Automated Feature Learning: Mining Unstructured Data for Useful Abstractions

Abhishek Bafna, abafna@umich.edu
EECS, University of Michigan, Ann Arbor

Jenna Wiens, wiensj@umich.edu
CSE, University of Michigan, Ann Arbor

Abstract—When the amount of training data is limited, the successful application of machine learning techniques typically hinges on the ability to identify useful features or useful abstractions. Expert knowledge often plays a crucial role in this feature engineering process. However, manual creation of such abstractions can be labor intensive and expensive. In this paper, we propose a feature learning framework that takes advantage of the vast amount of expert knowledge available in unstructured form on the Web. We explore the use of unsupervised learning techniques and non-Euclidean distance measures to automatically incorporate such expert knowledge when building feature representations. We demonstrate the utility of our proposed approach on the task of learning useful abstractions from a list of over two thousand patient medications. Applied to three clinically relevant patient risk stratification tasks, the classifiers built using the learned abstractions outperform several baselines including one based on a manually curated feature space.

I. INTRODUCTION

Informative features are often instrumental for the successful application of machine learning techniques. When the number of training examples is limited, an appropriate feature space can lead to the identification of important similarities among the data. Domain-specific knowledge often plays a key role in engineering such features or abstractions. For instance, given a long list of medications, an expert in pharmacology is capable of creating a taxonomy for the drugs based on their molecular structure and/or physical and chemical properties. This kind of expert abstraction can encode crucial information necessary for identifying similarities between patients. For example, such abstractions could help identify two patients, one taking penicillin and another taking amoxicillin, as similar.

Here, we focus on feature-learning techniques for automatically incorporating auxiliary expert knowledge. We focus on building abstractions from categorical data since such data are common across a number of different domains including healthcare, social sciences, recommender systems, etc. The often high dimensionality of these categorical data can make it difficult to identify relationships in the data using standard statistical methods [1]. In order to obtain a robust result, the amount of data required can grow exponentially with the number of categories [2]. Moreover, traditional Euclidean-based measures of similarity fail to capture the often complex relationships among categories/features. Thus, expert knowledge is typically required to identify these relationships and engineer useful lower-dimensional feature representations of the data.

Unfortunately, the process of manually designing abstractions using expert knowledge is labor- and time-intensive in addition to being subjective. This has led researchers to devise more automated methods for building abstractions [3]–[6]. However, when the data are categorical (like in the example described above), conventional unsupervised dimensionality reduction techniques (e.g., Principal Component Analysis (PCA), Latent Semantic Indexing (LSI), Locally Linear Embedding (LLE)) can miss important domain-specific relationships and similarities among the features [7]–[9]. We aim to solve this problem through the incorporation of readily available expert knowledge. For this, we turn to the Web.

In this paper, we present an unsupervised feature-learning framework for building useful abstractions for categorical data. Our proposed method automatically incorporates expert knowledge available in an unstructured form from the Web. The framework is composed of two stages:

1) Using unstructured data procured from the Web, we learn a hierarchical Pachinko allocation model (HPAM) [10] to discover a set of latent variables (i.e., supertopics and subtopics that describe relationships among the categories).

2) Using the Earth Mover’s Distance (EMD), we account for the non uniform distances between the latent variables [11], [12]. This enables us to build a relevant similarity matrix that can be used for both supervised and unsupervised tasks.

In the first stage, we employ hierarchical topic models for discovering semantic relations among categories. In doing so, we exploit the innate hierarchical structure often present in categorical data. Applied to data represented by high-dimensional categorical features, this stage results in a lower-dimensional feature representation. In the second stage, we further capture the complex relationships among categories by implicitly incorporating the ground distance (i.e., the underlying distance between individual features) using the EMD when computing similarity between examples.

