Scoring with Analytic Stores

Merve Yasemin Tekbudak, SAS Institute Inc., Cary, NC

In supervised learning, scoring is the process of applying a previously built predictive model to a new data set in order to generate predictions for a target variable. Once the model has been built, you can save the model variables, effects, coefficients, and other descriptors so that they can be easily used for scoring.

For complex models, the traditional scoring code (which is created by the SAS DATA step) cannot adequately store the state of the analytic model because of the massive amount and variety of information that is needed. The analytic store has been developed as an alternative to the traditional scoring code.

An analytic store is a binary file in which you can save the information about the state of an analytic object, such as a predictive model after the training stage is completed. An important feature of the analytic store is its transportability, which enables users to produce an analytic store on one platform and use it on another platform without needing a traditional SAS export or import. Furthermore, the analytic store can be used to score models in a distributed environment or in-database with SAS® Scoring Accelerator for Hadoop, SAS® Scoring Accelerator for Teradata, and SAS® Scoring Accelerator for SAP Hana.

An analytic store can be generated by the following:

- HPSVM and HPFOREST procedures in SAS® High-Performance Data Mining 14.2
- HP SVM and HP FOREST nodes in SAS® Enterprise Miner™ 14.2
- FACTMAC, FOREST, GRADBOOST, and SVMACHINE procedures in SAS® Visual Data Mining and Machine Learning 8.1 on the SAS® Viya™ platform
- Forest and Support Vector Machine models in SAS® Factory Miner 14.2

The analytic store file can be used later by the ASTORE procedure for scoring. The ASTORE procedure is an interactive procedure that enables you to describe limited information about the analytic store and to score an input data set. You can produce an output data set by using an optional DS2 scoring code that uses the specified analytic store.

Scoring Home Equity Loans

The Home Equity Loan Data Set

The data set **Hmeq**, which is available in the SAS **Sampsio** library, includes information about fictitious mortgages for 5,960 mortgage applicants. A variable named **bad** indicates whether the customer has paid on the loan or has defaulted on it. Table 1 describes the variables in **Hmeq**.

Table 1 Variables in the Home Equity Data Set (**Hmeq**)

Variable	Role	Level	Description
bad	Response	Binary	1 = customer defaulted on the loan or is seriously delinquent 0 = customer is current on loan payments
clage	Predictor	Interval	Age of oldest credit line in months
clno	Predictor	Interval	Number of credit lines
debtinc	Predictor	Interval	Debt-to-income ratio
delinq	Predictor	Interval	Number of delinquent credit lines
derog	Predictor	Interval	Number of major derogatory reports
job	Predictor	Nominal	Occupational category
loan	Predictor	Interval	Requested loan amount
mortdue	Predictor	Interval	Amount due on existing mortgage
ninq	Predictor	Interval	Number of recent credit inquiries
reason	Predictor	Binary	`'DebtCon'` = debt consolidation `'HomeImp'` = home improvement
value	Predictor	Interval	Value of current property
yoj	Predictor	Interval	Years at present job

The following DATA step inludes an **id** variable, which is bound to the observation number _N_. The **id** variable is used to join the input records with their corresponding scores.

```
data Hmeq;
   set sampsio.Hmeq;
   id = _N_;
run;


proc print data=Hmeq(obs=10);
run;
```

Figure 1 shows the first 10 observations of the **Hmeq** data.

Figure 1 First 10 Observations of the **Hmeq** Data

Obs	BAD	LOAN	MORTDUE	VALUE	REASON	JOB	YOJ	DEROG	DELINQ	CLAGE	NINQ	CLNO	DEBTINC	id
1	1	1100	25860	39025	HomeImp	Other	10.5	0	0	94.367	1	9	.	1
2	1	1300	70053	68400	HomeImp	Other	7.0	0	2	121.833	0	14	.	2
3	1	1500	13500	16700	HomeImp	Other	4.0	0	0	149.467	1	10	.	3
4	1	1500	.	.			.	.	.	.	.	.	.	4
5	0	1700	97800	112000	HomeImp	Office	3.0	0	0	93.333	0	14	.	5
6	1	1700	30548	40320	HomeImp	Other	9.0	0	0	101.466	1	8	37.1136	6
7	1	1800	48649	57037	HomeImp	Other	5.0	3	2	77.100	1	17	.	7
8	1	1800	28502	43034	HomeImp	Other	11.0	0	0	88.766	0	8	36.8849	8
9	1	2000	32700	46740	HomeImp	Other	3.0	0	2	216.933	1	12	.	9
10	1	2000	.	62250	HomeImp	Sales	16.0	0	0	115.800	0	13	.	10

Scoring Data by Using an Analytic Store Generated by the HPSVM Procedure in SAS High-Performance Data Mining 14.2

The support vector machines (SVM) algorithm is popular in the data mining area of classification. The HPSVM procedure, a SAS Enterprise Miner high-performance procedure in SAS High-Performance Data Mining 14.2, executes the SVM algorithm in high-performance mode, which enables you to run the procedure on multiple threads in single-machine mode or distributed mode.

