SAS2079-2018

Do You Need to Put Your Data on a Diet?

Howard Plemmons, Priya Sharma, SAS Institute Inc.

ABSTRACT

You have amassed an amazing amount of data in your Hadoop environment in a relatively short amount of time. Unfortunately, you realize that all of this data cannot be processed in the environment in which it is contained. You have come up with alternatives that cannot be presented to the executive sponsors of this data rich environment. What's next?

Many have traveled the path that you are now on. This presentation will address the concept of how to do more with less by coming up with a diet plan for your data. Big data examples will be discussed along with analytical concepts used to trim the fat.

INTRODUCTION

Applying analytics after the data ingestion process is complete might require that you develop data organizational techniques to enable timely analysis. There are several reasons that might require techniques to enable data organization or reduction before analysis begins:

- Data ingestion was designed to capture volumes of data that were not based on a specific analytical outcome.
- Data volumes exceed the processing capacity of the environment to deliver results in a timely fashion.
- Data from multiple sources is required for a specific analytical model development.
- Reducing data to pertinent columns can significantly affect processing time.

For example, data from a recent financial services company engagement contained over 40,000 columns and 7 billion rows. The objective was to develop analytical models to provide insight into this data. To do so required a data diet.

The concept of diet is explained using two different processes. The first provides consideration for a standard diet process using information and techniques that exist in current best practices. The second uses more of an analytical method to identify data of value. In doing so, the result set or process set can be significantly reduced. The concept of analytic data identification is not new. However, the description here outlines a process we have used with many SAS® customers.

In general, the term diet implies loss that might not be equivalent to data diet. This, in turn, can imply organization by relevance, not data loss. To understand the concepts presented requires a basic understanding of the SAS language, SAS analytics, and general data processing techniques.

DATA DIET CONCEPTS

Before the “invention” of the term big data, diet techniques that helped organize data for analytics were common place. Some of those techniques included the following:

SAS DROP/KEEP – column reduction for both input and output processes. As your data grows, you might want to review the footprint on both input and output. Drop and keep can help speed up SAS Read/Write operations and reduce result set space requirements.

SAS VIEWS – provides the ability to establish a baseline of what the data set looks like. It enables you to control both columns and rows by establishing what will be processed by SAS. As with drop and keep, the view locks in the columns that will be used in the SAS Read/Write operations. This method also protects sensitive columns from the end user by eliminating the column from the SAS view.

PROC SQL – extends the reach into DBMS and other data systems that can be read by SAS. With access, a user could create DBMS views that, when used with SAS, would provide the same column and row characteristics seen with SAS.

Processing data using traditional SAS jobs requires that the initial data state is created or retained. The results of using these techniques can be data ready for analytics.

A consideration of many customers is how to manage the data footprint within their environments. Some have invested time to scan SAS libraries and report on when the last update was performed. You might also want to consider investigating the concept of last read as an identification method for SAS data at rest. This should supply you with more information to help with your disk space diet plans. For example, consider the following for SAS on Linux:

```
/*--- code snip from SAS Data Management Communities ---*/
filename oscmd pipe "ls -ul &filename";
data acctime;
infile oscmd;
length perms $10 links 3 user $8 group $8 size 8 month $3 day 3 ytime $5
  fname $100 year 3 time 8;
format time time5. acc_time datetime19.;
input perms links user group size month day ytime fname;
if length(ytime) = 4
then do;
  year = input(ytime,4.);
  time = 0;
end;
else do;
  year = year(date());
  time = input(ytime,time5.);
end;
acdate = input(put(day,z2.)||month||put(year,z4.),date9.);
acc_time = acdate * 86400 + time;
run;
```

The OS particulars to fuel the SAS code snip above are explained here:

https://unix.stackexchange.com/questions/8840/last-time-file-opened

You might need help from your SAS systems administrator or Linux system administrator to implement a comprehensive SAS data set diet plan. This example should give you an idea of some of the potential solutions to managing big SAS data:

A SAS customer was managing about 40 TB of SAS data stored on mainframe backup media. The task at hand was to take the SAS data from backups and port it over to a Hadoop environment. Rather than conducting an analysis of data usage in the existing SAS mainframe programs, the data and programs were ported with minimal change. The reason here was to store the SAS data in an environment where it could be accessed if needed.

There are options that could be applied in this situation to actually reduce the data foot print now in the Hadoop environment. For example, a mapping of the SAS code interaction with the data now in Hadoop could be produced. The legacy SAS jobs that use this code could be evaluated, which might lead to identifying data that is no longer needed or useful.

There is a cost—data reduction tradeoff—that must be considered. If you are dealing with legacy data and legacy programs, the effort to identify and eliminate unneeded data could be time consuming. Unfortunately, the fear of losing meaningful data will kill any data diet initiatives.