As a case study, we consider the task of learning useful abstractions from a list of over 2,000 medication names. In healthcare applications, patient medications are typically encoded using an n dimensional feature space, where n is the number of distinct medications [13]–[15]. Since patients are typically on only a few medications at a time, the data are sparse. Identifying similar patients in such a sparse, high-dimensional feature space is challenging. Thus, using the methods described above, we leverage the large amount of pharmacology knowledge available on the Web to learn a more useful feature representation.

We apply this representation to over 25,000 patient ad-
missions and build a non-linear kernel representing patient similarities. We demonstrate the utility of this approach, by considering three different clinical prediction tasks: mortality within 30 days of discharge from the hospital admission, readmission to the ICU 100 days after discharge from the hospital and readmission to the ICU during the same hospital visit. We compare the classification performance of our approach to a number of baselines including a straightforward \( n \) dimensional feature representation with a Euclidean distance based kernel. When the amount of training data is limited, the classifiers that incorporated unstructured Web data outperform classifiers built using conventional feature engineering techniques on all three prediction tasks. Somewhat surprisingly, the classifiers based on the learned abstractions even outperform classifiers based on a manually curated taxonomy.

II. BACKGROUND/RELATED WORK

Typically, in machine learning and data mining applications, a categorical variable with \( n \) categories is represented by an \( n \) dimensional feature vector. When \( n \) is large and each example is associated with only a small number of possible categories, the resulting feature space is both high dimensional and sparse. These issues are compounded by the fact that such a representation ignores any underlying similarities among categories.

In some applications, there exists readily available information regarding the relationships among categories. E.g., in image processing and color perception, the categorical values (colors) can be easily represented by a vector of RGB intensities [16], [17]. This additional information results in a more meaningful comparisons than if we compared the categorical values alone. However, in many other application domains it can be difficult to quantify the relationships among categories, e.g., stand-alone databases with a large number of categorical variables. To this end, unsupervised dimensionality reduction techniques like \textit{k-means} clustering have been extended to include categorical information [18]–[20].

Although these techniques aim to discover abstractions from the data itself, in many cases expert or domain-specific knowledge is essential for building informative low-dimensional representations of categorical data. To this end, ontology and taxonomy development has played an important role in the conceptualization of domain-specific knowledge for the purpose of information retrieval and interoperability between domains. Manually curated ontologies and taxonomies remain laborious, expensive, and challenging to maintain in dynamic domains [21], [22]. This has led to the fast rise of ontology and taxonomy learning systems that automatically or semi-automatically discover knowledge. Some systems rely on structured data like a database schema [3], [23], some, like OntoLearn use semi-structured data inputs like WordNet [4], [24]–[26]. Systems like OntoUSP learn taxonomies from unstructured data and induce a probabilistic ontology using dependency-parsed text as input [6], [27]–[30]. Many of these systems are based on probabilistic and graphical models. In natural language processing it has been shown that probabilistic topic models are a powerful tool for unsupervised taxonomy learning [31]–[33]. In [32], the authors use topic models such as latent semantic analysis (LSA), probabilistic LSA (pLSA) and LDA to induce ontologies from data. To date, much of this work has focused on lexical semantics to enable applications like information retrieval and question answering.

Here, we consider these kinds of learning systems in a different setting, one that is focused on feature engineering. We induce a taxonomy-structure from unstructured data as a means to engineer informative low dimensional features and capture important relationships in categorical data. Our goal is to learn a useful representation of the data that leads to the identification of important similarities. We describe our feature-learning framework in the sections that follow.

III. METHODS

In this section, we present a formal description of the problem statement and introduce notation. We then describe our feature-learning framework. First, we briefly describe the hierarchical topic modeling approach employed. Second, we outline how this model can be used to construct feature vectors in a low-dimensional space. Finally, we show how the learned relationships among categories can be incorporated into a non-Euclidean distance measure that can be used to identify similar examples.