The following statements show how you can use PROC HPSVM in SAS High-Performance Data Mining 14.2 to produce an analytic store that can be used to score future home equity loan applications. The following statements execute the SVM algorithm on the **Hmeq** data set:

```
proc hpsvm data=Hmeq;
   input loan mortdue value yoj clage clno debtinc /level=interval;
   input reason job derog delinq ninq /level=nominal;
   target bad / order=desc;
   id id;
   savestate file='savehmeq1';
run;
```

The first INPUT statement defines the input variables **loan**, **mortdue**, **value**, **yoj**, **clage**, **clno**, and **debtinc** as continuous variables. The second INPUT statement defines the input variables **reason**, **job**, **derog**, **delinq**, **clage**, and **delinq** as categorical variables. The TARGET statement defines **bad** as the target variable and specifies the order of the target levels as descending. The ID statement specifies **id** as the record identifier. The SAVESTATE statement saves the state of the HPSVM procedure in the analytic store, which is stored in the table **savehmeq1**. You can use the analytic store in the ASTORE procedure to score new data.

Figure 2 shows that the analytic store file has been created successfully.

Figure 2 Log for the SAVESTATE Statement in the HPSVM Procedure

```
NOTE: For HPSVM target variable, the binary type is used.
NOTE: The HPSVM procedure is executing in single-machine mode.
NOTE: The HPSVM training process stopped because the minimal tolerance 1E-6 was
reached.
NOTE: Savestate successful. File "savehmeq1" has been written.
NOTE: There were 5960 observations read from the data set WORK.HMEQ.
```

The analytic store file can be generated for the HPFOREST procedure in SAS High-Performance Data Mining 14.2 in a similar way.

The following example shows how to score an input table by using the information in the analytic store. In this example, the input data table is **Hmeq**, the output data table is **scoreout1**, and the analytic store is in the data table **savehmeq1**, which was created by the HPSVM procedure previously.

The following statements use the ASTORE procedure to score the input data:

```
proc astore;
   score data=Hmeq out=scoreout1
         store='savehmeq1';
run;


proc sort data=scoreout1;
   by id;
run;
```

```
proc print data=scoreout1(obs=10);
run;
```

The SCORE statement enables you to score the **Hmeq** table. The OUT= option creates a new data table to contain the scored results. The STORE= option specifies the analytic store data table that was created earlier.

Figure 3 displays the scoring results for 10 observations. The P_BAD1 and P_BAD0 columns show predicted probabilities for each level of the target variable **bad**. The I_BAD column shows the final score of the target variable. An entry in the _WARN_ column shows that at least one input value is missing for the corresponding observation.

Figure 3 Scoring with PROC ASTORE

Obs	id	_P_	P_BAD1	P_BAD0	I_BAD	_WARN_
1	1	1.00002	0.00004	0.99996	0	M
2	2	1.00000	0.00004	0.99996	0	M
3	3	1.00002	0.00003	0.99997	0	M
4	4	-1.53736	0.75623	0.24377	1	M
5	5	1.00002	0.00003	0.99997	0	M
6	6	1.00001	0.00004	0.99996	0	
7	7	-0.99994	0.66666	0.33334	1	M
8	8	1.00001	0.00004	0.99996	0	
9	9	1.00001	0.00004	0.99996	0	M
10	10	1.00000	0.00004	0.99996	0	M

Scoring Data by Using an Analytic Store Generated in a SAS Enterprise Miner 14.1 or 14.2 Node

This section describes how to generate the analytic store files and score new data in nodes in SAS Enterprise Miner 14.1 and 14.2. When a process flow diagram in SAS Enterprise Miner includes an HPDM (High-Performance Data Mining) modeling node, applying the flow's score code to the data that reside in the distributed environment is supported by SAS Scoring Accelerator, either directly or via SAS® Model Manager.

