To appear in the Tenth International Conference on Machine Learning and Applications, 2011

Relational Classifiers in a Non-relational world: Using Homophily to Create Relations

Sofus A. Macskassy
Fetch Technologies
El Segundo, CA

***Abstract*—Research in the past decade on statistical relational learning (SRL) has shown the power of the underlying network of relations in relational data. Even models built using *only* relations often perform comparably to models built using sophisticated relational learning methods. However, many data sets–such as those in the UCI machine learning repository–contain no relations. In fact, many data sets either do not contain relations or have relations which are not helpful to a specific classification task. The question we investigate in this paper is whether it is possible to construct relations such that relational inference results in better classification performance than non-relational inference. Using simple similarity-based rules to create relations and weighting the strength of these relations using homophily on instance labels, we test whether relational inference techniques are applicable–in other words, do they perform comparably to standard machine learning algorithms. We show, in an experimental study on 31 UCI benchmark data sets, that relational inferecne wins more than any of the 6 classifiers we compare against, including a transductive SVM, and that it wins the majority of the time when compared against any one of them.**

***Index Terms*—machine learning; statistical relational learning; feature construction; supervised learning**

I. Motivation

Statistical relation learning (SRL) is a relatively young but growing field within machine learning which focuses on the problem of classification with networked data in various domains and settings (e.g., [1], [2], [3], [4], [5], [6]). Networked data consists of instances, generally of the same type such as web-pages or text documents, that are connected via various explicit relations such as one paper citing another, hyperlinks between web-pages, or people calling each other. Contrast this with traditional machine learning where instances to be classified are assumed to be *iid* (independent and identically distributed), an assumption that is critical for the underlying theory of traditional machine learning. Machine learning has a rich history with mature algorithms for inducing classifiers, many of which generally perform very well. Although most relational learning algorithms can be used on traditional machine learning data, they degenerate into standard machine learning algorithms because there are no relations between the instances to leverage. However, we note that many data sets are inherently relational even if their published form has been "cleaned" to fit into a non-relational format ready for standard machine learning or other analytics. Even for data that may not be inherently relational, one can certainly argue that two instances that are very similar may be "related" due to their similarity, whether the semantics of such truly hold. However, for relational learning and modeling, even such non-sensical relations may in fact improve classification performance if chosen carefully. Conversely, some relations–whether true or inferred–may not be at all relevant to a given classification problem whereas others may be very relevant. The problem is then to first identify or create candidate relations and then selecting the relations which are useful for a particular classification task. In other words, identifying (or creating) relevant relations is analogous to standard feature extraction and construction.

This work focuses on the question of whether it is possible to use the attributes of the instances to construct "meaningful" relations between instances in such a way as to have relational learning algorithms and relational inference not only be applicable but also be able to perform comparable to, or better than, existing mature (non-relational) machine learning algorithms. As mentioned above, constructing such relational features is in large part analogous to standard feature construction, except the potential power of this approach is that we can now leverage powerful relational inference techniques such as collective classification (see, e.g., [7]).

Recent work has shown a method for using an univariate graph-based relational learning algorithm (only labels and relations are used) with local attributes as well as explicit relations [8]. This work combined results from SRL and semi-supervised learning to produce a hybrid network consisting of explicit links and text-mined links. The work relied on an univariate algorithm for *within*-network classification: given a partially labeled network (some nodes have been labeled), label the rest of the nodes in the network.[1] The work showed how to combine multiple networks in such a way that relations from each network are weighted in a manner consistent with the amount of signal there is in the network. However, the paper failed to show how to use local attributes beyond using text-similarity for text-based instances. As such it failed in the promise of truly using local attributes–namely how to convert local attributes into relations that can be used by relational learners.

This paper focuses specifically on this short-coming and describes how to create relations from the attributes in traditional machine learning benchmark data sets such that relational learners and relational inference can be used and that they

[1]This analogous semi-supervised learning and transductive learning, but focused on networked data.

perform comparably to standard machine learning methods. This conversion to a network makes it possible to harness the power of relational learning and collective inference. We know that collective inference has the ability to greatly improve classification performance [7]. By providing a systematic and objective framework for creating relations based on similar attribute values, we now open the door for relational inference to be applied to traditional non-relational data sets.