A. Notation and Problem Statement

Consider a dataset of \( N \) data points \( \{x_i | x_i \in \mathcal{X}\}_{i=1}^{N} \), where \( \mathcal{X} \) is an \( M \)-dimensional binary feature space. \( x_i \) represents \( M \) categories i.e. \( x_i = \{x_{i1}, x_{i2}, \ldots, x_{iM}\} \) and \( x_{ij} \in \{0,1\} \). Here, we use boldface notation to denote vectors.

In addition to these categorical data, we have a corpus of \( M \) documents, consisting of unstructured text data, having a vocabulary of size \( V \). We assume that each of the \( M \) documents pertains to a single unique category in \( \mathcal{X} \).

Given these data, we aim to learn a lower-dimensional feature representation, \( \mathbf{x}' \in \mathbb{R}^k \) where \( k < M \), that captures useful relationships among the \( M \) categories.

To achieve this goal, we first apply topic modeling techniques to the corpus of documents. Each document/category is modeled as a distribution over topics. By exploiting the overlap in these distributions we construct a lower-dimensional representation of the data.

B. Hierarchical Pachinko Allocation Model

Given the corpus of unstructured text data, where each document corresponds to a category we aim to learn a low-dimensional shared representation of these data (i.e., topics). We begin by modeling the unstructured text data using a hierarchical Pachinko allocation model (HPAM), a variant
of the Pachinko allocation model (PAM) [34]. A PAM uses a directed acyclic graph (DAG), where the nodes represent topics, to capture hierarchical relationships among topics. In a PAM, a subset of topics (i.e., supertopics) is associated with a distribution over other topics (i.e., subtopics) and only these subtopics are associated with a distribution over the vocabulary (i.e., word distribution). An HPAM also captures the hierarchical relationships among topics, but in contrast to a PAM, every topic is associated with a word distribution. Here we consider the second variant of HPAM, presented in [10] and illustrated in Figure 2.

For an HPAM having \( R \) supertopics and \( S \) subtopics, the notation is as follows:

- \( \alpha_i \) is the concentration parameter for distribution \( \theta_i \).
- \( \theta_{0m} \) is the supertopics distribution for each document \( m \).
- \( \theta_{0r} \) is the subtopics distribution for each document \( m \).
- \( \phi_z \) is the word distribution for the exit topic.
- \( \phi_r \) is the word distribution for each supertopic \( r \).
- \( \phi_{R+z} \) is the word distribution for each subtopic \( s \).

In this model the Dirichlet distribution of each internal node has one extra ‘exit’ dimension. In our case, the root node has an \( R+1 \) - dimensional distribution over the supertopics and each supertopic has an \( S+1 \) - dimensional distribution over the subtopics. This additional dimension corresponds to the event that a word is sampled directly from the word distribution associated with that topic.

While LDA and its variants do capture correlation patterns in words, they capture none or limited correlations among the topics themselves. But in most of the real-world data, topics are generally not independent of each other. Ignoring these correlations makes an unrealistic assumption and this can lead to the discovery of incoherent topics. Also the categorical data might possess innate hierarchical structure. A PAM captures all correlations between topics using a directed acyclic graph (DAG). However it does not represent a nested hierarchy of topics. A hierarchical pachinko allocation model (HPAM) [10] combines the advantages of a topical hierarchy representation with PAM’s ability to capture topic correlations.

![Fig. 2: A Hierarchical Pachinko Allocation Model (HPAM), through a set of supertopics and subtopics, captures the innate hierarchy present within a corpus of documents. \( \theta_0 \) represents a distribution over supertopics while \( \theta_t \) represents a distribution over subtopics for supertopic \( r \).](image)

### C. Low-Dimensional Representation

Now, we employ the topic model learned in the previous section to construct a low-dimensional representation of the dataset \( \{x_i\}_{i=1}^N \). Given the topic model, for each document (i.e., category) \( m \) in our corpus we have: \( \theta_{0m} \in \mathbb{R}^{R+1} \) and \( \theta_{0m} \in \mathbb{R}^{S+1} \). Let \( \Theta_m = [\theta_{0m}, \theta_{1m}, \ldots, \theta_{Rm}] \). We represent each document by its distribution over supertopics concatenated with a weighted combination of its distributions over subtopics. The feature vector for the document \( m \), \( z_m \) is defined as:

\[
  z_m = [\theta_{0m}; \Theta_T m \theta_{0m}]
\]

