1729-2014

How to Interpret SVD Units in Predictive Models?

Murali Pagolu, Analytical Consultant, SAS® Institute Inc., Cary, NC
Goutam Chakraborty, Professor, Department of Marketing, Spears School of Business, Oklahoma State University

ABSTRACT

Recent studies suggest that unstructured data such as customer comments or feedback can enhance the power of existing predictive models. SAS® Text Miner can generate SVD (Singular value decomposition) units from text documents which is a vectorial representation of terms in documents. These SVDs when used as additional inputs along with the existing structured input variables often prove to capture the response better. However, SVD units are sort of black box variables and are not easy to interpret or explain. This is a big hindrance to win over the decision makers in the organizations to incorporate these derived textual data components in the models.

In this paper, we demonstrate a new and powerful feature in SAS® Text Miner 12.1 which helps in explaining the SVDs or the text cluster components. We discuss two important methods useful to interpret them. For this purpose, we used data from a television network company which has transcripts of its call center notes from 3 prior calls of each customer. We are able to extract the key terms from the call center notes in the form of Boolean rules which have contributed to the prediction of customer churn. These rules provide an intuitive sense of which set of terms when occurring in either the presence or absence of another set of terms in the call center notes may lead up to a churn. It also provides insights into which customers are at a bigger risk of churning from the company's services and more importantly why.

INTRODUCTION

ABC Corp. (name changed due to the non-disclosure agreement with the party) is a national leader in telecommunication industry providing wide range of services such as television channels, high speed internet and home phone. They have a massive customer base with millions of households in the country. They are interested in understanding the factors which cause their customers to churn in the next 6 months. Initially, we built traditional models using structured numeric data extracted from their standard database. These models provided the much needed information for understanding the critical causal factors leading to a customer's churn. In the next phase, we incorporated the textual data from the customer call center notes the company maintains in a separate database into the existing models. Inclusion of the textual component in the traditional models proved useful in increasing the performance of the models in predicting the customer's propensity to churn. However, the executive committee of the company found these textual components as inexplicable. Thus, they turned down these models for the lack of transparency in the information used to build the model. In this paper, we discuss our approaches to interpret and explain those textual components so they can be used in the models and made transparent to top management.

Unlike the structured data, textual data is not a category or a continuous measurement data but is a collection of documents composed of words from a natural language such as English. In any real world business scenario, each document is likely very different from the others. Also, each document can be as simple as a noun group, a word phrase or as complex as a lengthy sentence. In a real world example, each document can have more than one sentence such as customer reviews or call center notes. Thus, a document collection can contain literally hundreds if not thousands of distinct terms. Think each of these distinct terms as variables and each observation represent a uniquely identifiable document in the collection. Now, the value for each of these variables may represent the number of times that term appears in that document. A transposed version of this structure is called a term-by-document frequency matrix in the literature. Representing the whole document collection in this kind of a structure can create a variety of problems as the number of variables required to represent all the documents will be really huge. Thus, we land up in the curse of the dimensionality.

SINGULAR VALUE DECOMPOSITION (SVD)

Singular value decomposition (or SVD) is a dimension reduction technique in which the textual data represented in a p x q term-by-document frequency matrix (A) is converted into a set of vectors in a k dimensional space. The value of k is less than r (rank of matrix A) such that,

$$A = U \Sigma V^T$$

U is an orthogonal matrix p x r where its columns are called left singular vectors.
Σ is a square diagonal matrix of dimension r where the diagonal elements are called singular values.
V is an orthogonal matrix r x p where its columns are called right singular vectors.

Albright (2004) explains how these left singular vectors can be imagined as the weights on the individual terms in the document collection. When these weights are projected on to the term frequencies for each of the documents, we get the SVD projections for the documents. These SVD components are simple real numbers and they can be included in the predictive models just like any other structured data elements. These SVD components when normalized on a unit scale will help in calculating the Euclidean distance between documents and also similarity scores in this context. These measures help in finding the sets of documents which are very similar. This process is called text clustering.

TEXT RULE BUILDER

A new feature is added to SAS® Text Miner 12.1 and is named text rule builder. As the name of the node suggests, it produces a set of rules using terms in the document collection stringed together by means of Boolean operators (AND, OR and NOT). These rules help in both predicting and explaining a target which can be a binary or a nominal categorical variable. An example of such rule can be explained as the presence of “Term A” and “Term B” but not “Term C”. These type of rules are typically combined at another level using a root operator such as ‘OR’. For example, the set of rules which may define the first level (Value ‘1’) of a binary target variable can be something as shown below.

