SAS761-2017

Breaking through the Barriers: Innovative Sampling Techniques for Unstructured Data Analysis

Murali Pagolu, SAS Institute Inc.

ABSTRACT

As in any analytical process, data sampling is a definitive step for unstructured data analysis. Sampling is of paramount importance if your data is fed from social media reservoirs such as Twitter, Facebook, Amazon, and Reddit, where information proliferation happens minute by minute. Unless you have a sophisticated analytical engine and robust physical servers to handle billions of pieces of data, you can't use all your data for analysis without sampling. So, how do you sample textual data? The standard method is to generate either a simple random sample, or a stratified random sample if a stratification variable exists in the data. Neither of these two methods can reliably produce a representative sample of documents from the population data simply because the process does not encompass a step to ensure that the distribution of terms between the population and sample sets remains similar. This shortcoming can cause the supervised or unsupervised learning to yield inaccurate results. If the generated sample is not representative of the population data, it is difficult to train and validate categories or sub-categories for those rare events during taxonomy development. In this paper, I show you new methods for sampling text data. I rely on a term-by-document matrix and SAS® macros to generate representative samples. Using these methods helps generate sufficient samples to train and validate every category in rule-based modeling approaches using SAS® Contextual Analysis.

INTRODUCTION

Large volumes of data require robust hardware and advanced analytics software to execute algorithms that can crunch numbers, perform complex computations, and deliver results in a matter of minutes or even seconds. When it comes to structured data, a million records and several hundred variables are not necessarily considered big data nowadays. However, with unstructured data, it is a different ball game. A few hundred thousand data records of textual data requiring some sort of exploratory or predictive analysis can demand lot of CPU horsepower, capable hardware, and software to support high-performance computing with parallel processing. SAS has a wide range of high-performance computing offerings and in-memory offerings that can cater to such needs. But not every business is ready to move there yet. So, how do we deal with this volume of data in such situations? Statistical sampling to the rescue. You can perform sampling to generate smaller representative data samples before you begin any type of analysis. But here is the tricky part. When you perform sampling, it is expected that the distribution of the data elements in the sampled data is approximately close to the distribution in population data. Sampling on unstructured textual data might not effectively generate a representative sample simply because the sampling process (either random or stratified) doesn't take into account whether the distribution of terms in the sampled data set is reasonably representative of the original data.

In this paper, I propose some innovative methods you can use to improve the situation, giving your sampled data a much better chance of representing your population data. SAS® uses a term-by-document matrix as the basis for the statistical analysis of text data. Each document in the collection of text data is broken down into individual terms, rows (documents), and columns (terms) identified. A cell value in the matrix represents the frequency of a particular term in a particular document. If we imagine this as our input data set with several hundred thousand rows (documents) and several thousand individual columns (unique terms), this is a huge data set by itself. I use the term-by-document matrix (terms-columns, documents-rows) as a reference for my proposed methods of sampling text data.

Throughout this paper, you will find references to SAS® Contextual Analysis and SAS® Text Miner. SAS Contextual Analysis is primarily built for rule-based modeling while SAS Text Miner is meant for modeling text data using statistical algorithms. I use SAS Text Miner to generate and describe term-by-document matrix. Last of the four methods described in the following sections uses SAS Contextual Analysis for testing and validating rule-based models using samples generated from the term-by-document matrix.

EXTERNAL VERSUS INTERNAL DATA

Prior to looking into the methods I propose for sampling, let us peek into various types of data and the typical sources of it. Some of the prominent ones are:

- Customer satisfaction surveys (internal)
- Customer complaints (internal or external)
- Manufacturer warranty claims (internal)
- Call center logs—Service requests (internal)
- Technician notes (internal)
- Legal documents (internal)
- Chat conversations (internal)
- Tweets from Twitter (external)
- Facebook posts and conversations (external)
- User discussion forums (external)
- Customer product reviews (external)

If we carefully examine internal versus external data sources, the fundamental difference lies in the quality and the scope of data patterns. Data sourced externally from social networking sites and user discussion forums are particularly notorious for containing conversations or posts that are very elaborate yet at the same time diluted with unrelated topics and discussions, drifting away from the purpose of the original discussion. Adding to these complications, conversation-based data poses a serious challenge for analysts. Tying multiple contexts and conversations with the original post in a discussion thread is an extremely difficult exercise to tackle. For these reasons, it is difficult to squeeze useful information out of such data, filtering out unrelated and useless information. Hence, external data is typically considered to be too broad in scope, with a low signal-to-noise ratio, compared to internal sources. Internal data sources are, however, limited in scope, with discussions or notes restricted to the component, product, service, brand, or the organization involved with the customer. For this reason, analysts can usually make better sense of internal data as compared to external data by collaborating with their organization's subject matter experts.