For the data point \( i \), we have a high-dimensional feature vector \( x_i \in \{0,1\}^M \). Let \( |x_i| \) represent the number of non-zero features. Then we can derive a lower-dimensional representation of \( x_i \), \( x'_i \in \mathbb{R}^k \) where \( k = R + S \) (the combined number of supertopics and subtopics) as:

\[
  x'_i = \frac{1}{|x_i|} \sum_{m=1}^{M} \mathbb{I}(x_i=1) z_m
\]

where \( \mathbb{I}(x) \) is an indicator function that evaluates to 1 if \( x \) is true or to 0 otherwise. Note that we do not include the “exit” topics in our final feature vectors since we do not expect these topics to be informative.

### D. Measuring Similarity

Given the low-dimensional feature representation of the data obtained in the previous section, we can begin to identify meaningful similarities among examples. We could quantify similarity based on the Euclidean distance in the low-dimensional space. However, this would implicitly assume that the data lie in an orthogonal feature space, and that each feature is equally important. This assumption does not hold in our application, since some features/topics are closer to each other than to others. Thus we consider a similarity measure based on a cross-bin comparison that takes into account the distance between topics represented by the word distributions. We use the Earth Mover’s Distance (EMD) to efficiently incorporate this distance.

The EMD is based on a solution to the well-known transportation problem [35]. It gives a minimal cost transformation from one distribution to another, where cost is determined by the flow between the bins of respective distributions, and the ground distance between them. In our case, the ground distance matrix is determined by computing all pairwise distances between the supertopics and subtopics. These topics are themselves defined as a distribution over the vocabulary \( w \), and thus we use a symmetric version of the Kullback-Leibler (KL) divergence [36] to compute the distance between two topics \( z_i \) and \( z_j \) represented by their word distributions \( \phi_{z_i} \) and \( \phi_{z_j} \), respectively, as in the following equation.

\[
  D_{KL}(z_j, z_i) = \sum_w \phi_{z_j}(w) \log \frac{\phi_{z_j}(w)}{\phi_{z_i}(w)} + \phi_{z_i}(w) \log \frac{\phi_{z_i}(w)}{\phi_{z_j}(w)}
\]

Given this ground distance, we compute the EMD between each pair of examples \( EMD(x_i, x_j) \) as in [11]. We can then transform this distance matrix into a similarity matrix by replacing the squared-Euclidean distance with the EMD as in:

\[
  e^{\exp(-\frac{EMD(x_i, x_j)}{2\sigma^2})}
\]
where $x \in \mathbb{R}^k$. The result is $K$, a similarity matrix based on a low-dimensional feature representation that incorporates the ground distance between features. This similarity matrix can be used in a number of learning applications, including both supervised and unsupervised tasks. However, before employing $K$ as a kernel in kernelized-methods, it is important to note that it is not guaranteed to be positive semi-definite (PSD). Here, we use symmetric KL divergence as the ground distance function, which is not a metric. Thus, as defined above the EMD-kernel is not guaranteed to be PSD [37]. Still, empirically, such kernels have been shown to often work well in practice [38], [39]. In the next section, we describe a series of experiments designed to measure the utility of the resulting similarity matrix and our proposed feature-learning framework.

IV. Case Study

To test the utility of our feature-learning framework we consider a healthcare application in which we focus on representing patients in the intensive care unit (ICU) in terms of their prescribed medications. We compare the abstractions we learn using our approach to a number of other feature representations. We begin by describing our dataset and the specific classification tasks under consideration.

