Feature Extraction for Massive Data Mining

V. Seshadri[†], Sholom M. Weiss[‡], and Raguram Sasisekharan[†]
[†] AT&T Bell Labs,
Middletown, NJ 07748
[‡] Department of Computer Science, Rutgers University
New Brunswick, New Jersey 08903

Abstract

Techniques for learning from data typically require data to be in *standard form*. Measurements must be encoded in a numerical format such as binary true-or-false features, numerical features, or possibly numerical codes. In addition, for classification, a clear goal for learning must be specified. While some databases may readily be arranged in standard form, many others may be combinations of numerical fields or text, with thousands of possibilities for each data field, and multiple instances of the same field specification. A significant portion of the effort in real-world data mining applications involves defining, identifying and encoding the data into suitable features. In this paper, we describe an automatic feature extraction procedure, adapted from modern text categorization techniques, that maps very large databases into manageable datasets in standard form. We describe a commercial application of this procedure to mining a collection of very large databases of home appliance service records for a major international retailer.

Introduction

Machine learning techniques have evolved over time, and range from classical linear statistical models to nonparametric learning models such as decision trees, neural nets, or nearest neighbors(Weiss & Kulikowski 1991). In terms of underlying models, these methods differ greatly. The application of these methods can vary dramatically, too. There may be large differences in performance, both in terms of quality of results and in terms of the time needed to extract the best answers. Performance is typically of paramount concern and often the defining criterion in reporting results. Still, there is another major issue, data preparation, that, while often considered mundane, often occupies the largest amount of effort spent in mining a database (Hirsh & Noordewier 1994).

Most machine learning techniques require that data be in *standard form*. The measurements or features must be encoded in a numerical format such as binary true-or-false features, numerical features, or possibly numerical codes. In addition, for classification, a clear learning goal must be specified. While some databases may readily be arranged in standard form, many others may be combinations of numerical fields or text, with thousands of possibilities for each data field, and multiple instances of the same field specification. In this paper, we demonstrate an automatic feature extraction procedure, adapted from modern text categorization techniques(Apté, Damerau, & Weiss 1994, Lewis 1992), that maps very large databases into manageable datasets in standard form. We describe a commercial application of this procedure to mining a very large database of washing machine and dryer repair records.

Standard Form

While machine learning methods may differ greatly, they share a common perspective. Their view of the world is samples or cases organized in a spreadsheet format. Table 1 illustrates this general organization, where a row C_i is the i-th case, and column entries are values of measurements, $V_{i,j}$, for the j-th feature f_j. For example, samples of medical patient record could be organized in this format. If Row 1 is the case of John Smith, and column 1 is the feature blood pressure, then the entry for the intersection of row 1 and column 1, $V_{1,1}$, is a measurement of John Smith's blood pressure.

The spreadsheet format becomes a *standard form* when the features are restricted to certain types. Individual measurements for cases must conform to the specified feature type. The standard feature types are

Case	f_1	...	f_k
C_1	$V_{1,1}$	...	$V_{1,k}$
	...	...	
C_i	$V_{i,1}$	...	$V_{i,k}$
	...	...	
C_n	$V_{n,1}$	...	$V_{n,k}$

Table 1: Spreadsheet Data Format

the following:

- binary true-or-false variables

- ordered variables. These are numerical measurements where the order is important and X > Y has meaning. A variable could be a naturally occurring measurement such as age, or it could be an artificial measurement such as a severity index for grading illness.

Although some techniques can also handle unordered (categorical) variables or artificial codes, most approaches will translate these into many binary variables. All of the standard form variables are measured as numerical values.

While databases are sometimes found in spreadsheet format, or can readily be converted into this format, they often may not be easily mapped into standard form. Some of the difficulties that can be encountered with mapping data into standard form data fields are:

- High dimensionality. The number of possibilities for an unordered feature may be very large. For example, the number of models and replacement parts for products such as washing machines may number in the thousands.

- Free text. The data field may allow for free text that cannot be anticipated in advance. For example, in a washing machine repair context, the complaint of the customer may be recorded.

- Replication of fields. It may be necessary to specify multiple instances of the same feature. The order of the features is not meaningful. For example, a data field might be a part that was replaced. Multiple parts may be necessary in order to successfully complete a repair. A database may allow for this type of situation by allocating several data fields for specification of a part replacement.

These three issues of dimensionality, free text, and field replication, strongly support the idea of mapping data into a standard form. However, the mapping into features and measurements is more than a straightforward mechanical process. The transformation represents a general conceptual model that is applicable across a broad group of learning methods.

Computer-based learning methods can be divided into roughly 3 categories:

- weighted methods, such as linear discriminants and neural nets;

- symbolic methods, such as decision trees or rules;

- case-based, i.e nearest neighbor methods.