“Term A” and “Term B” and not “Term C” (OR) “Term D” and Term “E” (OR) “Term F” and not “Term A” and not “Term B”

These rules are generated in a syntax that is acceptable in SAS® Content Categorization Studio. Hence, if the objective is to classify documents into known categories based on their contents, then generating these rules and reusing them perhaps with modifications is easy to do using this feature. This feature requires a training set of documents in order to learn the terms and the best combinations to identify the rules. In the following sections of this paper, we discuss how this feature will help in deriving the rules from the call center notes document collection and to identify the cluster membership and/or the

SVD components. This node offers four important configurable properties to train the data. We chose to run the node with default options. They can be described as shown below:

Variables – Select the categorical variable to be used as Target in the rule building process.

Generalization Error – Helps to prevent overtraining the data and reducing prediction errors. Selecting a higher value may prevent overtraining but also result in identifying not so useful generic rules.

Purity of Rules – Useful to control the maximum p-value required to join a term with a rule. Choosing a higher value generally results in very few but pure rules. Selecting a lower value may result in too many rules and a majority of them being useless.

Exhaustiveness – Helps in managing the extent of exhaustiveness to use in searching rules. Selecting a higher value may result in increased CPU time to run the node and over-trained rules.

METHODOLOGY

In this paper, we built several models which used inputs from either structured data only or unstructured data (SVD/cluster membership) only or both. The model comparison results using ROC on validation data as the selection criterion shows the best model as 'Reg2' – Regression with forward selection and Text Mining inputs (ROC:80.8%). The SVD measurements generated by the SAS® Text Miner are used as additional inputs along with the structured data as inputs to the model. When the SVDs are optimally binned against the target variable and used as inputs, the model performance didn't change (ROC: 80.8%). Hence, 'Reg4' – Regression with forward selection and optimally binned Text Mining inputs performs the same as 'Reg2'. However, when cluster memberships are used instead of SVDs or the optimally binned SVDs, the model performance dropped by few points (ROC: 78.3%). All other models show lower performances comparatively (See Table 1).

Table 1: Reference table providing the models compared and the inputs used respectively

Model	Model Description	Inputs Used	ROC
Reg2	Regression – Forward w TM	Structured + SVDs	**80.8%**
Reg4	Regression – Forward w OPT TM	Structured + Optimally binned SVDs	**80.8%**
Reg5	Regression – Forward w TM (Clus)	Structured + Cluster memberships	**78.3%**
TextRule	Text Rule Builder	Boolean Rules from Text Rule Builder	75.8%
Reg3	Regression – Forward only TM	SVDs Only	75.6%
Reg	Regression – Forward no TM	Structured data Only	75%
Ensembl	Ensemble of all the above models	-N/A-	73.8%

We can observe that the top 3 winning models are those which have inputs from both the structured and unstructured data elements. Also, the model built using text rule builder surprisingly performs slightly better than the Reg3 and Reg models. It should be noted that hybrid models are always better than text rule only models. Hybrid models predominantly contain measurable structured data inputs which can be used for reporting purposes often in graphical representation. They are also easy to implement from an operational and regulatory viewpoint. Customer profiles with distinguishing characteristics helps in building marketing campaigns targeted for those specific segments. In the next sections, we discuss two different methods to try and explain the text components of the models that are found significant in predicting the churn. Method I will be used to explain cluster memberships used in the model (Reg5) and Method II to explain SVD units used in the models (Reg2 and Reg4).

METHOD I: (When Cluster memberships are chosen as inputs)

We used the expectations-maximization clustering algorithm to generate disjoint sets of clusters from the call center notes. This procedure has produced 8 clusters and figure 1 shows the best descriptive terms defining those clusters. We can observe these clusters provide some information of how the customers may have felt about the company's services or explain the purpose of their call etc. For example clusters 1 and 5 shows terms such as account, credit, payment, bill, cci, charge, fee etc. which can be related to queries or concerns related to the billing. Content of the notes in cluster 8 appears to be related to the requests for change of address, e-mail, phone number etc. in the customer's profile.