A. Dataset and Classification Tasks

We use the MIMIC II Clinical Database [40]. This publicly available database contains clinical data from approximately 32,000 ICU patient admissions. For the purpose of this study, we focus on representing each patient admission by the pharmacy orders (i.e., ordered medications). Using these data we aim to build classifiers for predicting patient risk for the following adverse outcomes:

1) mortality within 30 days of discharge from the hospital,
2) readmission to the ICU within 100 days of discharge from the hospital, and
3) readmission to the ICU during the same hospital admission.

For the first task, we consider all admissions in which the patient is discharged alive from the hospital. We make our predictions at the time of discharge and consider all medications prescribed during that admission. Patients can have multiple hospital admissions (see Figure 3). We omit admissions for which there were no recorded medications from our analysis. Here, we focus on predictions based on the first hospital admission; this results in a single prediction per patient. For the second task, we consider the same set of admissions as the first task. This results in a single prediction for each admission. However, we make multiple predictions per patient. Therefore, the number of total examples is higher than in the first task (see Table I). For the third task, we consider all ICU visits in which a patient was discharged alive. Since patients can have multiple ICU visits per hospital admission, we have the greatest number of examples for this task. Similar to the tasks above, when predicting readmission to the ICU we consider only those medications ordered during the most recent ICU visit. As in the previous two tasks, we omit ICU visits for which there were no recorded medications.

All three tasks have high class imbalance. This is typical for many patient risk stratification tasks in healthcare. Table I gives the number of examples for each task and the total number of positive cases (i.e., adverse outcomes).

<table>
<thead>
<tr>
<th>Outcome</th>
<th>Total # Examples</th>
<th># Positive Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>30-day Mortality</td>
<td>22,949</td>
<td>2,828</td>
</tr>
<tr>
<td>Hosp. Readmission</td>
<td>23,292</td>
<td>2,082</td>
</tr>
<tr>
<td>ICU Readmission</td>
<td>23,998</td>
<td>5,587</td>
</tr>
</tbody>
</table>

TABLE I: We consider three clinically relevant classification tasks: 30-day mortality, 100-day hospital readmission, and readmission to the ICU within the same hospital admission. All three tasks have a high degree of class-imbalance.

In addition to these outcomes we extract information regarding the medication orders for each example. In the MIMIC II database physician order entries for medications are represented by a free-text entry and a timestamp. Based on the free-text entries for the entire population, we started with a list of 2,285 unique drug names. These prescriptions correspond to not only to the type of medication but often also the dose and route. E.g., 'nitroglycerin ointment', 'dorzolamide ophth soln'. After we removed erroneous entries, 2,164 unique drug names remained. Therefore, in our application, each patient may be represented as a feature vector with 2,164 features (or categories). In the next section we apply our feature-learning framework to learn a lower-dimensional representation.

B. Model Construction

1) Web Data Mining: In order to learn a meaningful low-dimensional representation of our data, we required expertise in pharmacology. For this we turned to the Web, using Google and Wikipedia as our auxiliary source of “expert” knowledge. We queried each of the 2,164 drugs using Google and downloaded the corresponding top Wikipedia article. (We also derived similar results from specialized web-sites like Rxlist.com but the results are not mentioned here). This resulted in a corpus of 1,124 unique Wikipedia articles, since many of the drugs mapped to the same URL.

Next, we extracted plain text from each Wikipedia page in our corpus. We applied standard preprocessing techniques (e.g., filtered stop words and words with low frequency). In addition, we sorted the dictionary of words based on a tf-idf transformation and removed the top 2% of the words (e.g., 'years', 'possible'). These words occurred repeatedly throughout the corpus and offered little or no information while constructing topics. Our final vocabulary consisted of 21,785 words.
Fig. 4: (a) We plot the distribution over learned topics for a subset of drugs. We observe that important relationships among medications are successfully captured by the model. For example, a number of related antimicrobials (e.g., ampicillin, ticarcillin, etc.) all share the same topic. (b) We use “Miconazole”, a topical medication, as an example. The most probable words associated with miconazole’s most probable supertopic are given, along with the two most probable subtopics. The probabilities of each topic are shown in parentheses.