For ordered numerical variables, all methods can readily reason with these variables (possibly with minor normalizations). At one time some symbolic approaches, notably decision tree methods, did treat ordered numerical variables like categorical variables. Most empirical evidence supports the conclusion that (multiple) binary decisions, using 'greater than' and 'less than' operators, is a more effective approach.

The main difficulty is with unordered numerical variables, i.e. codes. Because a specific code is arbitrary, it is not suitable for many methods. For example, a weighted method cannot compute appropriate weights based on a set of arbitrary codes. A case-based method cannot effectively compute similarity based on arbitrary codes. While some symbolic methods may process codes without the transformation into multiple binary codes, they will implicitly make this transformation, for example binary tree induction. An exception is non-binary tree induction, a method which can be mimicked, often more compactly, by binary tree induction. The standard form model presents a data transformation that is uniform and effective across a wide spectrum of learning methods.

Describing the Goal

A dataset can be in standard form, but at least one of the features is given a special designation. Typically, this is a class label. For classification or prediction, we must describe the target. The objective then becomes a mapping from the other variables to the target or class. Often, the designation of a target requires careful consideration prior to data accumulation. For data mining, we might expect to be less certain of targets, and we may be engaged in a wide search to find patterns in the data without having an easily specified set of goals and classes.

Mapping Databases into Standard Form

Text Categorization

Before we directly consider the transformation of a database into standard form, let's examine a different problem known as text categorization. Text, which can be any written document, would appear to be very far from standard form, and possibly the least amenable form of data for data mining. One might suspect that an analysis of the written word would introduce more complicated issues, such as human reading and comprehension.

The task of text categorization is usually performed by humans. They read documents, and assign a label or topic from a pre-specified code sheet. This process categorizes articles by topics of interest, and allows for effective dissemination and future retrieval. For example, the Reuters news service has hundreds of topics, like sports or financial stories, to which newswire articles are assigned.

If we consider the divergence of text data from the standard model, we see that the data is organized by case, and a label is typically assigned to each case by the human reader; but the initial feature set specification is missing. Thus, to map the text data into standard form, two tasks must be achieved: (a) specification of a feature set, and (b) measured values of these features for each case.

Figure 1 gives an overview of the process of mapping a database of text documents into standard form. Research results in text categorization demonstrate that a comprehensive statistical analysis of word frequencies in labeled documents *combined* with relatively strong machine learning techniques can lead to effective automated categorization systems(Apté, Damerau, & Weiss 1994). While there are many variations on the basic methodology, we considered one basic approach. This approach can be improved somewhat for text categorization, but our task of data mining is somewhat different. The most direct approach to the mapping process was found to be quite effective.

In Figure 1, the first step is the creation of a word dictionary. This dictionary contains the feature set. A word is a set of non-numeric characters that are preceded and succeeded by word boundary characters (non-alphabetic characters like blank characters). It is a relatively simple process to find and sort all words that appear in a text document. Of these words, the most frequently occurring *j* words are selected. This set of *j* words contains, as might be expected, some words that are of little predictive value, words such as "the" or "it." A separate *stopword* dictionary is maintained, containing a list of frequently occurring nondiscriminatory words. These words are then removed from the main dictionary, and the remaining *k* frequently occurring words are maintained in the dictionary along with the class label. The dictionary, then, is the set of *k* features for the text categorization task.

After completing the above-mentioned procedure, the set of measurements for each case is still missing. In Figure 1, the next step is indicated, namely, mapping of the text for each case into standard form. This is accomplished by searching for the presence of each dictionary word in the text for each case. If the word is found, the value of the measurement is true; otherwise it is false. At the completion of this step, we now have a standard form data set, one that can be used by many machine learning methods.

General Feature Extraction

Variations of the feature strategy that we have described have proven effective for text categorization. Large datasets, with high dimensional feature spaces, are mapped into a standard form that is effective for machine learning. A very large database may have more structure than pure text. Clearly, those data fields that are in standard form can immediately be used. Those data fields that are known to be irrelevant can be eliminated. The issue to be addressed is what to do with the fields that are not in standard form, particularly those with high dimensions. In this paper, we describe the use of text categorization techniques with the following modifications for feature extraction prior to general data mining:

- A word consists of alphanumeric characters
- Every word in the dictionary is a valid candidate for designation as a class label.

Alphanumeric characters are now allowed because many coded fields may contain them, for example, descriptions of parts of a product. Because the objective is data mining and there are no prior designated class labels, experiments are performed to cycle through all binary features. This process determines whether any patterns emerge with some predictive capability. While patterns may emerge, domain experts determine whether these patterns are noteworthy.

Experimental Results and Discussion

A large database, with data fields in non-standard form, was compiled from several databases of maintenance and repair records for an international merchandiser. The initial analysis was limited to a portion of the records, namely washing machine and dryer records over several months, with a view to test whether the above-mentioned approach would yield results of commercial value; if so, the goal was to extend the approach to a broader set of appliances.

