Learning Classifiers from Distributional Data

Harris T. Lin*, Sanghack Lee*, Ngot Bui* and Vasant Honavar
Department of Computer Science
Iowa State University
Ames, IA 50011 USA
{*htlin,shlee,bpngot,honavar*}*@iastate.edu*

***Abstract*—Many big data applications give rise to *distributional data* wherein objects or individuals are naturally represented as K-tuples of *bags of feature values* where feature values in each bag are sampled from a *feature and object specific* distribution. We formulate and solve the problem of learning classifiers from distributional data. We consider three classes of methods for learning distributional classifiers: (i) those that rely on *aggregation* to encode distributional data into tuples of attribute values, i.e., instances that can be handled by traditional supervised machine learning algorithms; (ii) those that are based on *generative models* of distributional data; and (iii) the discriminative counterparts of the generative models considered in (ii) above. We compare the performance of the different algorithms on real-world as well as synthetic distributional data sets. The results of our experiments demonstrate that classifiers that take advantage of the information available in the distributional instance representation outperform or match the performance of those that fail to fully exploit such information.**

***Keywords*-classifier; distributional data**

I. Introduction

The standard classification problem entails assigning an instance x from an instance space $\mathcal{X}$ (that is typically modeled by a tuple of measurements or attribute values) a label from a set of mutually exclusive classes $\mathcal{C}$. A classifier h is a mapping $h : \mathcal{X} \mapsto \mathcal{C}$. The goal of learning in such a setting is to identify a classifier from a space of classifiers $\mathcal{H}$, one that optimizes a desired performance measure, e.g., accuracy of the classifier. Consider for example, a clinical diagnosis scenario which calls for classifying a patient as healthy or suffering from a particular illness based on a set of tests or measurements. Suppose the k^{th} feature or test result takes values from the domain Δ_k. The standard approach is to represent each patient o_i by a K-tuple or a K-dimensional vector of feature values $x_i = (x_{i1}, \ldots, x_{iK}) \in \Delta_1 \times \Delta_2 \times \cdots \Delta_K$ (where each $x_{ik} \in \Delta_k$) that encode the results of specific medical tests or measurements [1]. The goal is to predict the class label $c_i \in \{\textit{Healthy}, \textit{Ill}\}$ for each patient o_i. However, because of the variability associated with physiological measurements such as the heart rate, blood pressure, or body temperature of an individual, it is often necessary to repeat the tests or measurements in order to arrive at a reliable diagnosis. If the measurements are synchronous, then it is possible to represent each patient by a collection or bag of instances (K-tuples of feature values) and model the problem of predicting the class label for each patient as a multiple instance learning problem [2], [3]. However, because the different tests or measurements have different sources of variability associated with them, it is not uncommon to carry out the tests in an asynchronous fashion, with each test repeated different number of times. Hence, as illustrated in Table I, it is far more meaningful to model the input to the classifier, in this case, an individual o_i, by a K-tuple of bags (multi-sets) of feature values $(B_1^i, \ldots, B_K^i)$ where each B_k^i represents a bag of values of the k^{th} feature of object o_i, sampled from the specific feature and individual specific distribution. Note that, in general, the size of the bag B_k^i can differ from feature to feature and for a given feature, from one object to another.

Many big data applications give rise to *distributional data* wherein objects or individuals are naturally represented as K-tuples of *bags of feature values* where feature values in each bag are sampled from a *feature and object specific* distribution. We refer to the resulting representation $x_i = (B_1^i, \ldots, B_K^i)$ of an object o_i as the *distributional instance* (DI) representation of o_i. We refer to the problem of learning classifiers that predict the class labels of distributional instances as the *distributional instance learning* (DIL) problem. One way to apply traditional approaches to classification in this setting is to simply replace each bag of feature values B_k^i by an aggregate value, e.g., the mean or mode computed from the observed values of the feature for a given individual. However, such an aggregation process can result in significant loss of useful information. It is much more natural to view the input to the classifier as a K-tuple of bags of attribute values. Against this background, we formulate and solve the DIL problem.

We consider representative algorithms from three classes of methods for DIL: (i) those that rely on *aggregation* to encode distributional data into tuples of attribute values, i.e., instances that can be handled by traditional supervised machine learning algorithms; (ii) those that are based on *generative models* of distributional data; and (iii) the discriminative counterparts of the generative models considered in (ii) above. We compare the performance of the different algorithms on two real-world as well as one synthetic dis-

*These authors contributed equally in this work.

Table I
AN EXAMPLE OF A DISTRIBUTIONAL DATA SET OF THREE PATIENTS WITH THEIR FOUR ATTRIBUTES WHERE STATUS REPRESENTS THE CLASS LABEL.

Status	Body Temperature in F°	Heart Rate in BPM	Blood Pressure Systolic in mmHg	Blood Pressure Diastolic in mmHg
Healthy	{98.5, 98.8, 99.0}	{70, 68}	{100, 106}	{75, 77}
Ill	{99.2, 100.3, 98.6, 99.0, 98.4}	{80, 75, 83, 76}	{120, 116, 126}	{76, 76, 83}
Healthy	{98.6}	{61, 69}	{95}	{65}

tributional data sets. The results of our experiments demonstrate that DIL algorithms that take advantage of the information available in the distributional instance representation outperform or match the performance of their counterparts that fail to fully exploit such information. We conclude with a brief summary of the main results, discussion of related work, and some directions for further research.