Additionally, external data sources are typically used in the context of understanding social trends and public preferences for brands, products, or services, in comparison to competition in the market, opinions concerning recent events, and so on. This lends itself to making external data a good choice for sentiment mining. Internal data sources typically are not so useful for gathering opinions or mining sentiment, since data sources like customer complaints, call center logs, and the like are usually about issues or concerns customers have with a product or service. However, they can be used to analyze the root cause of existing issues or trending issues, identifying products or components that lead to warranty claims or expensive repairs, reasons for failure, and so on.

The average size of the data also dictates the size of the term-by-document matrix. In the majority of cases, external data blows up the dimensionality of unique terms that we can find within a similar volume of data in comparison to internal data sources.

SUPERVISED VERSUS UNSUSPERVISED LEARNING

Functionalities in SAS Text Miner such as text clustering and text topic extraction are usually suitable for unsupervised learning needs, while text rule builder and text profiling serves the needs of supervised learning, for which a target variable with a defined class value is available to train the data. While the focus of this paper is on sampling text data, it might not make much difference whether the type of analysis we need to perform is supervised or unsupervised. However, if a target variable is available with two or more levels in the original data, it might be helpful to perform sampling by ensuring that the

generated sample data has a similar proportion of target variable levels. Let us go into the details of the proposed methods to generate better samples on text data.

METHOD 1: USING CLUSTER MEMBERSHIPS AS STRATA

Text mining is an iterative process in which you need to rerun the process either from the text parsing phase through the text filtering phase, or you rerun just the text clustering or text topics algorithm alone in a repetitive fashion. Through these iterations, you have a chance to modify your synonyms lists and stop lists, and you can drop or keep terms as you keep refining your process, based on the use case at hand. Here we are talking about several repetitions to arrive at the point at which you think you have extracted the optimum number of insights from your analysis by suppressing the maximum amount of noise (by filtering unwanted terms) and condensing the dimensionality of the term-by-document matrix (by adding synonyms and performing lemmatization). In this way, you reduce the number of terms in the final term-by-document matrix before you execute additional computationally intensive algorithms to derive a Singular Value Decomposition matrix that is required for generating clusters. This whole process can take several days, depending on how long it takes for each iteration of the process to execute. Each iteration can take from a few to many hours.

For this method, I propose that you perform text mining once on the entire available population data, as long as it doesn't break the CPU and the process continues to run for a few hours and does not encounter any out-of-memory errors. The key take-away here is that if the data volume and the size of the term-by-document matrix doesn't halt your process to perform text clustering in SAS® Text Miner, then do it once. Once cluster memberships are identified within the population data, even after several hours of executing the text-mining process, use the cluster membership as the strata variable to perform stratified random sampling. Documents within the generated clusters are most likely to have similarities in terms distribution and therefore stratification using cluster membership is appropriate. The derived sample data should be a better representative sample of the population data so that you can work on the sample data for all subsequent iterations. This methodology is ideal for internal data sources in which the average size of text documents is not typically large as compared to external data sources and volumes are in the few hundred thousand rather than in millions or billions.

STEPS FOR REDUCING THE TERM-BY-DOCUMENT MATRIX

Dropping a vast majority of terms from the term-by-document matrix can reduce the burden on text filtering and text clustering/text topic extraction to a large extent. These are some of the steps you can perform to reduce terms in the term-by-document matrix.

- When you perform text parsing, you can drop terms that are particular parts of speech (POS) that might not be helpful for your analysis, and which constitute a significant portion of terms generated in the term-by-document matrix. In my experience working on several use cases and data sources, I think that the parts of speech identified by SAS Text Miner listed in Table 1 seldom add any value:

(Abbr) Abbreviation	(Pron) Pronoun
(Aux) Auxiliary or Modal	(Part) infinitive maker, negative participle, or possessive marker
(Conj) Conjunction	(Prop) Proper noun
(Interj) Interjections	(Num) Number or numeric expression
(Det) Determiner	(Prep) Preposition
Adv (Adverb)	VerbAdj (Verb Adjective)
Adj (Adjective)	

Table 1: Parts of Speech Elements to Be Ignored

- Terms belonging to the following attribute type are typically not useful.
  - (Abbr) abbreviated terms

- (Num) term characters include a number
- (Punct) punctuation character

- You can also consider using the option to detect standard entities in text parsing, and then drop the majority of terms identified under any of the listed standard entities in Table 2. Except for the PRODUCT and VEHICLE standard entities, most of the other entities are not useful if your objective is to identify issues or concerns trending in your data.

