Chapter 1: Introduction to SAS® Enterprise Miner™ and SAS® Text Miner

1.1 Data Mining and Text Mining
1.2 Working with Data Sources
1.3 Using SAS Enterprise Miner and SAS Text Miner

Objectives

– Describe SEMMA data mining methodology.
– Tour the SAS Enterprise Miner user interface.
– Describe SAS Text Miner.
– Explain some of the nodes in SAS Text Miner.
– Use the SAS Text Miner nodes to explore a document collection.
SAS Enterprise Miner: Interface Tour

Menu bar and shortcut buttons

Project panel
SAS Enterprise Miner: Interface Tour

Properties panel

Help panel
SAS Enterprise Miner: Interface Tour

SEMMA tools palette

Sample
Explore
Modify
Model
Assess
Using SAS Text Miner

- In this class, you also use the High Performance (HP) Text Miner node.

Using the Text Parsing Node

- Table showing properties of the Text Parsing node with values.
The Text Parsing Node

- The Text Parsing node
  - builds the corpus dictionary
  - associates terms with parts of speech and controls which parts of speech to recognize
  - performs stemming to equate terms that are different verb tenses of the same verb, or to equate terms that are either singular or plural versions of the same noun
  - identifies up to 16 entities such as address, company name, currency, and person’s name
  - imports custom entities created by a product such as SAS Concept Creation or SAS Content Categorization Studio
  - controls recognition of numbers or punctuation as separate terms.

The Text Parsing Node

- Verb stemming example:
  - type
  - typed
  - typing
  - types

- Noun stemming examples:
  - house, houses
  - matrix, matrices
  - criteria, criterion
The Text Parsing Node
• The Text Parsing node special tables:
  – Synonyms
  – Multi-word term dictionary
  – Start/stop list - table of terms to include or exclude from the analysis

The Text Parsing Node
• Dictionaries when a stop list is specified:
  – Corpus dictionary: the union of all terms in the corpus (derived, not specified)
  – Stop list: a dictionary of terms to be ignored in the analysis (specified by the user)
  – Start list: terms in the corpus dictionary that are not in the stop list (derived)

• The stop list is typically used to remove low information terms that add only noise to the analysis. Noisy data has no descriptive or predictive value.
The Text Parsing Node

• Dictionaries when a start list is specified:
  – Corpus dictionary: the union of all terms in the corpus (derived, not specified)
  – Start list: a dictionary of terms to be used in the analysis (specified by the user)
  – Stop list: terms in the corpus dictionary that are not in the start list (derived)

• The start list can be a technical or business dictionary that is developed by the analyst or obtained from other sources.

Using the Text Filter Node
The Text Filter Node

• Text Filter Properties
  – Frequency weights and term weights are included.
  – The optional Check Spelling property uses a spelling dictionary and word-similarity algorithms to find and correct misspellings.
  – The Minimum Number of Documents property performs frequency filtering for rare terms. This property can be used rather than searching for rare terms and adding them to the stop list.
  – The Filter Viewer enables you to interactively control terms to drop or keep, interactively create synonyms, and perform queries and view concept links.

The Text Filter Node

• Analysis Features
  – Frequency weights
    • Log (default)
    • Binary
    • None (count or frequency)
  – Term weights
    • Entropy (default)
    • Inverse Document Frequency
    • Mutual Information
    • None
The Text Filter Node

• Query filters have the following characteristics:
  – can be used in the Properties panel and in the Interactive Filter Viewer
  – return documents satisfying the query
  – can be used to subset the document collection for the continuing downstream analysis of the collection

The Text Filter Node

• Query Operators
  – +term returns all documents having at least one occurrence of term.
  – -term returns all documents having zero occurrences of term.
  – “text string” returns all documents having at least one occurrence of the quoted text string.
  – string1*string2 returns all documents that have a term that begins with string1, ends with string2, and has text in between.
  – >#term returns all documents that have term or any of the synonyms that are associated with term.
Using the Text Cluster Node

The Text Cluster node separates the entire corpus of documents into mutually exclusive clusters.

– Each document belongs to one and only one cluster, and the user has control over the number of generated clusters.

– For interpretation, key descriptive terms from the documents are automatically displayed for each cluster.

– The descriptive terms help the analyst understand the types of documents that are being put in a cluster.

The Text Cluster Node
The Text Cluster Node

– For example, suppose the corpus of documents is a collection of newspaper articles – some of them about sports and others about politics. Then you might expect to see the following:

  • one cluster of documents with key descriptive terms such as *baseball*, *soccer*, *score*, and so on
  • another cluster of documents with key descriptive terms such as *election*, *campaign*, *votes*, and so on
– The Text Cluster node is run after
  • the Text Parsing node performs its natural language processing and “tokenized” the terms
  • the Text Filter node selects the terms to work with and applies certain weights.

The Text Cluster Node

– Clustering is performed by using a linear algebra approach to the term-document frequency matrix.
– As an example of the raw input data that ultimately is processed to produce clusters, the table below shows that “cat” occurred three times in Document 1
  and “dog” occurred two times in Document 2.