II. DISTRIBUTIONAL INSTANCE CLASSIFICATION PROBLEM

We introduce some key definitions before proceeding to formulate the problem of learning classifiers from distributional data. For brevity subscripts are omitted when they are clear from context.

Definition 1 (Distributional Instance Representation)**.** Let $\Delta_1, \ldots, \Delta_K$ be K sets (discrete of continuous) that correspond to the domains of a finite number (K) of measurable attributes of objects to be classified, and $\mathcal{C}$ a finite set of class labels. A Distributional Instance representation x_i of an object or individual o_i is a K-tuple of bags (multi-sets) of feature values $x_i = (B_1^i, \ldots, B_K^i)$ where each B_k^i represents a bag of values of the k^{th} feature of object o_i, sampled from the specific feature and individual specific distribution. We denote by s_k the size of k^{th} domain, $s_k = |\Delta_k|$.

Note that the widely used bag of words representation of text is a special case of distributional representation where $K = 1$ and Δ_1 is simply the vocabulary of the document collection.

Definition 2. Let $c_i \in \mathcal{C}$ be the class label of x_i. A *distributional data set* $D = \{(x_1, c_1), \ldots, (x_n, c_n)\}$ is a multi-set of labeled *distributional instances*.

Example 1. Table I shows an example of a distributional data set consisting of three objects (patients).

Definition 3 (Distributional Classifier)**.** Each classifier h accepts as input, a distributional instance x, and outputs a predicted class label $h(x) \in \mathcal{C}$.

Definition 4 (Distributional Classifier Learning Problem)**.** Given a distributional data set D, a hypothesis class $\mathcal{H}$ of classifiers, and a performance criterion f, a distributional classifier learning algorithm L outputs a *distributional classifier* $h \in \mathcal{H}$ that optimizes f.

III. DISTRIBUTIONAL INSTANCE LEARNING ALGORITHMS

We consider three basic approaches to DIL: (i) those that rely on *aggregation* to encode distributional data into tuples of attribute values, i.e., instances that can be handled by traditional supervised machine learning algorithms; (ii) those that are based on *generative models* of distributional data; and (iii) the discriminative counterparts of the generative models considered in (ii) above.

A. Aggregation

Here we represent each bag of features in the DI representation of an instance by a single value, by applying a suitable aggregation function, e.g., min, max, $average$ for continuous Δ and $mode$ for discrete Δ. Hence we reduce the data set into a traditional attribute-value data set where each instance is represented by a finite number of attributes each of which takes a single value from the set of possible values for the corresponding attribute. This approach reduces the problem of learning from distributional data to the standard supervised learning problem which can be solved using a variety of existing supervised learning algorithms [1], [4].

Within this framework, we consider a variety of sophisticated aggregation schemes proposed in [5]. Without loss of generality, consider a distributional data set D in which the distributional instances are encoded using DI representation and class labels are binary, i.e., $\mathcal{C} = \{+, -\}$. Suppose that B_k^i is the bag of values of k^{th} attribute of an instance x_i. After [5], we define $V_k^i = \left(v_{k1}^i, \ldots, v_{ks_k}^i\right)$ to be a vector of counts (or histogram) of values in B_k^i where v_{kt}^i is the number of occurrences of the t^{th} value $d_{kt} \in \Delta_k$. Next we define an unconditional reference vector as $V_k^{(*)} = \sum_i V_k^i$, and also a class-conditional reference vector for $c \in \mathcal{C}$ as $V_k^{(c)} = \sum_i \delta_{c,c_i} V_k^i$ where δ is a Kronecker delta function. A number of aggregation schemes can be defined using various measures of distance between V_k^i and the reference vectors [5] (see below). Let $DIST$ be a set of M distance functions between two vectors such as cosine or Euclidean, then we describe three aggregation schemes as follows.

1) Unconditional vector distances (UCVD): We compute an M-element vector $(dist_m(V_k^{(*)}, V_k^i))_{m=1}^M$ where $dist_m \in DIST$. We concatenate the feature vector representations from each of the K bags of features of x_i to obtain a single feature vector of length MK.

2) Class-conditional vector distances (CCVD): We compute a $|\mathcal{C}|M$-sized vector, e.g., $(dist_m(V_k^{(+)}, V_k^i),\ dist_m(V_k^{(-)}, V_k^i))_{m=1}^M$. The rest follows the scheme of UCVD and reduces into a traditional attribute-value data set where each instance is a vector of length $|\mathcal{C}|MK$.
3) Differences of class-conditional vector distances (DCCVD): We compute the pair-wise difference between every two class-conditional vector distances, resulting with a vector of size $M|\mathcal{C}|(|\mathcal{C}|-1)/2$, e.g., $(dist_m(V_k^{(+)}, V_k^i) - dist_m(V_k^{(-)}, V_k^i))_{m=1}^M$.

By applying this process to each of the distributional instances in the data set D, we can effectively reduce the problem of learning distributional classifiers to the well-studied problem of supervised learning of classifiers in the traditional setting where each object to be classified is represented by a tuple of attribute values.