Figure 1: Clusters generated on the textual data using expectations-maximization method

Clusters

Cluster ID	Descriptive Terms	Frequency	Percentage
1	+account +acct +amount +credit +disc +payment applied posted ...	3089	15%
2	'account number' +account +address +cancel +effective +email +immediately +number ...	1311	7%
3	+disconnect +fee +service disconnected moved moving services +account ...	1238	6%
4	+cancel +credit +order canceled cancelled charges hd oms ...	709	4%
5	+bill +cci +charge +credit +cust +customer +equipment +fee ...	4344	22%
6	+confirmation +date +dvd +letter +rec +scan ala dcn ...	2858	14%
7	+'rebate form' +cci +confirmation +cust +dvd +dvr +form +letter ...	1761	9%
8	'account number' +account +address +email +number +phone +ref +service ...	4745	24%

The cluster membership of each observation is output in the variable '**TextCluster_Cluster**' with 'Segment' as the role. We have changed this segment variable to an input variable and used it along with other structured data inputs in the regression model. Model results (See Figure 2) shows 5 out of the 7 dummy coded variables '**TI_TextCluster_cluster_n**' (where n ranges between 1 and 7) are selected as the significant effects in predicting churn. From figure 2, you can see the cluster membership variables representing clusters 2 and 3 (highlighted in red arrows) have higher odds ratio estimates with the modeling level of the variables 1 vs. 0.

Figure 2: Odds ratio estimates when cluster memberships are used in the regression model

Odds Ratio Estimates

Effect		Point Estimate
INV_tenure_months		3.655
OPT_credit_score	01:low-993.5, MISSING vs 02:993.5-high	1.074
OPT_tenure_months	01:low-0.5 vs 04:14.5-high, MISSING	0.955
OPT_tenure_months	02:0.5-10.5 vs 04:14.5-high, MISSING	0.612
OPT_tenure_months	03:10.5-14.5 vs 04:14.5-high, MISSING	1.270
TI_OPT_addl_cpe_cnt1	0 vs 1	4.700
TI_OPT_addl_cpe_cnt2	0 vs 1	7.083
TI_OPT_cpe_cnt1	0 vs 1	0.096
TI_OPT_cpe_cnt2	0 vs 1	0.631
TI_OPT_cpe_cnt3	0 vs 1	0.868
TI_OPT_delta_metric_011	0 vs 1	0.795
TI_OPT_delta_metric_012	0 vs 1	0.751
TI_TG_acct_type1	0 vs 1	1.113
TI_TextCluster_cluster_1	1 vs 0	0.807
TI_TextCluster_cluster_2	1 vs 0	4.452
TI_TextCluster_cluster_3	1 vs 0	1.302
TI_TextCluster_cluster_5	1 vs 0	0.451
TI_TextCluster_cluster_7	1 vs 0	0.318

It implies the customers who fall into either of these clusters have a higher chance to voluntarily churn from the company. Though there are descriptive terms explaining these clusters from figure 1, there is no easy way to identify the relationships between those terms or assess their contribution. Hence, we used the text rule expression builder in SAS® Text Miner 12.1 to identify a set of terms connected by AND, OR and NOT logical operators. For this purpose, the role of segment variable 'TextCluster_cluster' is changed to Target to run the rule builder. Figure 3 shows the partial results of the text rules generated by the node.