2) Learning the HPAM: From the corpus of Wikipedia data, we built an HPAM with 30 supertopics and 30 subtopics. We learned the super/subtopic distributions and word distributions using the Gibbs EM algorithm and adopted the implementation of Gregor Heinrich [41], [42]. We determined the number of topics, \( R + S = 60 \), using a hierarchical Dirichlet process (HDP) and analyzing the concentration vector of the base Dirichlet distribution, \( \alpha_0 \) [43]. Keeping the total number of topics constant at 60, we selected the ratio of supertopics to subtopics using five-fold cross validation on the training dataset to maximize the likelihood of the held out test set. The hyperparameters in this model include the concentration parameters of the Dirichlet priors, \( \alpha_0, \{ \alpha_r \}, \{ \beta_r \} \). We estimate these parameters using the fixed-point iteration method described in [44].

In Figure ??, as an example, we show the major supertopic and subtopics corresponding to the medication Miconazole. Miconazole comes as a cream, powder or spray liquid and is an antifungal agent used to treat topical skin infections. We see that these traits are effectively captured by the HPAM applied to the corpus of Wikipedia documents.

C. Feature Representations

Using the HPAM learned in the previous section we transform the high-dimensional categorical patient data into low-dimensional feature vectors as in Section III-C. These low-dimensional features can then be compared using the EMD-based similarity measure from Eq. 4. Along with labeled data, this similarity matrix can be used to learn a classifier. We refer to this approach as \( HPAM+EMD \) in the next section when presenting results. Here, we present several additional feature representation approaches to which we compare our method.

\( M \)-Categories: The first baseline we consider is a common approach used in healthcare applications when applying machine learning methods to categorical variables. Here, each category is simply mapped to its own feature/variable. Applied to our data this approach represents each patient by a feature vector with 2,164 dimensions. On average, 0.79% of the vector is nonzero.

Curated: As a proxy for a manually created taxonomy, we chose a taxonomy freely available from the website Drugs.com. Drugs.com, a privately held trust in New Zealand is regarded as the largest, freely accessible, independent medicine information website available on the Internet. It manages an extensive library of information including content U.S. Food and Drug Administration (FDA) making it an ideal proxy for our study. Applied to our dataset, we managed to classify all but 92 of the 2,164 drugs into 305 different classes. The remaining 92 medications, which were not explicitly included in the manually curated taxonomy, were included as individual features. This resulted in a feature vector with 397 features.

Wikipedia: When querying each drug name, at times multiple drug names corresponded to the same Wikipedia page. This in itself is a form of abstraction, since features that map to the same Wikipedia page can be combined into one feature. This representation results in a feature vector with 1,124 dimensions, where each dimension corresponds to a unique Wikipedia page. Given this representation, each binary feature takes the value 1 if there is at least one prescribed drug corresponding to that particular page and 0 otherwise. While we chose a binary representation here, we note that other non-binary representations are possible (e.g., set each feature according to the total number of prescribed drugs corresponding to that page).

LDA: To measure the utility of choosing HPAM over other topic modeling approaches, we also learn a latent Dirichlet allocation (LDA) model. For the purpose of comparison, we constructed this using the same number of topics as the HPAM model. As before, we optimized the hyperparameters of the Dirichlet distributions using the fixed-point iteration method described in [44]. Based on the learned topics, we construct patient feature vectors using the approach described in III-C.

HPAM: We also use the abstractions generated from the HPAM model as a baseline approach and compare it to our feature representation. This is primarily to test the utility of the EMD distance metric in estimating the distances between the different topics generated by the model.

D. Learning the Classification Models

From the dataset, we obtain labeled data corresponding to each patient risk stratification problem presented in Section IV-A. We represent each example from the labeled data according to the feature representation generated from our proposed approach as well as the other approaches described in the previous section. In this section, we compare the utility of these approaches for learning a classifier for each of the three classification tasks described earlier.