The approach we take in this paper is generate relations based on each attribute and compute the signal in those relations using a metric known as *node-based assortativity* [5], a metric that measures the likelihood that two related instances share the same class. We then keep only the relations with high signal. Based on the approach taken in earlier work, we also generate relations based on instance similarity [9], [8]. Even though the approach taken in this paper is quite simple, the experimental results show that it works quite well, suggesting that more research in this area can yield ever more significant improvements.

We test our approach to attribute-based relationship creation on 31 data sets taken from the UCI repository [10], comparing the classification performance of an univariate relational learner on the created networks to that of standard machine learning on the original data. Our results show that this approach works quite well and that relational inference outperforms traditional learners on more data sets than any of the other learners and that, when compared against any one learner, relational inference wins the majority of the time. We further explore how much of the performance gain is due to collective inference and how this changes based on the size of the training examples. As this is similar to transductive learning, we also compare to transductive SVM on the binary classification problems, where we show that relational inference wins the majority of the time but is orders of magnitude faster than transductive SVM.

We next describe related work, followed by a description of our approach to the within-network classification task, how attribute-based relations are created and how we combine the different kinds of relations. We then describe our case study in which we test our approach, and conclude with a discussion of the results.

II. Related Work

The focus of this paper is on ways to enable SRL algorithms to be applicable to single-table data used in traditional machine learning.

Macskassy and Provost [5] investigated a simple univariate classifier, the weighted-vote relational neighbor (wvRN). They instantiated node priors simply by the marginal class frequency in the training data. The wvRN classifier performs relational classification via a weighted average of the estimated class membership scores ("probabilities") of the node's neighbors. Collective inference is performed via a relaxation labeling method similar to that used by Chakrabarti et al. [11]. We use this classifier in our case study.

Relational Bayesian Networks (RBNs, a.k.a. Probabilistic Relational Models [12], [1], [13] were applied in a within-network classification by Taskar et al. [13] to various domains, including a data set of published manuscripts linked by authors and citations. Loopy belief propagation [14] was used to perform the collective inference. The study showed that the PRM performed better than a non-relational naive Bayes classifier and that using both author and citation information in conjunction with the text of the paper worked better than using only author or citation information in conjunction with the text.

Techniques recently developed in the area of semi-supervised learning (e.g., [15], [16], [9], [4]) in a transductive setting (cf. [17]) are directly relevant to the work presented in this paper. Specifically, they consider data sets where labels are given for a subset of cases, and classifications are desired for a subset of the rest. They connect the data into a weighted network, by adding edges (in various ways) based on similarity between cases. In fact, prior work on combining explicit links and text-mined links [8] leveraged the work of Zhu et al. [9] and Wang and Zhang [4].

There has been a lot of work in creating similarity metrics or distance metrics between instances over the past three decades, more than we can cover here, and some of which we used in this work. These have been used in a variety of problem settings such as (relational) instance based learning (e.g., [18], [19]), nearest neighbor approaches (e.g., [20]), semi-supervised learning (see, e.g., [4]), and relational learning (e.g., [21]).

III. Classification in Networked Data

We use an existing and proven method for performing classification of networked data: the weighted-vote relational neighbor (wvRN) paired with relaxation labeling (RL) [22] for collective inference [5]. Using wvRN with an iterative label propagation such as relaxation labeling has been shown to perform better than other collective or exact inference methods [4], [5].

A. The weighted-vote Relational Classifier (wvRN)

The wvRN classifier estimates class-membership probabilities based on two assumptions: (1) that the label of a node depends only on its immediate neighbors, and (2) the entities in the graph exhibit homophily—i.e., linked entities have a propensity to belong to the same class (cf. [23]). This homophily-based model is motivated by observations and theories of social networks [23], where homophily is ubiquitous.

Definition. Given $v_i \in V^U$, wvRN estimates $P(x_i|\mathcal{N}_i)$ as the (weighted) mean of the class-membership probabilities of the entities in $\mathcal{N}_i$:

$$P(x_i = X|\mathcal{N}_i) = \frac{1}{Z} \sum_{v_j \in \mathcal{N}_i} w_{i,j} \cdot P(x_j = X|\mathcal{N}_j),$$

where V^U is the set of unlabeled vertices in graph G, $\mathcal{N}_i$ is the set of neighbors for node v_i, and Z is the usual normalizer.