In SAS Enterprise Miner 14.1 and 14.2, the Score node and Score Code Export node create the analytic store files that contain the score code when the Score node is preceded by the HP Forest or HP SVM node. In SAS Enterprise Miner 14.2, these files are also generated when a SAS Viya Code node preceding the Score node creates a model by using a procedure that supports an analytic store (the FACTMAC, FOREST, GRADBOOST, or SVMACHINE procedure). The Score node is then used to accumulate and apply the score code. The Score Code Export node exports files that are necessary for score code deployment. In SAS Enterprise Miner 14.1, the Score Code Export node writes the SAS scoring model program **score.sas** and the analytic store **score.sasast**. In addition to these two output files, the DS2 scoring model program **epscore.sas** is created in SAS Enterprise Miner 14.2. These files can be used by SAS Scoring Accelerator to deploy the score code in a database. You can find an example that includes the steps for using the analytic store files to publish and score a data set in the distributed Hadoop environment in this SAS High-Performance Analytics Tip.

Note that the analytic store files that are generated in SAS Enterprise Miner 14.1 contain the score code of the model only, whether it be from the HP FOREST or HP SVM node; in other words, they do not include score code from all preceding nodes in the SAS Enterprise Miner process flow diagram. For example, when you create a flow diagram which includes the HP Impute node (that produces DATA step score code) and the HP SVM node (that produces the analytic store file), you need to manually use the score code generated by the first node before passing it on to model score code. On the other hand, in SAS Enterprise Miner 14.2, the analytic store files contain the aggregated score code for an entire process flow diagram that includes an HP Forest, HP SVM, or SAS Viya Code node. The analytic store files **score.sasast** and **score.sas** that are created by SAS Enterprise Miner are located in the *<EM_Project>/Workspaces/EMWSx/HPDMForestx* folder for the HP Forest node and in the *<EM_Project>/Workspaces/EMWSx/HPSVMx* folder for the HP SVM node.

The following statements show how you can use the files from the Score Code Export node in a SAS Enterprise Miner 14.2 flow to score new data. The files are generated by the SVMACHINE procedure (which runs in the SAS Viya Code node) and are transferred to the Score Code Export node. Then the ASTORE procedure uses the files to score new data.

```
%let dir = %sysfunc(pathname(&em_lib))&em_dsep&&EM_METASOURCE_NODEID&em_dsep.Score;

proc astore ;
   score data=&EM_IMPORT_DATA out=score2
         epcode="&dir&em_dsep.epscore.sas"
         store="&dir&em_dsep.score.sasast";
run;
```

Scoring Data by Using an Analytic Store Generated by the GRADBOOST Procedure in SAS Visual Data Mining and Machine Learning 8.1

The GRADBOOST procedure in SAS Visual Data Mining and Machine Learning 8.1 creates a series of decision trees that together form a single predictive model called a gradient boosting model.

You can load the **sampsio.Hmeq** file into your CAS session by specifying your CAS engine libref in the second statement in the following DATA step (in this example, The CAS libref is named **mycas**):

```
data mycas.Hmeq;
   set sampsio.Hmeq;
   id = _N_;
run;
```

The following statements show how you can use the GRADBOOST procedure to build a gradient boosting model on the **mycas.Hmeq** data and save an analytic store for scoring data about new loan applicants:

```
proc gradboost data=mycas.Hmeq outmodel=mycas.GBmodel;
   input delinq derog job ninq reason / level=nominal;
   input clage clno debtinc loan mortdue value yoj / level=interval;
   target bad / level=nominal;
   id id;
   savestate rstore=mycas.savehmeq2;
run;
```

The OUTMODEL= option specifies the data table **mycas.GBmodel** to which the gradient boosting model is saved. The INPUT statements define the character variables as nominal inputs and numeric variables as interval inputs. The TARGET statement defines **bad** as the target variable and specifies the level of measurement of this variable as nominal. The ID statement specifies **id** as the record identifier. The **id** variable is required for scoring; it is used to merge the input data with their corresponding scores. The SAVESTATE statement creates an analytic store for the gradient boosting model and saves it as a binary object in the data table **mycas.savehmeq2**. The FILE= option that was specified in the SAVESTATE statement in PROC HPSVM is now replaced by the RSTORE= option in PROC GRADBOOST (as in all SAS Visual Data Mining and Machine Learning 8.1 procedures).

In the ID statement, the inclusion or exclusion of the target variable depends on your purpose for scoring data with ASTORE procedure. If you are scoring data that include the target for the purpose of assessing the fit of the model on the new data, then include the target variable in the ID statement. If you have new data that do not contain the target and you are using PROC ASTORE to make predictions on that data, then omit the target variable from the ID statement in the procedure that creates the analytic store.

The FACTMAC, FOREST, and SVMACHINE procedures use the same ID and SAVESTATE statements to create the analytic stores.