DATA DIET CONCEPTS – APPLIED TO DATA NOT FROM SAS

Unless you are moving to an in-memory strategy or in-database processing strategy, the data that you have in external sources might have to come to SAS for processing. While SAS can process many different data sources including databases, the concept of data diet applies here as well. The challenges of Read/Write come into play whether you are storing the data as SAS data sets or creating output in the data source not from SAS. We have seen customers replicate data in SAS that can cause some significant resource use problems and cast doubt on the freshness of the data. For data that is retrieved from these types of data sources and persisted as SAS data sets, consider the following:

- What is my strategy for purging/reusing the disk space after the analysis is complete?
- Will I create a data inconsistency issue by not refreshing the data or if the refresh fails?
- Do I need all the data that I will aggregate for the SAS job that I am executing?
- Will results be stored in the area that stores data not from SAS? If so, do I have resources there to hold the results of the analysis over time?

DATA DIET CONCEPTS – APPLIED TO BIG DATA ENVIRONMENTS

We have seen customers with significant investments in data capture, aggregation, and security in big data environments. Given that some have adopted the approach of capture it all and then figure out how to use it seems to impact many different business sectors. The awareness that we can't optimistically process all the data is slowly being realized.

The bright side of this conundrum is that you might not have to process all that is stored to reap competitive advantage from your data. Your requirement of moving models and processes beyond sampling to physical data can be realized. As with other data sources, diet techniques can be applied. However, preplanning data acquisition, storage, and usage can help avoid excess data:

- Usage processes are known before the data is positioned for use. This would typically be applied to data that you might create from data already stored in your big data environment. This would include techniques to aggregate, summarize, and produce result sets tuned for analytics.
- Physical reduction of data for use with analytics. This would include standards applied to the data for positioning in a data zone accessible by the end users. If analytical processes are known, then non-analytic data components can be eliminated or transformed into data that can be leveraged to produce the same result with a smaller footprint. An example would be taking common character data and creating lookup table that can reconstitute a row of data. The ultimate goal here is to reduce the data footprint.
- Applying techniques to large data to reduce the processing footprint. An example here would be processing data returned through a view that can restrict both column and row components. The view language could also be used to provide some pre-aggregation of the data before passing rows to the analytical process. Note that running in-database with DS2 can leverage this technique when running on a supported, big data environment.
- Most big data environments have a replication factor that protects the data but also leverages processing to data nodes that are under-used for specific data components. One thing that customers can forget is that the replication factor applies when transforming and saving large data tables. At times, this can create resource contention on the big data environment, which might require a return to techniques that do not replicate the data.

DATA DIET SUMMARY

The process defined in the diet sections above is somewhat atypical in many customer environments. We have seen some that need to clearly map intention before beginning and some that quickly hit resource thresholds on expensive data environments. Retooling the data to live within resource constraints after the fact can result in a very time-consuming process.

The next evolution in data diet involves more automation during the data discovery process, which might help you create a dynamic data preparation environment. Doing so can enable you to position the data for analytic consumption and at the same time reduce the fear of missed opportunity.

DATA DIET CONCEPTS – ANALYTICAL TECHNIQUES

Enormous amounts of data collected by organizations are causing the ability to handle, store, and analyze it to become increasingly complex. It has become more relevant to find data of value to gain fast and accurate insights to enable better decision making in a competitive market. Variable selection is an important step in big data analytics to improve performance without missing opportunities.

Variable selection methods can be broadly divided into supervised and unsupervised methods. This paper describes a few of the prevalent variable selection methods used while dealing with large volumes of data.

- *Unsupervised:* Descriptive statistics (PROC HPSUMMARY) describes the basic features of data like number of missing values, levels, mean, median, and so on. It can also be used to determine sparsity and high variance in data. This analysis can provide an insight into which variables to drop based on high missing percentage, unary values, and so on, all of which contributes to better understanding of data.
- *Unsupervised:* Hierarchical reduction based on frequency can be used when dealing with transactional data, which tends to be enormous in size. For example, transactional data can capture a customer's interaction along with time and date. This data can be very useful in providing a better understanding of customer behavior. This data can be transposed for each interaction type to create customer-level data. New features such as count of interactions in a particular time period can be created to get more granular insights. But this process can create very wide data. In this scenario, a hierarchical reduction based on frequency of each interaction in a time period can be done by using a suitable threshold to filter out lower frequency interactions.
- *Supervised:* The HPREG procedure is another high-performance procedure that can be used for variable reduction for large data using a forward, backward, or stepwise selection method, leveraging multistage screening option. In the multistage screening approach, the first stage consists of screening the regressors and selecting the model for the response from the screened subset. The second stage repeats the first stage except that you use the residuals from the first stage as the response variable in this second stage. You can iterate this process by using the residuals from the previous stage as the response for the next stage. The final stage forms the union of all the screened regressors from the first stage with all the selected regressors at the subsequent stages and selects a model for the original response variable from this union. Thus, for wide data, statistically significant predictors are kept for model fitting. In addition, PROC HPREG provides a variance inflation factor (VIF) that can be used to check multicollinearity. Variables with high VIF can be dropped.
- *Supervised and Unsupervised:* PROC HPREDUCE is a high-performance procedure that can be used for both supervised and unsupervised variable selection. It performs unsupervised variable selection by identifying a set of variables that jointly explain the maximum amount of data variance. It is also used for supervised variable selection by identifying a set of variables that jointly explain the maximum amount of variance contained in the response variables. This