<table>
<thead>
<tr>
<th></th>
<th>Doc 1</th>
<th>Doc 2</th>
<th>Doc 3</th>
<th>…</th>
<th>Doc N</th>
</tr>
</thead>
<tbody>
<tr>
<td>apple</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>cat</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>dog</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>farm</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>White House</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Senate</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

– Basically, documents that have similar term usage tend to be put in the same cluster. In this case, Document 2 and Document 3 look somewhat alike.
The Text Cluster Node

– The user also has control of the cluster derivation.
  • Choose the exact or maximum number.
  • Choose the maximum number of clusters.
  • Choose the cluster algorithm that is used.
    – Expectation-Maximization (EM)
    – Hierarchical

– Clustering documents is a powerful analytic approach, but can you see a possible shortcoming of this idea?
  • To use the previous example, what happens to a newspaper article that deals with both sports and politics?
  • It can be placed only in one cluster or another, but not both. That is why there is also a Text Topics node.

Using the Text Topic Node
The Text Topics Node

– A topic is a subject or theme or idea that occurs in a document.

– For example, suppose the document is a newspaper article containing topics related to the following:
  • sports
  • politics
  • law

– Clearly, a document can contain more than one topic (whereas a document can belong only to one cluster).

– Topics are generated
  • automatically by the Text Topic node using basically the same underlying mathematical algorithm that Text Cluster uses (modified a bit)
  • also by user definitions.

The Text Topics Node

– The basic idea behind automatic topic generation is to find terms that occur frequently together within documents. Together these germs “define” the topic.

– This approach can be looked at by thinking of terms as potential “friends” in a social network.
  • The terms car and auto might co-occur (be “friends”) within many of the same documents.
  • However, even if they are not direct “friends” in the same documents, they can still be “friends of friends.”
The Text Topics Node

As an example of this indirect connection, suppose *car* and *auto* never co-occur in the same document, but each co-occurs frequently with *tire*. Therefore, *car* and *auto* are friends of the same friend and might be recognized by the Text Topic algorithm as key descriptive terms for the same topic.

– By default, 25 topics are automatically generated.
– These topics are identified (or interpreted) by the analyst most frequently by examining a list of five key descriptive terms that are automatically displayed.

The Text Topic Node

- **Text Topic Properties**
  - A custom topic table can be supplied by the user.
    The table can be imported, or the table can be manually created with a table editor.
  - The user can request up to 1,000 single-term topics to be derived. By default, no single-term topics are derived.
  - A user can specify up to 1,000 multi-term topics to be derived. By default, 25 multi-term topics are derived.
The Text Topic Node

• Topics
  – Single-term topics are not the same as filtering on a single term. For example, a topic can be derived based on the single-term *price*, but documents might be labeled as not having the topic even if the term *price* is present in the document.
  – The node might return fewer topics than requested. After the designated number of topics are derived, the node can decide that topics 24 and 25 are not sufficiently distinct to warrant including both topics, for example. If so, topic 25 (based on order of importance) is dropped.

The Text Topic Node

• Custom Topics
  – A custom (user-defined) topic consists of a label and one or more terms. Each term has a role and a weight.
  – The weight associated with a term-role pair indicates the analyst’s judgment about the relative importance of the term-role pair to the topic.
  – In practice, most users define weights in the range of 0 to 1, where 1 is the highest importance and a weight of 0 is the lowest.
Using the Text Topic Node

<table>
<thead>
<tr>
<th>Topic</th>
<th>Term</th>
<th>Role</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>analytics</td>
<td>analytics</td>
<td>noun</td>
<td>1.0</td>
</tr>
<tr>
<td>analytics</td>
<td>analyze</td>
<td>verb</td>
<td>0.9</td>
</tr>
<tr>
<td>analytics</td>
<td>logistic regression</td>
<td>NOUN_GROUP</td>
<td>0.5</td>
</tr>
<tr>
<td>data</td>
<td>data</td>
<td>noun</td>
<td>1.0</td>
</tr>
<tr>
<td>data</td>
<td>data warehouse</td>
<td>NOUN_GROUP</td>
<td>0.7</td>
</tr>
<tr>
<td>data</td>
<td>analyze</td>
<td>verb</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Columns in the custom topic table have names: _topic_, _term_, _role_, and _weight_.

**Text Analytics Illustrated with a Simple Data Set**

This demonstration uses a simple, artificial data set to introduce many features of some of the text mining nodes.
1.01 Multiple Choice Poll

• When you run the Text Cluster node, by default, which of the following variables are given the role of input for a predictive model?
  a. the `TextCluster_SVD` variables
  b. the `TextCluster_cluster_` variable
  c. the `TextCluster_prob` variables
  d. b and c above

1.01 Multiple Choice Poll – Correct Answer

• When you run the Text Cluster node, by default, which of the following variables are given the role of input for a predictive model?
  a. the `TextCluster_SVD` variables
  b. the `TextCluster_cluster_` variable
  c. the `TextCluster_prob` variables
  d. b and c above