Figure 3: Partial output of Text Rules for each of the cluster memberships

```
, "moving" , "moved" , "moves" ))))
F_TextCluster_cluster_ =3 ::
(OR
, (AND, (NOT, (OR, "dates" , "date" )), (NOT, (OR, "offices" , "office" )), (NOT, (OR, "cancel" , "cancels" )), (OR, "disconnect" , "disconnected" , "disconnecting" ), (I
), (NOT, "ref" ), (NOT, "direct" ))
, (AND, (NOT, (OR, "canceling" , "canceled" , "cancelling" , "cancel" , "cancels" , "cancelled" )), (NOT, "disc" ), (NOT, (OR, "date" , "dates" )), (NOT, (OR, "number" ,
, "move" ), (NOT, "effective" ))
, (AND, (NOT, (OR, "canceled" , "cancel" , "canceling" , "cancels" , "cancelled" , "cancelling" )), (OR, "service" , "services" ), (NOT, (OR, "date" , "dates" )), (NOT,
, "rebates" )), (NOT, "tv" ), "rest" , (NOT, "net" ))
, (AND, (NOT, (OR, "cancelled" , "canceled" , "canceling" , "cancelling" , "cancels" , "cancel" )), (OR, "services" , "service" ), (NOT, (OR, "date" , "dates" )), (NOT,
, "rebates" )), (NOT, "tv" ), (NOT, "net" ), "warranty" )
, (AND, (OR, "services" , "service" ), (NOT, (OR, "date" , "dates" )), (NOT, (OR, "cancels" , "cancel" )), (NOT, (OR, "number" , "numbers" )), (NOT, (OR, "rebate" , "reb
, (AND, (NOT, (OR, "date" , "dates" )), (OR, "disconnecting" , "disconnected" , "disconnect" ), (NOT, "direct" ), (OR, "move" , "moving" , "moves" , "moved" ))
, (AND, (OR, "service" , "services" ), (NOT, (OR, "date" , "dates" )), (NOT, (OR, "cancels" , "cancel" )), (NOT, (OR, "number" , "numbers" )), (NOT, (OR, "rebate" , "reb
, "service calls" ), (NOT, "immediately" ), (NOT, "effective" ), (NOT, "net" ), (NOT, (OR, "switched" , "switches" , "switch" , "switching" )))
, (AND, (OR, "service" , "services" ), (NOT, (OR, "dates" , "date" )), (NOT, (OR, "cancels" , "cancel" )), (NOT, (OR, "number" , "numbers" )), (NOT, (OR, "rebate" , "reb
, "suspend" , "suspends" ), (NOT, "effective" ), (NOT, "net" ), (NOT, (OR, "switched" , "switches" , "switch" , "switching" )))
, (AND, (OR, "services" , "service" ), (NOT, (OR, "dates" , "date" )), (NOT, (OR, "cancels" , "cancel" )), (OR, "fees" , "fee" ), (NOT, (OR, "number" , "numbers" )), (NOT
, (AND, (OR, "out service" , "out services" ))
, (AND, (OR, "move" , "moving" , "moves" , "moved" ), (OR, "landlords" , "landlord" ))
, (AND, (OR, "service" , "services" ), (NOT, (OR, "dates" , "date" )), (NOT, (OR, "cancel" , "cancels" )), (NOT, (OR, "number" , "numbers" )), (NOT, (OR, "rebate" , "reb
, (OR, "switched" , "switches" , "switch" , "switching" )), (OR, "steal" , "stole" , "stealing" , "stolen" )))
```

RESULTS

As we know from figure 2, cluster members 2 and 3 represent significant number of customers who have higher propensity to churn. From figure 3, we have the text rules which provide us with a sense of the terms and their occurrences which make these clusters. So, we can use these Boolean rules to interpret those clusters and thus understand how the call center notes make sense in the model. Each rule shown on an individual line is again a combination of one or more terms with operators OR, AND, and NOT. All the rules found in each line are then joined by an operator such as an OR at the root level. Not all rules have the same precision when predicting the cluster membership. Thus we considered rules with best possible precision and also those which are applicable for a high number of observations. These statistics along with the classification matrix of the overall model are automatically generated by the text rule builder node. Since these rules are generated in a specific syntax, we found writing them in simple English sentences is easy to understand.

Cluster 2:

Term 'cancel' in the absence of the term 'scan' (or) Term 'effective' in the absence of the term 'date' (or) co-occurrence of terms such as 'service', 'detail' and 'select' (or) 'service' and 'suspend' (or) 'disconnect' and 'move' etc. When these are put together in a structure like the one shown in Table 2, then it is even more simplistic to interpret how and occurrence of which terms in the call center notes dictate the cluster membership which has more propensity to churn. Similarly, Table 3 represents notes in cluster 3.

Table 2: Boolean Rules representing cluster 2 in a tabular structure

Rule	Term	in the presence of term(s)	in the absence of term(s)	Precision
1	cancel	-	scan	**99.4%**
2	effective	-	date	99.3%
3	service	detail, select	-	97.5%
4	ref	phone, service	-	88.3%
5	ref	phone, move	-	96.4%

Cluster 3:

Table 3: Boolean Rules representing cluster 3 in a tabular structure

Rule	Term	in the presence of term(s)	in the absence of term(s)	Precision
1	disconnect	-	date, address, ref, direct, cancel, office, number, phone	**90.6%**
2	move	-	ref, date, phone, number, effective, cancel, disc	88.5%
3	disconnect	move	date, direct	86.5%
4	service	suspend	number, date, cancel, tv, rebate, effective, net, switch, immediately	82.0%
5	service	fee	number, date, cancel, tv, rebate, effective, net	77.5%

