case of getting data that *appear* legitimate but have either implausible or downright misleading content is where a “novelty detector” has a chance to shine. Because you want to avoid having such technically correct but invalid data corrupt your DWH, a special purpose, proprietary indicator needs to alert you of the possibility that new feeds are suspect.

What makes data “suspect”?

Here we are talking about a data feed that appears to be correct at first glance. So in our dimension table with parts in use, the new feed

is indeed at least as long as the previous one. Now you cater to the expected *size of increase*, on a day to day basis. The number of added parts or customers has some “natural” variability. Some days you add no parts, on other days you add many. The question then becomes: how large a jump in number of parts is implausible?

If the feed is very consistent, from day to day, a small fluctuation may appear unusual. However, if a feed is characterized by large fluctuations, you won't be concerned if it moves around a bit. By “novelty” we mean *how much* a given feed is different from what might be expected. The way you assess “different”, of course needs to take into account historical fluctuations in numbers. Let's dust off our statistics books, here...

How to determine “novelty”

The thresholds that flag new data as “suspiciously different” from previous feeds need to be set with both type I and type II errors in mind. No matter where you set a threshold, you will always run a risk of either letting bad data in, or inadvertently stopping the ETL process for double checking, only to find that the data was correct after all. These two types of misclassification errors need to be balanced.

The good news here is that in practice usually feeds are either OK, or they go awfully wrong. In most settings where we have worked, there is a clear demarcation between fluctuations that are caused by “natural” processes, and which are aberrant. The former consist of genuine changes in the underlying primary process. The latter are the “novel” feeds we want to pick up, and which rightfully should cause the ETL process to halt.

With one of our clients we had this mechanism in place while the Euro conversion took place. On one of the columns the conversion had accidentally been performed twice. On a record level, such a change might be plausible, and would have probably not been noticed until later, much later. However, with this “novelty detector” it was picked up immediately, and the problem identified.

What you need to settle on is the amount of fluctuation from the “average” that should be flagged as suspect. To this end you calculate the historical variance of the time series of data feeds. When a feed contains a trend, that can be taken into account as well.

The variance (or standard deviation, its square root) is calculated from a sample, namely the feeds from the past days, weeks or months. Then you calculate a constant (typically around 2-3) times the standard deviation, and if the data lie outside that range (above or below), they are suspect.

Setting the threshold wide (roughly above 3) will make you accept most data. Setting it narrower (below 2) will cause more false positives: occasions where you were concerned about the data, but it turned out to be OK after all.

Building and running a program like this is a relatively small and simple effort and can nicely be integrated into existing ETL code. But the value is enormous as you will avoid loading poor data and acquire historical evidence on quality.

A commit and rollback may be straightforward for an OLTP system, but in the data warehouse we have no "simple" way of rolling back. By its very nature a data warehouse only gets additional data, existing rows are rarely if ever replaced by corrections (save type I slowly changing dimensions). That is why Bill Inmon talks about "non-volatile" data in the data warehouse.

Also, any of the reporting that may already have been run based on erroneous data will forever be in conflict with the contents of the DWH. Just when we were after "one single version of the truth"...

Dealing with "suspect" feeds

The question what to do after suspicious data have entered the ETL process is not straightforward. You have entered the imperfect world, and unfortunately there are no perfect solutions here.

Roughly three options exist when faced with suspect records. You can either:

- Halt the ETL process
- Put suspect records in quarantine, and process later
- Process questionable records but attach a "suspect" flag that is further elaborated in a dedicated audit dimension

Ralph Kimball (2004, the Data Warehouse ETL Toolkit) has outlined how to design an audit dimension. Elaborate qualification of the number and kinds of errors can be tied to any record this way.

Halting the ETL process obviously has its drawbacks. A manual restart is cumbersome. In particular if this happens too often, and for the wrong reasons. The advantage is that (hopefully) errors can be fixed before they enter the warehouse (can they?).

Putting records in quarantine for later processing may or may not work well. Some data warehouse models are better able to deal with partially missing records than others. In a "Kimball style" set of star schema's it will be a headache to properly assign keys for late arriving data (previously in quarantine). The Data Vault (Linstedt et al, 2008) is much better able to cope with this situation. Quite apart from this distinction, the quarantine queue needs to be managed.

Finally, tagging records as suspect, possibly imputing statically unbiased estimators, allows the ETL process to run uninterrupted. That is an obvious advantage. As a drawback, the query logic to include these qualifiers or constraints becomes a bit convoluted and less intuitive for end-users.

Conclusion

You can never take quality for granted. Besides testing data for conformance to standards as described in technical specifications, we recommend also testing for plausibility of *content*.

To this end we have described the use of a "novelty detector", a simple yet powerful program that can be seamlessly integrated into your existing ETL jobs. A novelty detector allows you to identify suspect feeds, although they conform to all provided specifications.

After suspicious data have been identified, you still need to decide how to deal with them. This can be either by fixing the data before they go in, or by tagging suspect data, and referring to an audit dimension that diagnoses all quality issues identified.

By constantly monitoring all data that enter your DWH you serve two purposes. First of all, tracing of issues allows you to acquire evidence and gives you the best possible opportunity to drive out root causes. Secondly, by staying on top of your "quality game", you will earn end-users' respect as a trusted source of reliable data, the "single version of the truth".

References

Dorian Pyle (1999) Preparing Data for Data Mining

Jack Olson (2003) Data Quality – the Accuracy Dimension

Ralph Kimball & Joe Caserta (2004) the Data Warehouse ETL Toolkit

Dan Linstedt, Kent Graziano & Hans Hultgren (2008) The New Business Supermodel – the Business Of Data Vault Modelling