For each outcome (i.e., task), we use support vector machines (SVMs) to learn a mapping from each feature representation to patient risk. We train a linear kernel SVM for each of the baseline approaches. (We also considered an RBF kernel, however the results were no better than the linear kernel and are thus not shown here.) In addition, we learn an SVM using our precomputed \( HPAM+EMD \) kernel.

For training and testing, we generate 50 random stratified splits of 0.7 (Training) to 0.3 (Test). To account for class imbalance, we use asymmetric cost parameters. We set the cost-factor, by which training errors on positive examples
When the amount of training data is limited (e.g., Fig. 5: applying machine learning and data mining techniques to \( f = 0 \) ranges (IQR) (i.e., the first and the third quartile).

AUROC along with error bars representing the inter-quartile significance of accounting for the distances between topics. HPAM+EMD approach consistently outperforms HPAM on the test data, as we vary \( f \) repeatedly randomly subsampled for each setting of \( f \). We plot the median values of AUROC along with error bars representing the inter-quartile ranges (IQR) for the area under the operating characteristic curve (AUROC). We keep the training and test set uniform across all these classifiers for each run.

E. Performance Evaluation

We compare the classification performance of each approach based on the AUROC since it allows for meaningful comparisons even in the presence of high class imbalance. As previously noted, feature engineering is of particular importance when the number of training examples is small. To this end, in our evaluation we varied the fraction of training data utilized, \( f \), in each run described above. Training data were repeatedly randomly subsampled for each setting of \( f \). Figure 5 shows the resulting average performance of each classifier on the test data, as we vary \( f \). We plot the median values of AUROC along with error bars representing the inter-quartile ranges (IQR) (i.e., the first and the third quartile).

In Table II, we note the classification performance for \( f = 0.03 \). For this setting of \( f \) training on approximately 500 samples. For each task, the HPAM+EMD model outperforms LDA and the other baselines when the number of training samples are limited (i.e., when \( f \leq 0.1 \)). Also, we see that the HPAM+EMD approach consistently outperforms HPAM and LDA at all levels of training data used, underlying the significance of accounting for the distances between topics.

V. DISCUSSION & CONCLUSION

In this paper, we address two issues that often arise when applying machine learning and data mining techniques to categorical data:

1) the high-dimensionality of the data results in sparse feature representations, which can make it challenging to identify meaningful similarities, and

2) the complexity of the underlying relationships among categories mean that standard Euclidean distance metrics often do not apply.

To address these issues, we propose a feature-learning framework that automatically incorporates expert knowledge. Our approach has two stages. In the first stage we learn a lower-dimensional feature representation of the data. Using hierarchical topic modeling applied to unstructured data (representing expert knowledge) we identify similar categories and collapse these categories into topics. These topics are the basis for the lower-dimensional feature space. The second stage accounts for the non-uniform distances between topics/features by incorporating the ground distance between topics (i.e., the distribution over the vocabulary) when calculating distance/similarity between examples in the lower-dimensional space.

We demonstrate the utility of our approach through a case study of patients and their medications. It is important to note that the goal of this case study was not to build the best possible classifier to predict the adverse patient outcomes, but to test the utility of the abstractions learned using our approach. The proposed feature-learning framework could be extended in a number of different ways. For example, one could incorporate expert knowledge in the form of structured data. Also if the data were time-varying temporal topic models could be used to discover topics that change across the period.

In the application we considered, the categorical data had a straightforward textual representation. However, our proposed feature-learning framework applies more generally (e.g., to the analysis of demographic features of different countries where the categories can be related to any trait of the study population). As with most dimensionality reduction techniques, we note that the abstractions help only in settings where the training data are limited. Despite the growing trend toward massive datasets, in many applications (particularly in healthcare) one still encounters a small number of examples. When the number of examples is small, but the dimensionality is high, the proposed feature-learning framework can help efficiently incorporate auxiliary expert knowledge.
REFERENCES