procedure supports variable selection in both the regression setting and the classification (categorization) setting.

- *Supervised:* Selection methods in PROC HPLOGISTIC can be similarly used to select effects that should be used in the model, although PROC HPREG supports most complete sets of options.
- *Supervised:* High-Performance Decision Tree (PROC HPSPLIT) can be used to select most useful predictors based on variable importance. PROC HPSPLIT measures variable importance based on the following metrics: count, surrogate count, RSS, and relative importance.
- *Supervised:* High-Performance Forest (PROC HPFOREST) creates an ensemble of hundreds of decision trees to predict a single target of either interval or nominal measurement level. The HPFOREST procedure creates a tree recursively. An input variable is chosen and used to create a rule to split the data into two segments. The process is then repeated in each segment, and then again in each new segment, and so on, until some constraint is met. PROC HPFOREST can also be used to calculate variable importance based on loss reduction. A loss function is a statistic that measures how well a model fits data.

There are other analytical techniques to reduce the number of variables for model development, like correlation, variable clustering, weight of evidence, and so on. But these methods are often more suitable for smaller volumes of data with a manageable number of attributes.

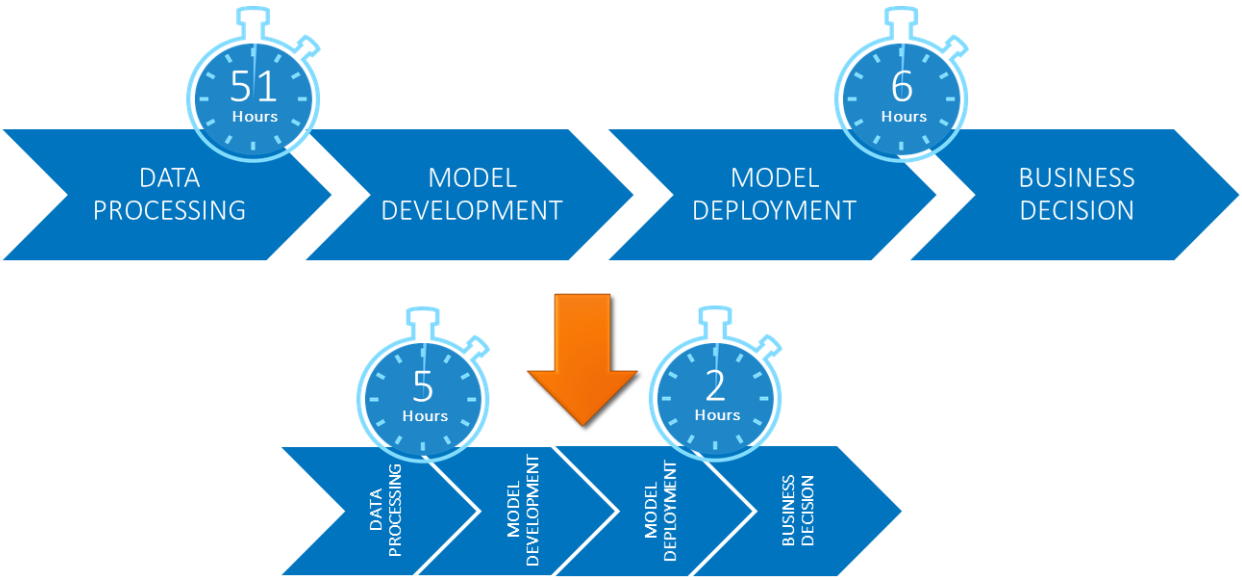
The methods discussed above fall into the variable selection category of dimensionality reduction. Another concept is variable extraction, and principal component analysis is one of the approaches. Both PROC PRINCOMP and PROC HPPRINCOMP perform principal component analysis (PCA). PCA is a multivariate technique for examining relationships among several quantitative variables. It provides an optimal way of reducing dimensionality by projecting the data onto a lower-dimensional orthogonal subspace that explains as much variation in those variables as possible. PCA reduces dimensionality by generating a set of new variables (variable extraction). But this technique is difficult to interpret, and you would still need thousands of attributes in order to score the model.