As this is a recursive definition (for undirected graphs, $v_j \in \mathcal{N}_i \Leftrightarrow v_i \in \mathcal{N}_j$) the classifier uses the "current" estimate for $P(x_j = X|\mathcal{N}_j)$.

B. Relaxation Labeling (RL)

For the collective inference part of our study, we use relaxation labeling (RL) as described in Macskassy and Provost [5]. Rather than treat graph G as being in a specific labeling "state" at every point (e.g., as a Gibbs sampler does), relaxation labeling retains the uncertainty, keeping track of the current probability estimates for $v_i \in V^U$. RL then "freezes" the current estimations so that at step $t+1$, all vertices will be updated based on the estimations from step t. Following prior work, we employ a simulated annealing approach to ensure convergence—each subsequent iteration gives more weight to a node's own current estimate and less to the influence of its neighbors.

More formally, RL inference with wvRN is defined as:

$$\mathbf{c}_i^{(t+1)} = \beta^{(t+1)} \cdot \text{wvRN}(\mathbf{C}^{(t)}) + (1-\beta^{(t+1)}) \cdot \mathbf{c}_i^{(t)},$$

where $\mathbf{c}_i^{(t)}$ is a vector of probabilities (probability distribution) which represents an estimate of $P(x_i|N_i)$ at time step t and wvRN($\mathbf{C}^{(t)}$) denotes applying wvRN using all the estimates from time step t. We define the simulated annealing constants as:

$$\beta^0 = k, \qquad \beta^{(t+1)} = \beta^{(t)} \cdot \alpha,$$

where k is a constant between 0 and 1. Following prior work ([5]), we set $k = 1.0$, and $\alpha = 0.99$ is a decay constant, which we set to 0.99.

IV. Creating Attribute-based Relations

First, we introduce some notation. An instance, x_i, is represented as $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i = \{x_{i,1}, \ldots, x_{i,n}\}$ represent the attribute values for instance x_i, and $y_i \in \mathcal{C}$ is its label. Each attribute value, $x_{i,k}$, belongs to an attribute class $\mathcal{X}_k$, where $\mathcal{X}_k$ is either a fixed categorical set (e.g., $\mathcal{X}_k = \{A, B, C\}$), an ordinal set (e.g., $\mathcal{X}_k = \{1, 2, 3\}$) or continuous (e.g., $\mathcal{X}_k = \Re$). We abstract these possibilities into two broad categories: categorical or numeric values. We note that this categorization is how machine learning generally handles attributes. In fact, many machine learning algorithms convert categorical into indicator variables and treat the indicator variables as numeric.

We represent the created network as a graph $G = (V, E)$, where V is the set of vertices in the graph (one vertex per instance) and E is the set of edges in the graph. The relations we create will form these undirected weighted edges. We denote different relations by different edge types. The weight of an edge between x_i and x_j based on attribute $\mathcal{X}_k$ is defined as $w_k(x_i, x_i)$.

The approach we take in this paper is to create one relationship per attribute and one relationship based on instance similarity (analogous to what has been done before). Therefore, an instance which is made up of n attributes will have $n+1$ candidate relations. These are weighted and pruned by how informative they are as described in Section V.

A. Creating relations from categorical attributes

For a categorical attribute, $\mathcal{X}_k$, the instance attribute value can only be one of a fixed set of values. If two instances have the same observed value for attribute $\mathcal{X}_k$, then we create an edge between them. The strength of the relationship is either 1 or no relationship. In other words:

$$w_k(x_i, x_j) = \begin{cases} 1 & \text{if} x_{i,k} = x_{j,k} \\ 0 & \text{otherwise} \end{cases}$$

We treat each possible value of the categorical attribute $\mathcal{X}_k$ as a separate candidate relation.

B. Creating relations from numerical attributes

We create an edge between two instances based on how close their two observed values are. We normalize the strength of relations to lie between 0 and 1 using the following equation:

$$w_k(x_i, x_j) = 1 - \min\left(1, \frac{|x_{i,k} - x_{j,k}|}{\max_k - \min_k}\right),$$

where $\min_k$ and $\max_k$ are the minimum and maximum observed values in the training set. We put an upper limit on their differences to ensure that the weight does not become negative (for test instances). We consider both the raw difference as well as the normalized difference when creating candidate relations.