The following statements use the ASTORE procedure to score the input data:

```
proc astore;
   score data=mycas.Hmeq out=mycas.scoreout2
         rstore=mycas.savehmeq2;
run;
```

```
data scoreout2;
set mycas.scoreout2;
run;


proc sort data=scoreout2;
   by id;
run;

proc print data=mycas.scoreout2(obs=10);
run;
```

The SCORE statement enables you to score the input data table **mycas.Hmeq**. The OUT= option creates the output data table **mycas.scoreout2** to contain the scored results. The RSTORE= option specifies the analytic store data table **mycas.savehmeq2** that was created earlier.

Figure 4 displays the scoring results for 10 observations. The P_BAD1 and P_BAD0 columns show predicted probabilities for each level of the target variable **bad**. The I_BAD column shows the final score of the target variable. An entry in the _WARN_ column shows that at least one input value is missing for the corresponding observation.

Figure 4 Scoring with PROC ASTORE

Obs	P_BAD1	P_BAD0	I_BAD	_WARN_	id
1	0.32152	0.67848	0	M	5
2	0.70667	0.29333	1	M	13
3	0.98594	0.01406	1	M	21
4	0.96245	0.03755	1	M	29
5	0.92992	0.07008	1		37
6	0.97316	0.02684	1	M	45
7	0.01644	0.98356	0	M	53
8	0.84450	0.15550	1	M	61
9	0.97895	0.02105	1	M	69
10	0.99673	0.00327	1	M	77

The ASTORE procedure supports the DOWNLOAD and UPLOAD statements to transfer binary analytic store files only in SAS Visual Data Mining and Machine Learning 8.1. The DOWNLOAD statement extracts the analytic store that was produced by the GRADBOOST procedure from the CAS session and stores it to the local file **mycas.GBlocalcopy** in the local file system, as follows:

```
proc astore;
   download rstore=mycas.savehmeq2 store="mycas.GBlocalcopy";
run;
```

In contrast to the DOWNLOAD statement, the UPLOAD statement moves the analytic store **mycas.GBlocalcopy** from the local file system into the data table **mycas.savehmeq2** in CAS, as follows:

```
proc astore;
   upload rstore=mycas.savehmeq2 store="mycas.GBlocalcopy";
run;
```

For more information about the statements and options in the ASTORE procedure, see the chapter "The ASTORE Procedure" in *SAS Visual Data Mining and Machine Learning: Data Mining and Machine Learning Procedures*.

Scoring Data by Using an Analytic Store Generated in SAS Factory Miner 14.2

SAS Factory Miner 14.2 is a web-based data mining application that is especially designed for segmentation modeling. SAS Factory Miner 14.2 enables you to build models for each of the segments in your data by using algorithms such as decision tree, gradient boosting, neural network, random forest, support vector machine, and so on. Combined with a SAS Scoring Accelerator, SAS Factory Miner 14.2 models can be deployed and executed within a database or Hadoop environment to score new data.

The Random Forest and Support Vector Machine models in SAS Factory Miner 14.2 generate the DS2 scoring model program (**score.sas** file), the analytic store (**score.sasast** file), and (if the training data include user-defined formats) a format catalog that are required for analytic store scoring by SAS Scoring Accelerator. The scoring files are placed in a directory that is accessed by the scoring model macros. The macros take the files that are created by the HPFOREST or HPSVM components in SAS Factory Miner 14.2 and store them in-database with SAS Scoring Accelerator for Hadoop, SAS Scoring Accelerator for Teradata, and SAS Scoring Accelerator for SAP Hana. These scoring files are then used to score the model.

Summary

This paper shows how you can use an analytic store (a transportable binary file that contains information about the state of an analytic model) for scoring certain types of complex predictive models. This analytic store can be generated on different platforms and used later by the ASTORE procedure for scoring new data. For models in SAS Enterprise Miner and SAS Factory Miner that create an analytic store, SAS Scoring Accelerator can be used to perform in-database scoring for Hadoop, Teradata, and SAP Hana.

Additional Information

The ASTORE procedure in *SAS Visual Data Mining and Machine Learning 8.1: Data Mining and Machine Learning Procedures*
http://go.documentation.sas.com/#!/?cdcId=vdmmlcdc&cdcVersion=8.1&docsetId=casml&docsetTarget=viyaml_astore_syntax.htm&locale=en

SAS Scoring Accelerator in *SAS® 9.4 In-Database Products: User's Guide, Seventh Edition*
http://support.sas.com/documentation/cdl/en/indbug/69750/PDF/default/indbug.pdf

SAS High-Performance Analytics Tip #5: Scoring with Analytic Store Files
https://communities.sas.com/t5/SAS-Communities-Library/SAS-High-Performance-Analytics-tip-5-Scoring-with-Analytic-Store/ta-p/253544