ADDRESS (postal address or number and street name)	PERCENT (percentage or percentage expression)
COMPANY (company name)	PERSON (person's name)
CURRENCY (currency or currency expression)	PHONE (phone number)
DATE (date, day, month, or year)	**PRODUCT**
INTERNET (e-mail address or URL)	PROP_MISC (proper noun with an ambiguous classification)
LOCATION (city, country, state, geographical place or region, or political place or region)	SSN (Social Security number)
MEASURE (measurement or measurement expression)	TIME (time or time expression)
ORGANIZATION (government, legal, or service agency)	TIME_PERIOD (measure of time expressions)
TITLE (person's title or position)	**VEHICLE** (motor vehicle including color, year, make, and model)

Table 2. List of Standard Entities Detected in SAS Text Miner

METHOD 2: USING ADDITIONAL INPUTS AS STRATA

Textual data originating from external sources such as social networking sites (Twitter, Facebook, and so on) and user discussion forums is difficult to sample because of high data volumes, broader scope of unique terms distribution, and less signal-to-noise ratio. Take a scenario where you would like use Twitter feeds to analyze trends and user sentiment about a product recently released in the market. When you pull data from social media feeds such as Twitter tweets or Facebook posts, you might also be able to get your hands on some additional information about the author of the tweets. For example, you can grab account information from Twitter such as Age, Gender, Location, Country, Twitter handle or Profile ID, Date and Time, Verified/Non-Verified, and so on. Additionally, certain internal data such as customer satisfaction surveys, warranty claims, and more might also contain information about the customer.

In this method, I suggest that you use this demographic, geographical, and time horizon information, along with any other standard information that is available as strata variables to generate your samples, as shown in Table 3. In this way, you are potentially gathering data from quarters of population that you would not typically have access to, which can provide a truer perception of the population. Additional measures to remove duplicate tweets or posts or from a sampling within a twitter handle or a Facebook profile ID might be necessary to avoid bias when a person tweets or posts repetitively on same topic or by using the same hashtag multiple times. If you are doing sentiment analysis on product reviews and a person tweets several hundred times within a week or a month on the same topic, unnecessary bias can be introduced into your analysis results.

Age Group	Gender	Location	Country	Month-Year	Hashtag	Total
18-24	M	Cary, NC	USA	June 2016	xyz	480
18-24	F	Detroit, MI	USA	July 2016	abc	322

Age Group	Gender	Location	Country	Month-Year	Hashtag	Total
25-38	M	Mumbai	India	April 2016	xyz	232
65 and above	F	Toronto	Canada	March 2016	xyz	120

Table 3. Example of Data Structure and Elements for External Data (Twitter)

METHOD 3: USING SME INPUTS TO DEVELOP STRATA

This method is particularly relevant and most suited to situations in which you are looking at internal data such as warranty claims, technician notes, call center logs, and similar sources. Usually, there are subject matter experts (SMEs) available who can provide some inputs based on their historical experience in analyzing data either manually or using other tools. This experience comes in handy when you think that the objectives for the analysis involve understanding trends, themes, or issues associated with certain products, components, features, or other areas of interest. In such cases, SMEs might be able to provide a thesaurus of terms that best describe what you are after. If you think of these terms as one segment, then the remaining unexplored (or unknown) set of terms in the corpus form another segment in the term-by-document matrix. The second segment of terms might have some useful information that can show strong associations with terms in first segment.

Filter unnecessary bunch of terms from the term-by-document matrix by leaving out the parts of speech terms that you deem useless, the terms added to a stop list, and the terms lost due to lemmatization and synonyms. Now, create new columns, which indicate the percentage of the total number of terms that were recommended by SMEs and are populated with nonzero values. For example, if 20 out of 1000 terms recommended by SMEs are populated with nonzero values for document D1, then enter 2 in the cell for D1 in the %SME column. Similarly, if 500 out of 10,000 unexplored set of terms (or unknown terms) are populated with nonzero values for document D1, then enter 5 in the cell for D1 in the %UNK column. The %TOT column contains the sum of %SME and %UNK columns.