C. Creating relations using instance similarity

The final type of relationship we consider in this paper is that of instance-based similarity (see [18]). We create a link between two instances with a weight that is the inverse of the euclidean distance between the two instances. Specifically:

$$w_*(x_i, x_j) = \left(1 + \sqrt{\sum_k \text{dist}^2(x_{i,k}, x_{j,k})/n}\right)^{-1},$$

where n is the number of attributes and the distance function is one of the functions above based on the attribute type.

Based on prior work, we limit the number of similarity based edges to 5 for each node [4], [8]. In other words, for instance x_i, we add an edge to its 5 most similar nodes.[2]

V. Selecting Relations

Prior work has addressed how one should go about combining multiple networks such as those created by the different attribute-based relations we have just described [8]. Specifically, they used a variant of the *assortativity coefficient* [24]–a metric to measure the amount of homophily in a network–called the *node-based* assortativity metric. We adopt the same approach here.

The node-based assortativity score uses the correlation between the classes linked by edges in a graph. Specifically, it is based on the graph's node-based *assortativity matrix*—a CxC matrix, where cell e_{ij} represents, for (all) nodes of class c_i, the average fraction of their weighted links that link them

[2] x_i may be in the top-5 of another instance and its actual degree can therefore be more than 5.

Name	Size	Attributes Nom.	Attributes Num.	Number Classes
annealing	898	32	6	5
autos.imports-85	203	10	15	2
balance-scale	625	0	4	3
breast-cancer	699	0	9	2
breast-cancer (wdbc)	569	0	30	2
cmc	1473	7	2	3
credit-screening	690	9	6	2
cylinder-bands	540	19	20	2
dermatology	366	0	34	6
echocardiogram	131	0	11	2
ecoli	336	0	7	8
glass	214	0	9	2
haberman	306	0	3	2
heart-disease (cleveland)	303	0	13	2
heart-disease (hungarian)	294	0	13	2
hepatitis	155	13	6	2
horse-colic	368	16	7	2
ionosphere	351	0	34	2
iris	150	0	4	3
liver-disorders	345	0	6	2
musk (clean1)	476	0	166	2
pima-indians-diabetes	768	0	8	2
sonar	208	0	60	2
tae	151	4	1	3
thyroid (new-thyroid)	215	0	5	3
thyroid (sick-euthyroid)	3163	18	7	2
vehicle	846	0	18	4
water-treatment	527	0	38	3
wine	178	0	13	3
yeast	1484	0	8	10
zoo	101	15	1	7

TABLE I
CHARACTERISTICS OF THE 31 DATA SETS USED IN THIS STUDY.

to nodes of class c_j, such that $\sum_{ij} e_{ij} = 1$. The node-based assortativity coefficient, A_E, is then calculated as follows:

$$A_E = \frac{\sum_i e_{ii} - \sum_i a_i \cdot b_i}{1 - \sum_i a_i \cdot b_i},$$

where $a_i = \sum_j e_{ij}$ and $b_j = \sum_i e_{ij}$.

We then reweight edges for a node such that the edge weights for edge type E sums to A_E.

The advantage of this approach is that it is very general and can easily be used with an arbitrary number of edge types, each having their own semantics of edge-weights and edge statistics.

VI. STUDY

The main thesis of this paper is that by converting a standard non-relational machine learning data set into a relational data set, we can apply relational inference and gain the inherent advantages of relational learning and collective classification. This case study will empirically test this thesis using the wvRN classification method (with and without collective classification) and compare it against standard machine learning and transductive learning classifiers.

A. Data

We use 31 benchmark data sets from the UCI machine learning repository [10]. The data sets we use are listed in Table I.

B. Experimental Methodology

In order to combine the newly created edges we must compute A_E. Since this computation requires labels in order to compute the individual e_{ij} cells in the assortativity matrix, we use only nodes for which the label is known. Second, as A_E tends towards 0, the signal in that type of edge disappears. We therefore remove any edge if their $A_E < 0.05$.

We use the NetKit toolkit [5][3] to run all our experiments as it has a framework to make it easy to use the exact same experimental environment across all methods. For traditional machine learners, we leveraged NetKit's capability to use Weka [25] classifiers.