B. Generative Models

We consider a joint distribution $p(B_1, \ldots, B_K, c)$. For simplicity, under the naive Bayes assumption that bags of features are conditionally independent given the class label c the most probable class label is given by:

$$\begin{aligned} h_{NB}(x) &\triangleq \arg\max_{c \in \mathcal{C}} p(c \mid B_1, \ldots, B_K) \\ &= \arg\max_{c \in \mathcal{C}} p(c) \prod_{k=1}^{K} p(B_k \mid c). \end{aligned}$$

We can now consider a variety of models for $p(B_k \mid c)$ including those based on Bernoulli or multinomial event models [6], Dirichlet distribution [7], [8] or Dirichlet-multinomial (Polya) distribution [9], [8]. We denote these models by NB(Ber), NB(Mul), NB(Dir), and NB(Pol) respectively, and outline each of them below.

Let $b_{kt} \in \{1, 0\}$ denote the presence or absence of $d_{kt} \in \Delta_k$ in an attribute bag B_k and, similarly, let v_{kt} denote the number of occurrences of d_{kt}. A class-conditional bag probability, $p(B_k \mid c)$, can be modeled by event models such as Bernoulli (1) or multinomial (2):

$$p(B_k \mid c; \boldsymbol{\theta}) \triangleq \prod_{t=1}^{s_k} \theta_{ckt}^{b_{kt}} (1 - \theta_{ckt})^{1 - b_{kt}} \tag{1}$$

$$p(B_k \mid c; \boldsymbol{\theta}) \triangleq p(|B_k|) \frac{(\sum_{t=1}^{s_k} v_{kt})!}{\prod_{t=1}^{s_k} v_{kt}!} \prod_{t=1}^{s_k} \theta_{ckt}^{v_{kt}} \tag{2}$$

where $\theta_{ckt} = p(d_{kt} \mid c)$.

Next, the Dirichlet distribution (3) allows us to treat B_k as a sample from a distribution which, in turn, is drawn from another distribution as follows:

$$\begin{aligned} p(B_k \mid c; \boldsymbol{\alpha}) &\triangleq p(\bar{V}_k \mid c) \\ &\triangleq \mathcal{D}(\boldsymbol{\alpha}_{ck}) \\ &= \frac{\Gamma(\sum_{t=1}^{s_k} \alpha_{ckt})}{\prod_{t=1}^{s_k} \Gamma(\alpha_{ckt})} \prod_{t=1}^{s_k} \bar{v}_{kt}^{\alpha_{ckt}-1} \end{aligned} \tag{3}$$

where $\boldsymbol{\alpha}_{ck} = (\alpha_{ck1}, \ldots, \alpha_{cks_k})$ is a vector parameter of Dirichlet distribution for class $c \in \mathcal{C}$ and $\bar{V}_k = (\bar{v}_{k1} \cdots \bar{v}_{ks_k})$ is the *normalized* vector of counts of values in B_k with $\bar{v}_{kt} = v_{kt} / \sum_t v_{kt}$. Finally, we describe the Dirichlet-multinomial (Polya) distribution (4) that compounds a Dirichlet distribution with a multinomial distribution:

$$\begin{aligned} p(B_k \mid c; \boldsymbol{\alpha}) &\triangleq p(V_k \mid c) \\ &\triangleq \int p(V_k; \boldsymbol{\theta}_{ck})\, p(\boldsymbol{\theta}_{ck}; \boldsymbol{\alpha}_{ck})\, d\boldsymbol{\theta}_{ck} \\ &= \frac{\Gamma(\sum_t \alpha_{ckt})}{\Gamma(\sum_t v_{kt} + \alpha_{ckt})} \prod_{t=1}^{s_k} \frac{\Gamma(v_{kt} + \alpha_{ckt})}{\Gamma(\alpha_{ckt})} \end{aligned} \tag{4}$$

where $\boldsymbol{\theta}_{ck} = (\theta_{ck1}, \ldots, \theta_{cks_k})$ is a vector of multinomial parameters.

For all four models, their parameters, which is a set of parameters for each class and for each attribute, are estimated by maximum likelihood employing the Laplace correction.

C. Discriminative Models

We consider the discriminative counterparts of the generative models described in Sec. III-B using standard techniques for transforming a generative model into its discriminative counterpart (e.g., a naive Bayes model into a logistic regression model [10]). Discriminative models [11] can be acquired by plugging in four different distributions (shown in Sec. III-B) for $p(B \mid c)$ in the following equation.

$$\begin{aligned} &p(c = 1 \mid x) \\ &\triangleq \frac{1}{1 + \exp\left(\ln \frac{p(c=0)}{p(c=1)} + \sum_{k=1}^{K} \ln \frac{p(B_k \mid c=0)}{p(B_k \mid c=1)}\right)} \end{aligned} \tag{5}$$

There exists an equivalent parametric form for posterior distribution, $p(c \mid x; \mathbf{w})$. We estimate a vector of parameters $\mathbf{w}$ as

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \sum_{i=1}^{n} \ln p(c_i \mid x_i; \mathbf{w}) - \lambda ||\mathbf{w}||_2^2. \tag{6}$$