- Documents in the Corpus: D_1 through D_m
- Terms selected from SMEs inputs: T_1 through T_p
- Unknown set of terms: T_{p+1} through T_n

After analyzing the distribution of these derived columns (shown in Table 4), you might be able to create some bins in which to group documents within a specific range of values. Thus, you will end up with two strata variables based on the %SME and %UNK columns and one from %TOT. You should be able to use these strata variables, either %SME and %UNK or just %TOT, to perform a stratified sampling of documents from this enhanced term-by-document matrix to generate a reasonable mixture of documents that contain terms you identified based on the inputs provided by the SMEs and terms that are unknown to you prior to the exploration. This kind of sample might add value to your analysis such as issue detection, root cause analysis, trend analysis, and so on.

You might be able to extend this type of enrichment to include the percentage of cells populated with nonzero values based on terms grouped by POS tags (Nouns, Verbs etc.) within the higher-level segments of terms suggested by the SMEs and the unknown set of terms. For example, take the use case of analyzing a large volume of manufacturer warranty claims. Your objective might be to group claims that have a common failure mode or failure part. Failure modes are usually terms that are verbs that describe the type of failure. Failure parts are nouns or noun groups that describe the part that has failed. Using this method for sampling data might be appropriate in this case.

Additionally, you can calculate the percentage of documents that contain terms T_1 through T_n as T_1% through T_n% (shown in the bottom row in Table 4) so that when you perform the sampling, you can verify whether the percentage values are similar in the resultant sampled table as compared to the population data for a majority portion of total terms in the document collection. Such metrics can be useful to reasonably approximate population dataset to a smaller sample but it is probably difficult to achieve without going through several iterations.

	Terms Selected from SMEs Inputs						Unknown Set of Terms					
	T_1	T_2	T_3	T_4	…	T_p	….	T_{n-1}	T_n	%SME	%UNK	%TOT
D_1										2	16	18
D_2										8	28	36
D_3										13	12	25
D_4										7	2	9
…												
D_m										4	8	12
	$\%T_1$	$\%T_2$	$\%T_3$	$\%T_4$	…	$\%T_p$	….	$\%T_{n-1}$	$\%T_n$			

Table 4. Sample Structure of an Enhanced Term-by-Document Matrix with SME Inputs

METHOD 4: CATEGORY-SPECIFIC SAMPLING FOR RULE-BASED MODELS

In SAS Contextual Analysis, you can develop linguistic rule-based categorization models for document classification using Boolean operators, qualifiers, regular expressions, and more. Usually, taxonomies developed primarily based on rule-based model development can have anywhere from a few categories to several hundred categories and sub-categories. This is very typical in use cases for which you are required to classify journals, news articles, product parts warranty claims, product descriptions for SKUs etc. The challenge here, though, is to generate a sample data set that can be used to test all these categories in the model taxonomy. If we generate samples from a large population data set, representative samples that can be used to test and validate category rules for all categories in the taxonomy might not be captured. Specifically, this can occur when you are required to test categories designed to capture a low-frequency of occurrence in the documents.

Let us consider a scenario for building a taxonomy to capture all types of federal government regulations based on analysis of customer complaints. Typically, categories are designed and developed in a hierarchical fashion such that they capture documents that fall under certain types of violations. Let us assume that there is a category that usually has 2–3 cases a day and that doesn't exceed 20–30 cases in a month. If your population data volume for a month is a few hundred thousand or a few million complaints, then how can you efficiently generate a sample that definitely contains some documents from this sort of a rarely occurring event?

In this method, I aim to solve that problem by proposing to have several data samples to test various categories in a taxonomy rather than just one sample data set. It can be tedious to maintain individual samples for individual categories, but it will lead to better accuracy of models. Table 5 shows an example of how you can enhance an existing term-by-document matrix table by deriving additional columns that indicate the composition of terms that make up the Boolean rules developed for a category as per the following guidelines:

- If all the terms that make up a category rule have nonzero cell values for a particular document, then flag that document as "F" for that category.
- If only a subset of terms that make up a category rule have nonzero cell values for a particular document, then flag that document as "P" for that category.
- If none of the terms that make up a category rule have nonzero cell values for a particular document, then flag that document as "N" for that category.