We also tried a very similar method with a small variation by treating Clusters 2 & 3 as Level 1 and clusters 1, 5 and 7 as Level 0 into a new cluster binary variable. This new category is now used as the target variable for generating the text rules. The reason is that the model results from figure 2 shows that the Odds ratio estimates of 1, 5 and 7 indicate they have a negative influence on churn while clusters 2 and 3 have strong positive influence. Hence, with this categorization Level 1 now represents cases with text notes indicating high propensity for churn and Level 2 indicating notes for low propensity to churn. This approach allows us to assess how the key clusters of call center notes identified significant in the model (1, 2, 3, 5 and 7) can indicate a true separation of churn vs. no churn cases in text rules. Figure 4 shows the results from this exercise. The results can be elucidated as shown below when rules with high precision and those which are applicable to maximum number of cases are considered.

Terms in clusters 2 and 3 which are indicative of a churn: service, disconnect, cancel, move, ref
Terms in clusters 1, 5 and 7 indicative of a no churn: form

Figure 4: Text Rules for cluster categories 1 and 0 (2, 3 vs. 1, 3 and 5)

```
F_Clus_Cat =1 ::
(OR
, (AND, (OR, "service" , "services" ))
, (AND, (OR, "disconnects" , "disconnect" , "disconnecting" , "disconnected" ))
, (AND, (OR, "cancel" , "cancels" ))
, (AND, (OR, "move" , "moves" , "moving" , "moved" ))
, "ref"
, (AND, (OR, "detail" , "details" ))
, "immediately"
, "effective"
, "yahoo"
, "final bill"
, (AND, (OR, "cancelled" , "canceled" , "cancel" , "canceling" , "cancelling" ))
, "complex"
, (AND, (OR, "subscriptions" , "subscription" )))
F_Clus_Cat =0 ::
(OR
, (AND, (OR, "form" , "forms" )))
```

Please note that this approach discards the other clusters which are found not significant when explaining the variance in the target variable (churn/no churn). However, you can see the key terms that surfaced in this and the earlier approach are very similar.

METHOD II: (When SVD units are chosen as inputs)

In this method, we suggest the best possible approaches to interpret the effect of textual data on a model when the SVD units are used in the model. As shown in Table 1, the best performing models are those which have structured data inputs along with the SVDs. In fact, both the winning models have the SVD units representing the textual data. Unlike the cluster memberships, SVD units bring better performance to the models but with additional complexity for interpretation. As we already know, SVDs are numerical representations of the text and there is no straightforward process to relate these numerical values back to the textual data.

Lu and Setino (1995) conceptualized a connectionist approach to a very similar problem they faced while interpreting a neural network model. They published a systematic and step-by-step approach which is also termed as de-compositional rule extraction method. This method allows us to understand and interpret how the classification mechanism of a neural network model works without the loss of much information. This method follows the following general steps:

a. Train and neural network model and prune it to remove redundant inputs without losing on its performance.

b. The hidden unit activation values which are continuous are discretized by means of clustering for each of the hidden units.

c. Identify the relationships between these hidden unit categories and the network output values and extract rules to describe the relationships.

d. Identify the relationships between the inputs and the hidden unit categories and again generate rules to describe the relationships.

e. Now, merge the rules generated from both the steps c and d to find the relationship between inputs and outputs.

This methodology proved to be very successful in interpreting the nature of the neural network models. We have adapted this approach to the current context where we have SVDs instead of hidden units which act as the black box variables. We followed the below steps to explain these SVD units.

1. Perform clustering on the call center notes from the prior 3 calls of customers.

2. Apply the optimal binning transformation over the SVD units to generate segments with high association to target.

3. Build the regression model using the structured inputs along with the optimally binned SVD units.

4. Identify the relationships between the transformed SVD units and the predicted value of the target variable using a decision tree.

5. Identify the relationships between the call center notes (textual data) and the optimally binned segments of the SVD units using the text rule builder node in SAS® Text Miner 12.1.

Now, merge the results from the steps 4 and 5 to identify the terms in the textual data connected by the Boolean operators with the predicted values of the target variable.

RESULTS

Decision tree using 'Gini' as the splitting rule criterion performed the best in fitting the model described in step 4 described above with sensitivity – 67.3%, precision – 85.29% and specificity – 95.35%. This is a reasonable model in terms of explaining the relationship between the SVD units and the predicted target variable (**I_vol_disco_ind**). Figure 5 shows an example English rule describing one of the purest leaf nodes classifying all the cases in that node as churn. All English rules from the decision tree can be then contracted such that it provides a final set of classification rules for both churn and no churn predictions in the total data.