It is possible to adopt ℓ^2–regularization by setting $\lambda > 0$ to reduce over-fitting to training data. Given the estimated parameter $\mathbf{w}^*$, prediction on a distributional data instance is given by $h(x) = \arg\max_{c \in \mathcal{C}} p(c \mid x; \mathbf{w}^*)$. Discriminative models for Bernoulli and multinomial distributions define

posterior distribution as below (respectively):

$$p\left(c=1 \mid x; \mathbf{w}\right) \triangleq \frac{1}{1+\exp\left(w_0 + \sum_{k,t} b_{kt} w_{kt}\right)}$$

$$p\left(c=1 \mid x; \mathbf{w}\right) \triangleq \frac{1}{1+\exp\left(\ln \frac{p(c=0)}{p(c=1)} + \sum_{k,t} v_{kt} w_{kt}\right)}$$

which are logistic regression models. Parameters for these two models can be estimated with optimization tools specialized in logistic regression (e.g., [12]). For Dirichlet and Polya distributions, we first formulate $p\left(c \mid x; \boldsymbol{\alpha}\right)$ by substituting $p\left(B \mid c\right)$ in Eq. 5 with Eq. 3 and 4, respectively. By setting $\mathbf{w} = \ln\left(\boldsymbol{\alpha}\right)$, we drop the constraint $\boldsymbol{\alpha} > 0$ and employ an unconstrained gradient ascent method[1].

IV. Experimental Results

We report results of experiments designed to answer the following questions.

(i) How do representative DIL algorithms within each of the three classes of DIL methods outlined in Sec. III compare with each other?

(ii) How do the three classes of DIL methods compare with each other?

(iii) How do classifiers that take advantage of the information available in the distributional instance representation compare with their counterparts that reduce DIL to traditional supervised learning (by transforming distributional instances into tuples of attribute values)?

We use two real world data sets to address questions (i) and (ii) above, and we use a synthetic data set to address question (iii).

A. Experiment I

1) Data sets and Experimental Setup: Due to lack of publicly available distributional data benchmarks, we turn to available data sets that can be modeled as distributional data.

The first data set, the Last.fm data set, is crawled from a social music network Last.fm[2] using its API[3] (an example is shown in Fig. 1). We select two disjoint groups that contain approximately equal number of users (2098/2081). We collect the *track*, *artist*, *track's tags*, and *artist's tags* favored by each user and represent them as bags. All collections of tags are processed with stop-word removing and stemming, using Apache Lucene[4]. We use only tracks and artists whose number of occurrences greater or equal than 45 and 100, correspondingly. The result is a distributional data set of 8340 tracks attributed to one or more of the 3753 artists.

[1]In our experiments, we used Hessian-free Newton method implemented by Mark Schmidt. See http://www.di.ens.fr/~mschmidt/Software/minFunc.html

[2]http://www.last.fm/

[3]http://www.last.fm/api

[4]http://www.luccene.apache.org/

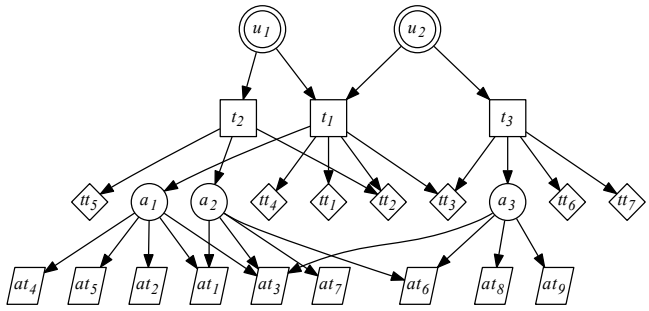

Figure 1. A portion of Last.fm data set with entities and links among them. Entities are users u, tracks t, track's tags tt, artists a, and artist's tags at. Corresponding distributional data instances are $x_1 = \{\{t_1, t_2\}, \{a_1, a_2\}, \{tt_1, \ldots tt_5\}, \{at_1, \ldots, at_7\}\}$ and $x_2 = \{\{t_1, t_3\}, \{a_1, a_3\}, \{tt_1, \ldots, tt_7\}, \{at_1, \ldots at_6, at_8, at_9\}\}$.

Table II
Data set statistics. Since bags contain duplicate elements, the size of a bag may exceed the domain size.

Data set (+/− count)	Attribute	Domain Size	Average Bag Size
Last.fm (2098/2081)	track	8340	120
	track's tag	5077	15874
	artist	3753	58
	artist's tag	3640	10535
Splog (695/693)	word	7968	2627
	anchor	7819	316
	URL	7833	937
	HTML tag	152	538

Likewise, we eliminate the track tags and artist tags that occur fewer than 350 times and 120 times respectively.

The second data set is obtained from the Splog (spam blogs) data set which contains 700 authentic and 700 spam blogs in HTML format [13]. For each blog, we extracted four attributes: a bag of words, a bag of anchors (words marked up with hyperlinks), a bag of URLs, and a bag of HTML tags. Then, similarly we remove infrequent elements for each attribute and remove instances with missing values. The statistics of the two data sets are shown in Table II.