We compared wvRN to 5 off-the-shelf standard machine learning algorithms: j48 (Weka's implementation of C4.5), k-nearest neighbor with $k = 5$[4], logistic regression, naive Bayes, and smo (a linear svm). These classifiers are available in Weka and were all run within the NetKit to keep as much of the environment identical as possible. We note here that we are using vanilla classifiers and do not try to optimize for parameters in any classifier, and not for the relation creation either. Clearly all methods can perform better with learning of the hyper-parameters, but we here explore how well these methods work against each other in general.

Because the within-network classification is analogous to semi-supervised learning and transductive learning for non-relational data, we also compare to the transductive SVM classifier which is available in SVM^{light}.[5] We ran it with a linear kernel.

We use accuracy as the measure of performance, where accuracy is based on 10-fold stratified cross-validation.

C. Results

The five standard supervised learners, the transductive SVM, and wvRN (without collective inference) were all run on the data and the best learner on a data set (best average accuracy over 10 runs) was compared against the next-best learner using a paired t-test to see if the accuracy difference was significant at the $p = 0.05$ level. The results are shown in Table II.

We see that wvRN had the best accuracy on 12 of the 31 data sets (5 of which were significant wins), which means it won more than any other learner. The next-best classifier was logistic regression which won 8 times, but only one of those wins were significant. In addition, we see that when compared one-on-one against any of the other classifiers, it wins the majority of the time. This validates the main thesis of the paper that generating relations and using relational learning can in fact result in significant classification performance, often beating other classifiers. We also see that when it does not win it often performs quite poorly, suggesting that there may be some underlying data characteristics where relational inference will work and others where it fails. We did explore whether there were some indicators based on the type of relations

[3]Available at: `http://netkit-srl.sourceforge.net`
[4]k was set following prior work [8].
[5]Available at `http://svmlight.joachims.org`

		Classifier						
Data Set	**prior**	**j48**	**knn5**	**logistic regression**	**naïve Bayes**	**smo**	**tsvm**	**wvRN**
anneal	0.762	0.924	**0.937**	NA	0.718	NA		0.777
autos.imports-85	0.562	0.897	0.895	0.814	0.884	0.889	0.895	**0.900**
balance-scale	0.461	0.767	0.879	0.896	**0.907**	0.877		0.706
breast-cancer	0.655	0.951	0.964	**0.967**	0.959	0.965	0.939	0.896
breast-cancer (wdbc)	0.627	0.952	0.967	0.949	0.926	**0.972**	0.919	0.902
cmc	0.427	0.514	0.477	0.505	0.516	0.499		**0.677***
credit-screening	0.555	0.862	**0.876**	0.846	0.796	0.856	0.838	0.868
cylinder-bands	0.578	0.698	0.713	0.735	0.789	**0.817**	0.779	0.770
dermatology	0.306	0.964	0.958	0.964	**0.972**	0.967		0.630
echocardiogram	0.672	0.908	0.875	0.885	0.850	0.892	0.833	**0.954**
ecoli	0.426	0.852	0.885	0.854	0.882	0.821		**0.976***
glass	0.762	0.935	0.955	0.931	0.935	0.945	0.900	**0.995**
haberman	0.735	0.709	0.683	0.742	0.753	0.783	0.723	**0.784**
heart-disease (cleveland)	0.541	0.789	0.786	**0.838**	0.807	0.828	0.800	0.730
heart-disease (hungarian)	0.639	0.772	0.639	**0.857**	0.829	0.821	0.804	0.824
hepatitis	0.794	0.760	0.850	0.843	0.836	**0.864**	0.857	0.807
horse-colic	0.663	0.676	0.653	0.739	0.681	0.744	**0.817***	0.673
ionosphere	0.641	**0.909***	0.859	0.877	0.829	0.879	0.832	0.777
iris	0.333	0.971	0.971	0.986	0.964	0.971		**0.993**
liver-disorders	0.580	0.687	0.591	0.681	0.576	0.545	0.658	**0.844***
musk (clean1)	0.565	0.840	0.836	0.859	NA	0.838	0.816	**0.992***
pima-indians-diabetes	0.651	0.730	0.751	**0.780**	0.766	0.763	0.751	0.738
sonar	0.534	0.750	0.795	0.713	0.695	0.775	0.760	**0.995***
tae	0.344	0.540	0.393	**0.573**	0.543	0.536		0.527
thyroid (new-thyroid)	0.698	0.948	0.950	0.963	0.980	0.895		**0.995**
thyroid (sick-euthyroid)	0.907	**0.981**	0.925	0.957	0.839	0.910	0.927	0.964
vehicle	0.258	0.726	0.720	**0.799***	0.468	0.736		0.551
water-treatment	0.861	0.881	0.900	0.894	0.867	**0.921***		0.892
wine	0.399	0.944	0.953	0.949	0.965	0.988		**0.989**
yeast	0.312	0.538	0.569	**0.594**	0.573	0.559		0.549
zoo	0.406	0.920	0.900	**0.970**	0.922	0.911		0.800
# Wins (bold)		2	2	8	2	4	1	12
# wins vs. wvRN (out of 31)		12	14	15	14	15	8/18	