	Terms chosen from Category Rules						Unknown set of Terms					
	T_1	T_2	T_3	T_4	…	T_p	….	T_{n-1}	T_n	Cat1	Cat2	Cat_x
D_1										F	F	N
D_2										P	F	N
D_3										P	N	N
D_4										N	P	F
…												
D_m										F	N	P

Table 5. Sample Structure of an Enhanced Term-by-Document Matrix for Rule-Based Models

- Once you derive such additional category-specific columns, you might consider using those individual columns as stratification variables and generate samples specific to each category. For example, you can generate a sample data set to test Category1 in your rule-based model using Cat1 as stratification variable.
- Instead of loading each and every individual data set to the contextual analysis solution in order to test a category, you should be able to use scoring code to execute the model on sampled data and generate a resulting table with the predicted value (Match/No Match) for that category.
- Carefully analyze the output and add another column to record what should have been the actual output (Match/No Match) for that category, for that document. In this way, you can create a gold-standard data set in no time that shows you how many of the documents in the sample data should match the category.
- After testing and validating a category on its corresponding generated sample data set over a few iterations, you should be able to come up with test results and an output table that shows the distribution of the matches for that category on the sample data set, as shown in Table 6.
- If your category rules are perfect, there shouldn't be any matches for documents flagged with a sampling indicator P or N. After you investigate those particular cases, describe the issue in the Reason(s) column. Maintaining this type of a log for sampled data on each category under development will yield better results over few iterations of fine tuning category rules.
- For documents flagged with the sampling indicator F, there are two possible outcomes and two actual outcomes. The underlying category rule might have been incorrectly developed, leaving out potential matches (FN), or the rule might have been liberally built, causing unwanted matches (FP).

Sample Flag	Actual	Predicted	Result	% of Docs	Reason(s)	Suggested Action
F	Match	Match	TP	65%	Perfect Matches	-NA-
F	Match	No Match	FN	2%	Terms in these documents out of range for DIST_5 rule	Change rule from DIST_5 to DIST_6 and test
F	No Match	Match	FP	14%	Matching documents with negation	Add negation rule and test
F	No Match	No Match	TN	4%	Perfect Non-Matches	-NA-
P	Match	No Match	FN	1%	Missing synonyms for certain terms in the rule	Add synonyms to the rule using an OR operator
P	Match	Match	TP	1%	Perfect Matches	-NA-
P	No Match	No Match	TN	1%	Perfect Non-Matches	-NA-
P	No Match	Match	FP	4%	Regex concept referred in the rule is catching wrong terms	Modify the Regex concept rule and test
N	Match	No Match	FN	1%	Terms or Boolean logic is not enough to capture these patterns	Modify rule to accommodate required changes
N	No Match	No Match	TN	5%	Perfect Non-Matches	-NA-
N	No Match	Match	FP	2%	Added sub-rule under OR root node is causing issue	Remove sub-rule and test

Table 6: Distribution of Testing Results on a Sampled Data Set for a Category

CONCLUSION

Sampling data for unstructured data is hard simply because of the dimensionality problem. In this paper, I show how to choose between several proposed methods, depending on the use case at hand, type of data, source of information (external versus internal), availability of additional information, and the purpose of text data analysis (exploratory or predictive). These are some of the methods that I am prescribing based on the experience I have gained from handling textual data in various customer environments. Research can be further advanced to study how we can generate better representative samples using the term-by-document matrix of a population data set. I suggest that the new methodology should focus on trying to maintain two metrics (the percentage of documents in which term exists, and the average number of occurrences of the term in the documents) similar between population data and generated sample for the majority of terms in the matrix. In this way, the term weights and frequency weights that are generated from population data for key terms that describe and discriminate the documents can also be replicated to a great extent in the sample data, thus validating the requirement of generating a representative sample.

REFERENCES

Chakraborty, G. M. Pagolu, and S. Garla. 2013. *Text Mining and Analysis: Practical Methods, Examples, and Case Studies Using SAS®*. Cary, NC: SAS Institute Inc.

SAS Institute Inc. Text Analytics Using SAS® Text Miner. Course Notes. Course information: https://support.sas.com/edu/schedules.html?ctry=us&id=1224

RECOMMENDED READING

Text Mining and Analysis: Practical Methods, Examples, and Case Studies Using SAS®

CONTACT INFORMATION

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

Murali Pagolu
SAS Institute Inc.
+1 (919) 531 4696
Murali.Pagolu@sas.com