We compare the three classes of DIL described in Sec. III. Specifically, for the aggregation models we select: (i) $mode$ aggregation combined with a simple naive Bayes classifier (denoted Mode+NB); (ii) CCVD combined with a Gaussian naive Bayes (denoted CCVD+NB); and (iii) CCVD combined with a logistic regression (denoted CCVD+LR). We only selected CCVD among the complex aggregation schemes since it is reported to yield more accurate classifier as in [5]. For ℓ^2-regularized discriminative models, we set $\lambda = 1$ in Eq. 6 without optimization. We evaluate both data sets using 10-fold cross-validation and draw their ROC (Receiver Operating Characteristic) curves with AUC (Area Under Curve).

2) Results: Among the models based on aggregation, CCVD+LR outperforms the rest in both accuracy and AUC

Table III
RESULTS FOR EXPERIMENT I. EACH NUMBER IN PARENTHESES IS STANDARD DEVIATION FROM 10-FOLD CROSS-VALIDATION. BOLDED NUMBERS REPRESENT BEST RESULTS FOR EACH COLUMN BASED ON PAIRED t-TEST ON 10-FOLD CROSS VALIDATION WITH ALPHA = 0.05.

Hypothesis Class	Model	Last.fm		Splog	
		Accuracy	AUC	Accuracy	AUC
Aggregation	Mode+NB	73.01 (2.17)	79.72 (2.20)	69.09 (2.70)	85.62 (1.91)
	CCVD+NB	76.21 (1.76)	82.42 (1.72)	63.49 (4.39)	79.35 (4.54)
	CCVD+LR	**81.55 (2.15)**	**89.64 (1.64)**	86.46 (4.26)	93.00 (3.60)
Generative	NB(Ber)	72.70 (2.76)	82.43 (3.04)	79.04 (3.55)	87.83 (3.57)
	NB(Mul)	**81.98 (2.17)**	86.38 (1.81)	88.69 (2.53)	93.51 (2.97)
	NB(Dir)	**81.65 (2.11)**	**89.52 (1.62)**	78.53 (2.36)	84.07 (3.06)
	NB(Pol)	**82.12 (2.05)**	**88.97 (1.33)**	**88.98 (3.68)**	93.87 (2.52)
Discriminative	DM(Ber)	79.59 (2.69)	87.84 (1.76)	89.34 (2.60)	95.57 (1.41)
	DM(Mul)	76.29 (2.78)	82.60 (2.55)	86.67 (3.46)	92.55 (2.64)
	DM(Dir)	79.47 (2.74)	87.92 (1.81)	89.63 (2.69)	95.87 (1.19)
	DM(Pol)	79.97 (2.46)	87.80 (1.92)	90.56 (2.27)	96.10 (1.36)
	DM_{ℓ^2}(Ber)	**81.36 (1.88)**	89.15 (1.61)	89.70 (2.33)	95.85 (1.37)
	DM_{ℓ^2}(Mul)	80.35 (1.98)	86.54 (1.37)	86.38 (3.19)	95.51 (2.56)
	DM_{ℓ^2}(Dir)	**80.21 (2.82)**	88.36 (1.75)	89.70 (2.49)	95.94 (1.17)
	DM_{ℓ^2}(Pol)	**80.98 (2.36)**	88.78 (1.79)	**91.07 (2.02)**	**96.25 (1.32)**

measures, thus confirming the conclusions reported in [5]. Among generative models, NB(Pol) outperforms the rest on both accuracy and AUC, while NB(Dir) is also competitive for the Last.fm data set. For discriminative models, DM_{ℓ^2}(Pol), DM_{ℓ^2}(Dir), and DM_{ℓ^2}(Ber) are equally competitive on the Last.fm data set; and DM_{ℓ^2}(Pol) outperforms the rest for the Splog data set. The ℓ^2-regularized discriminative models generally outperform their un-regularized counterparts (with the exception of DM_{ℓ^2}(Mul) on the accuracy measure for the Splog data set). Indeed, in this case un-regularized models are special cases of regularized models with the hyperparameter $\lambda = 0$; and in principle, the results of our regularized models could be improved further by optimizing the hyperparameter.

To answer the second question, we compare the best model from each approach. Considering the accuracy measure alone, NB(Pol) and DM_{ℓ^2}(Pol) consistently outperforms the rest for both data sets. We observe that the best models under the AUC measure is a subset of the best models under the accuracy measure, and this is possibly because in general AUC is statistically more discriminant than accuracy [14]. If we now consider the AUC measure alone, we observe that NB(Pol), NB(Dir), and CCVD+LR are equally competitive for the Last.fm data set, while NB(Pol) outperforms the rest for the Splog data set. In summary, NB(Pol) and DM_{ℓ^2}(Pol) show similar performance.

B. Experiment II

The second experiment is designed to examine as to how the classifiers that take advantage of the information available in the distributional instance representation compare with those that do not fully exploit such information. Recall that among the models described in Sec. III, only NB(Dir), NB(Pol), DM(Dir), and DM(Pol) model distributions of distributional instances. Since naive Bayes models and discriminative models have the same likelihood $p(B \mid c)$, we only consider naive Bayes models in this experiment.

We deliberately crafted scenarios in which some of the models that do not take advantage of the information available in the would fail to discriminate between two classes. For example, Mode+NB would fail if $p(c \mid mod(B))$ is close to 0.5; NB(Mul) would fail if the parameters for both classes (i.e. its sufficient statistics $\bar{V}_k^{(+)}$ and $\bar{V}_k^{(-)}$) are similar; and CCVD would fail if two reference vectors are similar and their distances from the DIL representation of the object to be classified are also similar.