Figure 5: One of the English rules from the decision tree with SVD inputs and predicted target

```
*------------------------------------------------------------*
 Node = 13
*------------------------------------------------------------*
if OPT_TextCluster_SVD5:01:low-0.0663805 IS ONE OF: 1
AND OPT_TextCluster_SVD2:01:low--0.535423 IS ONE OF: 1
AND OPT_TextCluster_SVD1:03:0.1605902-0.5503 IS ONE OF: 1 or MISSING
then
 Tree Node Identifier   = 13
 Number of Observations = 435
 Predicted: I_vol_disco_ind=1  = 1.00
 Predicted: I_vol_disco_ind=0  = 0.00
```

As a next step, each of the optimally binned SVD variables found in the English rules are assigned the role of target one at a time and text rule builder node is run to produce the rules which describes each of the bins in those SVDs. For example, role of the variable OPT_Text_Cluster_SVD1 which has 4 bins is changed to Target and text builder node is run to generate the Boolean expressions to explain each of the levels. From the result of the rule builder node, a subset of rules for the bin (high precision and reasonable number of observations) representing the variable OPT_TextCluster_SVD1:03:0.1605902-0.5503 is extracted. Similarly, the text rule builder node is used to produce Boolean rules for the variables OPT_TextCluster_SVD2 and OPT_TextCluster_SVD5. Tables 4, 5 and 6 shows these rules populated in the same way as we showed in Tables 2 and 3 from Method I.

Table 4: Boolean Rules representing OPT_TextCluster_SVD1:03:0.1605902-0.5503

Rule	Term	in the presence of term(s)	in the absence of term(s)	Precision
1	cancel	order	-	99.7%
2	cancel	scan	-	99.8%
3	account	wrong	-	99.1%
4	date	disconnect	-	96.5%
5	cancel	-	-	96.0%

Table 5: Boolean Rules representing OPT_TextCluster_SVD2:01: low -- 0.535423

Rule	Term	in the presence of term(s)	in the absence of term(s)	Precision
1	scan	-	-	98.7%
2	date	rec	-	96.8%
3	date	office	-	96.3%

Table 6: Boolean Rules representing OPT_TextCluster_SVD5:01: low -- 0.0663805:

Rule	Term	in the presence of term(s)	in the absence of term(s)	Precision
1	cancel	-	-	99.2%
2	ref	cancel	-	99.1%
3	cancel	-	fee	97.8%
4	address	detail	balance	98.8%
5	topic	-	-	99.2%

Finally, we merged all these Boolean rules from tables 4, 5 and 6 into the English rules in Figure 5. This gives us a better sense of what type of terms and their co-occurrences with other terms would cause a customer to churn or not churn from the perspective of textual data. This way, the so called black box SVD units can be explained to the decision makers in the organizations to convince them in implementing hybrid models (structured data + textual data). Thus, the description of that particular English rule can be something like “When at least one rule from each of the Tables 4, 5 and 6 occur in a customer’s call center notes, he/she is very likely to churn in the next 6 months.” Similarly, the same approach can be followed to describe those set of customers who are not likely to churn.

CONCLUSION

We can say the methodologies suggested in this paper can help in explaining the SVD units and text cluster memberships which may have a significant effect on predicting an event in a business problem. The new powerful feature in SAS® Text Miner 12.1 which helps in these methods plays a pivotal role in verifying the authenticity of the models involving components generated from textual data.

REFERENCES

Advanced Analytics for Customer Intelligence Using SAS® Course Notes. SAS Institute Inc., Cary, NC.

Albright R. (2004). Taming Text with the SVD. SAS Institute Inc., Cary, NC.

L. N. Trefethen and D. Bau, III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

Lu, H., Setiono, R., and Liu, H. 1995. Neurorule: A connectionist approach to data mining. In *Proceedings of the 21st International Conference on Very Large Data Bases (VLDB’95),* 478 – 489.

SAS® Enterprise Miner 12.1: Reference Help Documentation. Cary, NC.

TRADEMARKS

CONTACT INFORMATION

The authors welcome and encourage any questions, feedback, remarks, both on- and off-topic via email.

Murali Pagolu, SAS Institute Inc., Cary, NC, Email: murali.pagolu@sas.com

Goutam Chakraborty, Oklahoma State University, Stillwater OK, Email: goutam.chakraborty@okstate.edu