TABLE II

AVERAGE ACCURACIES OF THE SEVEN LEARNERS USING 10-FOLD CROSS VALIDATION. THREE CELLS ARE LISTED AS NA BECAUSE WEKA DID NOT RETURN A CLASSIFIER FOR THOSE DATA. TRANSDUCTIVE SVM (TSVM) ONLY HAS RESULTS FOR DATA SETS WITH A BINARY CLASSIFICATION PROBLEM. CLASS PRIOR IS SHOWN IN COLUMN 2 FOR REFERENCE. THE LAST TWO ROWS SHOW HOW OFTEN EACH LEARNER WON (HAD THE BEST PERFORMANCE) AND WON WHEN COMPARED 1-ON-1 VS. WVRN. BEST PERFORMERS FOR EACH DATA SETS IS SHOWN IN BOLD. PERFORMANCE WHICH WAS SIGNIFICANTLY THE BETTER THAN OTHER METHODS (AT $p \leq 0.05$)), AS MARKED BY A $*$.

	Classifier (Each cell shows (#overall wins/#wins against wvRN-RL)							
(# Wins / vs wvRN-RL)	**j48**	**knn5**	**logistic regression**	**naïve Bayes**	**smo**	**tsvm**	**wvRN**	**wvRN-RL**
10% training	3/25	4/24	9/27	5/26	7/26	3/10	1	2
30% training	3/20	3/17	6/18	5/20	8/21	1/6	4	7
50% training	2/14	1/14	4/14	5/14	6/16	1/5	7	13
70% training	2/12	3/12	5/15	4/14	5/17	1/6	9	12
90% training	2/12	2/14	8/14	2/12	4/15	1/8	12	10

TABLE III

NUMBER OF TIMES EACH OF THE SEVEN LEARNERS WON (BEST AVERAGE ACCURACY OVER 10 RUNS), AS WE INCREASE THE TRAINING SET SIZE FROM 10% TO 90%. EACH CELL SHOWS THE NUMBER OF TIMES (OUT OF 31; OR OUT OF 18 FOR TSVM) THAT THE LEARNER HAD THE BEST AVERAGE ACCURACY, AND HOW MANY TIMES IT WAS BETTER THAN WVRN-RL.

extracted or their assortativity score. Unfortunately, we were not able to identify any clear signal, but it is a question for further study.

One key technique in relational learning is its use of collective inference to find the optimal joint labeling of all the test labels and we next explore whether the power of collective inference does translate to these constructed relational data.

We added wvRN-RL (the collective inference version of wvRN) to the set of learners tested and varied the amount of training data from 90% down to 10% of the total data set. We tracked, for each run, how often each of the learners had the best performance similar to the first experiment as well as how often the standard classifiers won over wvRN-RL. Table III shows the summarized result of these runs.

The results are quite enlightening and highlight interesting problems that need to be explored better. First, we observe that logistic regression was generally a strong performance as was smo (linear SVM that comes with Weka). Unsurprisingly, we observe that the non-collective version of wvRN is not competitive at all when only a few training labels are available

and that it is only competitive towards the end when 90% of the data set is labeled. However, the collective version of wvRN is clearly the better classifier when training $\geq 50\%$. We see that the two versions of wvRN clearly dominate when 90% of the data is labeled. One very interesting observation is that wvRN-RL performs quite poorly when only 10% of the data is labeled, and while it has as many wins as other classifiers at 30%, it clearly does not perform as well when head-to-head against any of the other classifiers. The reason for this is because the relational data itself is not well constructed. Specifically, the relations are selected and weighted based on their respective assortativity values, which in turn are computed based on the available training set. However, the efficacy of that metric is questionable if there is not enough data, as noted by Macskassy and Provost [5]. In other words, the problem is not with relational inference but with the construction of the relations. Again, this suggests that more work is needed to understand how to set appropriate parameters for the relation creation and weighting, as well as when we might expect relational inference to be useful.