1) Data Set and Experimental Setup: We generated a synthetic data set with binary class ($\mathcal{C} = \{+, -\}$) and a single attribute ($K = 1$) by combining samples from two Polya distributions. The two Polya distributions have domains $\Delta' = \{0, 1\}$ and $\Delta'' = \{2, 3\}$ whose Dirichlet parameters are $\boldsymbol{\alpha}'$ and $\boldsymbol{\alpha}''$ respectively. The generation process is shown in Fig. 2, where Dirichlet parameters are defined deterministically as follows,

$$\boldsymbol{\alpha}' = \begin{cases} (z, z) & c = + \\ (z^{-1}, z^{-1}) & c = - \end{cases}, \quad \boldsymbol{\alpha}'' = \begin{cases} (z^{-1}, z^{-1}) & c = + \\ (z, z) & c = - \end{cases}$$

and we draw a bag B' from Polya distribution as follows,

$$\begin{aligned} \theta' &\sim \text{Dir}(\boldsymbol{\alpha}') \\ x' &\sim \text{Mult}(\boldsymbol{\theta}') \\ B' &\triangleq \left\{x'_1, \ldots, x'_{|B'|}\right\}. \end{aligned}$$

Similarly, B'' are drawn from Polya distribution with $\boldsymbol{\alpha}''$. Finally the bag B for an instance is defined as $B' \cup B''$. The process generates a set of n instances with a label c where each instance is characterized by z, $|B'|$, and $|B''|$ which we will denote by $f(n, c, z, |B'|, |B''|)$. A balanced data set with $2n$ instances can be described as $f(n, +, z, |B'|, |B''|) \cup f(n, -, z, |B'|, |B''|)$.

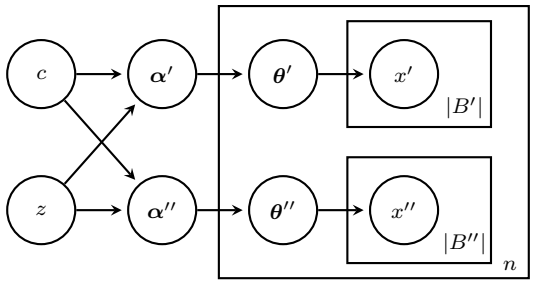

Figure 2. Generation process for the synthetic data. We control c and z to generate n data with bags of size $|B| = |B'| + |B''|$. Dirichlet parameter $\boldsymbol{\alpha}'$ and $\boldsymbol{\alpha}''$ is deterministically generated given c and z.

We defined three groups of balanced data sets where generations of data sets in a group only differ in z. The first group is balanced data sets with $n = 500$ and $|B'|, |B''| = 40$, and the second group differs by $|B''| = 80$. A data set in the third group is defined as the *union* of two balanced data sets of 1000 instances where their $(|B'|, |B''|)$ are $(40, 80)$ and $(80, 40)$, respectively. Given a fixed z, a data set in the first two groups can be represented as

$$\bigcup_{c\in\{-,+\}} f\left(500, c, z, 40, |B''|\right)$$

where $|B''|$ is 40 or 80 respectively, while a data set in the third group is

$$\bigcup_{c\in\{-,+\}} f\left(500, c, z, 40, 80\right) \cup f\left(500, c, z, 80, 40\right).$$

We repeat this process five times with different random seeds to obtain five different distributional data sets. We estimate the accuracy of the classifiers using 10-fold cross-validation. We report the average accuracy of the DIL methods over the five data sets, in each case, estimated using 10-fold cross-validation.

2) Results: Fig. 3 shows the results of this experiment. We observe that all naive Bayes models behave similarly in all three groups. In particular, accuracies for NB(Dir), NB(Pol), and NB(Ber) increase as z increases, while NB(Mul) is unable to discriminate between the two classes. NB(Mul) fails because $\bar{V}_k^{(+)}$ and $\bar{V}_k^{(-)}$ are designed to be similar by setting the Dirichlet parameters of both classes to be complement of each other.[5] Performance of NB(Dir) matches with NB(Pol), because the bag length is constant for all instances. As z increases, the good performance of NB(Ber) can be explained by observing it is more likely that the bag of feature values for the negative class contain all 0's or all 1's.

[5]Take B' for example, for the positive class, the bags are likely to contain approximately equal numbers of 0's and 1's whereas for the negative class, the bags are likely to contain either a majority of 0's or a majority of 1's; hence this ensures that the expected multinomial parameters for both classes are $(0.5, 0.5)$.

Interestingly, the behaviors of Mode+NB and CCVD+LR vary across three groups of experiments. In the case of Mode+NB, it fails in Group 2 because it is most likely that one of two values of the second bag B'' is chosen as mode for an instance *independent* to the label of the instance[6]; whereas it fails in Group 3 because all values are equally likely to be the mode. In the case of CCVD+LR, in all three groups, the expected class-conditional reference vectors are identical for both classes. The first and the third group guarantee the expected distance of positive instances and that of negative instances are the same. For the second group, the expected distances differ due to the asymmetry in $|B'|$ and $|B''|$.

Overall, these experiments clearly demonstrate that DIL methods that take advantage of the information available in the distributional instance representation can potentially outperform those that do not fully exploit such information.