The collated database contained 10,000 cases. The cases were a few months' records of service calls made

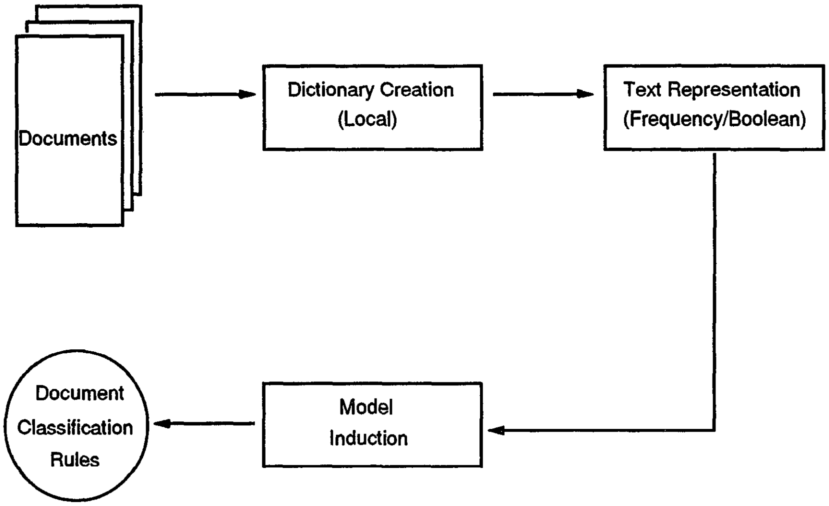

Figure 1: Overview of Text Categorization

for a well-known brand of washing machines and dryers, and covered a large geographical area across North America. Figure 2 illustrates the type of data, and the dimensions of some of the database fields. For example, it is not surprising that in a real-world application of this sort, there are 1100 different types of parts that are replaced. For each service call, up to 10 parts are entered into data fields. If multiple parts are replaced, there is no preferred order for noting a part in any of the 10 fields allocated for the description of replaced parts. Another field contains information about the initial phone call made by the customer to describe the problem, and is free form text up to a limit of 100 characters.

As can be seen by perusing the table, these data are not in standard form. We applied the procedures described earlier to the data, with an upper limit of 200 features. After eliminating stopwords, approximately 150 features remained. Once in standard form, each of these binary features was temporarily considered to be a labeled endpoint. In effect, 150 classification and prediction experiments were performed. For each label, an attempt was made to find an associative pattern. For example, an examination was made to determine whether there was any pattern in predicting that a specific part will need to be replaced, or any pattern associated with a specific model year.

As noted earlier, many different learning methods can be applied to a dataset in standard form. In our case, we applied a learning method that induces decision rules(Weiss & Indurkhya 1993), rules in the form of conjunctions of features. For example: If X AND Y then conclude Z. For the 150 endpoints that were examined, over 500 patterns were discovered.

Not all of these induced relationships were novel in the domain of washing machine products. Many were well-known to workers in the industry, and only novel to outsiders. Others were previously unknown to domain experts, but made sense upon reflection and examination. Both types of relationships were useful; the former type lent credibility to the approach, while the latter type could be used to improve maintenance and repair performance. Domain experts indicated that each 1 per cent improvement in performance would result in 20 million dollars increase in net revenue.

At the time of writing of this paper, domain experts were considering the applicability of this approach to multiple lines of appliances. Even if only one useful relationship is found in a merchandise category, it could have significant financial impact. From a technical perspective, however, the data are not well-formed, and the specific goals are ill-defined. In this paper, our objective has been to show how a difficult technical problem of feature extraction and high dimensionality can be made tractable. The procedures that have been described have proven feasible and efficient in automatically mapping a large database into a productive form for data mining.

Acknowledgments

Sholom Weiss received research support from NIH-NCRR grant RR-6235.

Description	Categories	Instances	Type
Parts	1100	10	alphanumeric code
Model	100	1	alphanumeric code
Model Year	25	1	number
Customer complaint	3000	1	free text

Table 2: Examples of Data Fields

References

Apté, C.; Damerau, F.; and Weiss, S. 1994. Automated Learning of Decison Rules for Text Categorization. *ACM Transactions on Office Information Systems* 12(3).

Hirsh, H., and Noordewier, M. 1994. Using Background Knowledge to Improve Inductive Learning. *IEEE EXPERT* 9(5):3–6.

Lewis, D. 1992. Text Representation for Text Classification. In Jacobs, P., ed., *Text-Based Intelligent Systems.* Lawrence Erlbaum.

Weiss, S., and Indurkhya, N. 1993. Optimized Rule Induction. *IEEE EXPERT* 8(6):61–69.

Weiss, S., and Kulikowski, C. 1991. *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems.* Morgan Kaufmann.