VII. Discussion and Limitations

The thesis of this paper was that it was possible to create relations among instances in a non-relational data set such that relational inference would perform better than standard classificiation. If this thesis was true, then we would increase the number of learners we could bring to bear on general machine learning problems.

We described a simple method of creating relational data by creating relations for each attribute using a specific attribute-type similarity function and described a principled way for selecting and weighting these constructed relations.

We empirically tested our thesis by applying a relational classifier on 31 data sets from the UCI machine learning repository. Our results supported our thesis, showing that the relational classifier performed quite well, being the best performer more than any other classifier and winning the majority of the time when competing against any single other classifier.

We also explored the contribution of collective inference. We varied the amount of training data from 10% to 90% of the data set to test how well collective inference performed as a function of sparsity of labels. The results showed a very strong classification performance using collective inference, but not with the relational classifier itself. In fact, we observed that the relational learning performance was poor when labels were very sparse, but that it dominated once we have more then 30% of the instances labeled. Our relation construction method relies heavily on having enough data to weight and construct relations, and we believe that the poor performance was due to poor relation construction. This follows observations made in prior work [5].

References

[1] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning Probabilistic Relational Models," in *Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 1999.

[2] C. Cortes, D. Pregibon, and C. T. Volinsky, "Communities of Interest," in *Proceedings of Intelligent Data Analysis (IDA)*, 2001.

[3] A. Blum, J. Lafferty, R. Reddy, and M. R. Rwebangira, "Semi-supervised learning using randomized mincuts," in *Proceedings of the 21st International Conference on Machine Learning (ICML)*, 2004.

[4] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," in *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, 2006, pp. 985–992.

[5] S. A. Macskassy and F. Provost, "Classification in Networked Data: A toolkit and a univariate case study," *Journal of Machine Learning Research (JMLR)*, vol. 8, no. May, pp. 935–983, 2007.

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

[7] D. Jensen, J. Neville, and B. Gallagher, "Why Collective Inference Improves Relational Classification." in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[8] S. A. Macskassy, "Improving learning in networked data by combining explicit and mined links," in *Proceedings of the Twenty-Second Conference on Artificial Intelligence*, 2007.

[9] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," in *Proceedings of the 12th International Conference on Machine Learning*, 2003.

[10] C. L. Blake and C. J. Merz, "UCI Repository of machine learning databases," Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[11] S. Chakrabarti, B. Dom, and P. Indyk, "Enhanced Hypertext Categorization Using Hyperlinks," in *ACM SIGMOD International Conference on Management of Data*, 1998.

[12] D. Koller and A. Pfeffer, "Probabilistic frame-based systems," in *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI)*, 1998, pp. 580–587.

[13] B. Taskar, E. Segal, and D. Koller, "Probabilistic Classification and Clustering in Relational Data," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)*, 2001, pp. 870–878.

[14] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[15] A. Blum and S. Chawla, "Learning from Labeled and Unlabeled Data using Graph Mincuts," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2001, pp. 19–26.

[16] T. Joachims, "Transductive Learning via Spectral Graph Partitioning," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2003.

[17] V. N. Vapnik, *Statistical Learning Theory*. John Wiley, NY, 1998.

[18] D. W. Aha and D. Kibler, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, 1991.

[19] T. Horvath, S. Wrobel, and U. Bohnebeck, "Relational instance-based learning with lists and terms," *Machine Learning*, vol. 43, no. 1/2, pp. 53–80, 2001.

[20] T. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[21] L. D. Raedt, *Logical and Relational Learning*. Springer, 2008.

[22] A. Rosenfeld, R. Hummel, and S. Zucker, "Scene labeling by relaxation operations," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 6, pp. 420–433, 1976.

[23] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a Feather: Homophily in Social Networks," *Annual Review of Sociology*, vol. 27, pp. 415–444, 2001.

[24] M. E. J. Newman, "Mixing patterns in networks," *Physical Review E*, vol. 67, 2003, 026126.

[25] I. H. Witten and E. Frank, in *Data Mining: Practical machine learning tools with Java implementations*. San Francisco: Morgan Kaufmann, 2000.