V. Summary and Discussion

A. Summary

Many big data applications naturally give rise to *distributional data* wherein objects or individuals to be labeled are naturally represented as K-tuples of bags of feature values sampled from a *feature and object specific* distribution. We have introduced distributional instance learning problem, i.e., the problem of learning classifiers from distributional data. We have considered three classes of methods for learning such classifiers: (i) those that rely on *aggregation* to encode distributional data into tuples of attribute values, i.e., instances that can be handled by traditional supervised machine learning algorithms; (ii) those that are based on *generative models* of distributional data; and (iii) the discriminative counterparts of the generative models considered in (ii) above. We have compared the performance of the different algorithms on real-world as well as synthetic distributional data sets. The results of our experiments demonstrate that DIL algorithms that take advantage of the information available in the distributional instance representation outperform or match the performance of their counterparts that make use of aggregation schemes that discard such information.

B. Related Work

The DIL problem is a generalization of (i) the traditional supervised learning problem where an object to be classified is represented by a tuple of attribute values [1], and (ii) the problem of learning document classifiers and image classifiers using a bag of words representation of documents [6], and bag of visual words representation of images [15] (which represent special cases of learning distributional classifiers with $K = 1$).

[6]With a rare chance, mode can be a value of the bag B' if an instance contains three values with 40 counts. In other words, B' consists of only one value and B'' consists of even number of two values.

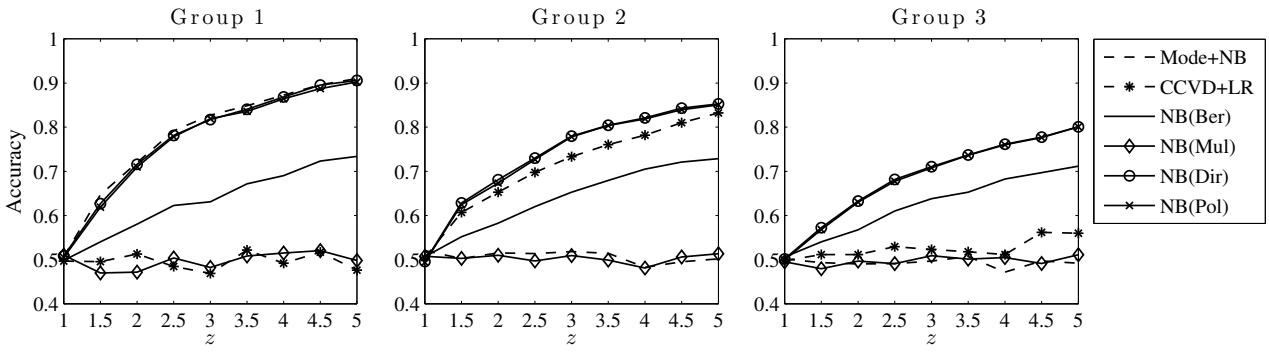

Figure 3. Accuracies of six classifiers on different groups of data sets where each synthetic data set consists of samples drawn from a stochastic process which is a composition of two Polya processes.

The problem of learning distributional classifiers is related to the multiple instance learning (MIL, [2], [3]), where an object to be classified is represented as a *bag of instances* and each instance is represented by a tuple of feature values. Only the label of the bag is specified in the training set, and MIL assumes that a bag is labeled negative if and only if all of its instances are negative, and a bag is labeled positive if and only if at least one of its instances is positive [2]. Since then, recent work on MIL has relaxed the standard MIL assumption to allow all the instances in a bag to contribute to the bag's label. However, unlike MIL which models an object to be classified by a *bag of tuples of feature values*, DIL models an object to be classified as a *tuple of bags of feature values*. Both MIL and DIL reduce to the same problem when the number of features (K) is one.

DIL bears some resemblance to Latent Dirichlet Allocation (LDA, [16]) or more generally probabilistic topic models [17], which are generative probabilistic models for documents that model each word in a document by a mixture of latent topics (i.e., distributions over a fixed vocabulary). While the topic models are typically learned in an unsupervised setting, a document's topic distribution can be effectively used for document classification as demonstrated in [16]. Supervised topic models such as sLDA proposed by [18] can be seen as a special case of DIL where the number of bags (K) in the DI representation of the objects to be classified (in this case, documents) is equal to one. However, recent work on topic models, e.g., Block-LDA [19], Nubbi [20], and Link-PLSA-LDA [21] has begun to explore topic models for objects with multiple features ($K > 1$). Block-LDA models documents where each document contains collections of entities of different types (which can be modeled by different bags that make up a distributional instance in DIL). DIL can be seen as a supervised variant of such topic models, i.e., supervised topic models defined over multiple features ($K > 1$) with discrete domain.

DIL can be used to model learning from relational data [22] and RDF data [23]. For example, relational Bayesian classifiers [24], [25] model an object (nodes in a network) to be classified using bags of values of features from those objects that are related to it via relational links.