ANALYTICAL TECHNIQUES SUMMARY

The techniques discussed above can be used to identify suitable attributes in order to derive best value from big data. You can use one or more of the techniques to find subsets of attributes. An optimum number of operations and methodologies are dependent on data and the analysis required.

The diagram below shows significant reduction in processing and run times from a recent financial services company engagement with application of in-memory analytical techniques to reduce the number of variables and build models. The improvement in performance can lead to faster time-to-market efforts by leveraging the full population of data to gain better insights and increase profitability of marketing efforts.

In this scenario, the run time for the ETL process to create an analytical base table was reduced from 51 hours to 5 hours using in-database techniques, contributing to 90% improvement in data processing. Using the analytical methods described above reduced the number of attributes from 40,000 to 1,500 that went into the modeling process. Furthermore, variable reduction (PROC HPREDUCE) within SAS® Enterprise Miner™ selected 200 top variables, which were leveraged to run a modeling tournament to select the best model. The new best model selected increased model lift by 5% to 7%, with an estimated impact of $2 million per year for every 1% lift. Model deployment time was reduced from 6 hours to 2 hours, enabling daily scoring for faster identification of potential customers.



CONSIDERATIONS FOR SAS VIYA

SAS® Viya® is a cloud-enabled, next generation, in-memory analytics engine that delivers everything you need for quick, accurate, and consistent results – every time. Elastic, scalable, and fault-tolerant processing addresses the complex analytical challenges of today and effortlessly scales to meet your future needs.

SAS Viya is an underlying foundation for a host of solutions that take advantage of this cloud-enabled extension of the SAS platform. Most offerings include a coding interface as well as an intuitive, visual interface. Current SAS Viya products include the following:

- SAS® Visual Analytics
- SAS® Visual Statistics
- SAS® Visual Data Mining and Machine Learning
- SAS® Visual Forecasting
- SAS® Optimization
- SAS® Econometrics
- SAS® Visual Investigator

Similar techniques discussed in the data diet concept are also available in SAS Viya to perform both variable reduction and variable extraction. Some procedures for supervised and unsupervised variable selection are cardinality, decision trees (PROC TREESPLIT), logistic regression (PROC LOGSELECT), forest (PROC FOREST), clustering (PROC KCLUS), variable reduction (PROC VARREDUCE), principal component analysis (PROC PCA), and so on. In addition to the procedures above, there are new machine learning algorithms available in SAS Viya for dimensionality reduction: factorization machine model (PROC FACTMAC), robust principal component (PROC RPCA), PROC MWPCA, PROC FASTKNN, and variable clustering (PROC GVARCLUS).

CONCLUSION

Data acquisition, transformation, and analysis can provide you with significant competitive advantages. If attention to detail is not focused at the data level cost, value add and missed opportunities are almost a guarantee. As you evaluate how to organize data for analytical processing, keep in mind the data lifecycle. This can help you avoid some data pitfalls before and after the data is collected, organized, and used in your analytical processes.

We have seen data stored in Hadoop exceed many billions of rows and thousands of columns. To help meet data processing expectations, a careful examination and elimination of columns produces a better performing model. We have also seen unexpected growth in this environment require a resource allocation change in Hadoop to stay within expected processing times. This situation should be considered as you extend your model to more data stored in Hadoop.

REFERENCES

SAS Institute Inc. 2017. *SAS/STAT 14.3 User's Guide: High-Performance Procedures*. Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2017. *SAS Enterprise Miner 14.3: High-Performance Procedures.* Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2017. *Base SAS 9.4 Procedures Guide: High-Performance Procedures, Fifth Edition.* Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2017. *SAS® 9.4 and SAS® Viya® 3.3 Programming Documentation.* Cary, NC: SAS Institute Inc.

RECOMMENDED READING

- *SAS Enterprise Miner 14.3: High-Performance Procedures*
- *SAS/STAT 14.3 User's Guide: High-Performance Procedures*
- *Base SAS 9.4 Procedures Guide: High-Performance Procedures, Fifth Edition*
- *SAS® 9.4 and SAS® Viya® 3.3 Programming Documentation*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Howard Plemmons
Sr. Consulting Manager
Database Technologies Practice
SAS Institute Inc.
Cary, NC

Priya Sharma
Sr. Analytical Consultant
Analytics Delivery and Global Enablement
SAS Institute Inc.
New York, NY
Priya.Sharma@sas.com