C. Future Work

We have explored only some of the simplest approaches to DIL. It would be interesting to explore DIL models that account for dependencies between feature values within a bag as well as between bags. It would be useful to consider variants of kernel methods (e.g., SVM) that use kernel functions to compute similarity between distributional instances, e.g., adaptations of kernel functions for distributions [26], [27] to the setting with $K > 1$. Of particular interest in this context are support measure machines (SMM) introduced by [28] which extend SVMs by representing distributions as mean embeddings in the reproducing kernel Hilbert space, which allows the application of standard kernel methods for classifying probability distributions. One subtle difference between the DIL formulation in this paper and that of SMMs is that the input to an SMM classifier is a probability distribution whereas the input to a distributional classifier is a K-tuple of bags where each bag is a finite sample drawn from a feature and object specific (albeit unknown) distribution. It would be interesting to consider DIL variants of decision trees, random forests, support vector machines, nearest neighbor classifiers, etc. as well as variants of DIL models and algorithms that can handle ordinal or continuous valued features. Of particular interest are DIL algorithms that can effectively handle massive data sets with billions or trillions of objects and millions or billions of attributes each represented using bags ranging in size from tens of thousands to millions of values.

ACKNOWLEDGEMENTS

This work is supported in part by a grant (IIS 0711356) from the National Science Foundation (NSF) and in part by the Iowa State University Center for Computational

Intelligence, Learning, and Discovery. The work of Vasant Honavar was supported by the NSF, while working at the Foundation. Any opinion, finding, and conclusions contained in this article are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[2] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, pp. 31 – 71, 1997.

[3] Z.-H. Zhou, "Multi-instance learning: A survey," Department of Computer Science and Technology, Nanjing University, Tech. Rep., 2004.

[4] K. P. Murphy, *Machine Learning: a Probabilistic Perspective*. MIT Press, 2012.

[5] C. Perlich and F. Provost, "Distribution-based aggregation for relational learning with identifier attributes," *Machine Learning*, vol. 62, no. 1-2, pp. 65–105, Feb. 2006.

[6] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *AAAI-98 Workshop on Learning for Text Categorization*. AAAI Press, 1998, pp. 41–48.

[7] T. Ferguson, "A bayesian analysis of some nonparametric problems," *The annals of statistics*, pp. 209–230, 1973.

[8] T. P. Minka, "Estimating a dirichlet distribution," Tech. Rep., 2012.

[9] R. E. Madsen, D. Kauchak, and C. Elkan, "Modeling word burstiness using the dirichlet distribution," in *Proceedings of the 22nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: ACM, 2005, pp. 545–552.

[10] G. Bouchard and B. Triggs, "The Tradeoff Between Generative and Discriminative Classifiers," in *16th IASC International Symposium on Computational Statistics (COMPSTAT '04)*, Prague, Czech Republic, 2004, pp. 721–728.

[11] T. P. Minka, "Discriminative models, not discriminative training," Microsoft Research, Cambridge, UK, Tech. Rep., 2005.

[12] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[13] P. Kolari, T. Finin, and A. Joshi, "Svms for the blogosphere: Blog identification and splog detection," in *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, 2006, pp. 92–99.

[14] J. Huang and C. Ling, "Using auc and accuracy in evaluating learning algorithms," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 3, pp. 299 – 310, march 2005.

[15] P. Tirilly, V. Claveau, and P. Gros, "Language modeling for bag-of-visual words image categorization," in *Proceedings of the 7th ACM International Conference on Image and Video Retrieval*, ser. CIVR '08. New York, NY, USA: ACM, 2008, pp. 249–258.

[16] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Mar. 2003.

[17] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, pp. 77–84, Apr. 2012.

[18] D. Blei and J. McAuliffe, "Supervised topic models," in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 2008, pp. 121–128.

[19] R. Balasubramanyan and W. W. Cohen, "Block-lda: Jointly modeling entity-annotated text and entity-entity links," in *Proceedings of the Eleventh SIAM International Conference on Data Mining*, 2011, pp. 450–461.

[20] J. Chang, J. Boyd-Graber, and D. M. Blei, "Connections between the lines: augmenting social networks with text," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09. New York, NY, USA: ACM, 2009, pp. 169–178.

[21] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen, "Joint latent topic models for text and citations," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08. New York, NY, USA: ACM, 2008, pp. 542–550.

[22] L. Getoor and B. Taskar, *Introduction to Statistical Relational Learning*. The MIT Press, 2007.

[23] F. Manola and E. Miller, Eds., *RDF Primer*, ser. W3C Recommendation. World Wide Web Consortium, February 2004. [Online]. Available: http://www.w3.org/TR/rdf-primer/

[24] J. Neville, D. Jensen, and B. Gallagher, "Simple estimators for relational bayesian classifiers," in *Proceedings of the Third IEEE International Conference on Data Mining*, 2003, pp. 609–612.

[25] H. T. Lin, N. Koul, and V. Honavar, "Learning relational bayesian classifiers from RDF data," in *Proceedings of the 10th international conference on The semantic web*, vol. 1, 2011, pp. 389–404.

[26] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Advances in Neural Information Processing Systems 11*. The MIT Press, 1998, pp. 487–493.

[27] T. Jebara, R. I. Kondor, and A. Howard, "Probability product kernels," *Journal of Machine Learning Research*, vol. 5, pp. 819–844, 2004.

[28] K. Muandet, B. Schölkopf, K. Fukumizu, and F. Dinuzzo, "Learning from distributions via support measure machines," *CoRR*, vol. abs/1202.6504, 2012.