Extended Discriminative Random Walk: A Hypergraph Approach To Multi-View Multi-Relational Transductive Learning

Sai Nageswar Satchidanand   Harini Ananthapadmanaban   Balaraman Ravindran
Indian Institute of Technology Madras, Chennai, India
sainageswar@gmail.com, harini.nsa@gmail.com, ravi@cse.iitm.ac.in

Abstract

Transductive inference on graphs has been garnering increasing attention due to the connected nature of many real-life data sources, such as online social media and biological data (protein-protein interaction network, gene networks, etc.). Typically relational information in the data is encoded as edges in a graph but often it is important to model multi-way interactions, such as in collaboration networks and reaction networks. In this work we model multi-way relations as hypergraphs and extend the discriminative random walk (DRW) framework, originally proposed for transductive inference on single graphs, to the case of multiple hypergraphs. We use the extended DRW framework for inference on multi-view, multi-relational data in a natural way, by representing attribute descriptions of the data also as hypergraphs. We further exploit the structure of hypergraphs to modify the random walk operator to take into account class imbalance in the data. This work is among very few approaches to explicitly address class imbalance in the in-network classification setting, using random walks. We compare our approach to methods proposed for inference on hypergraphs, and to methods proposed for multi-view data and show that empirically we achieve better performance. We also compare to methods specifically tailored for class-imbalanced data and show that our approach achieves comparable performance even on non-network data.

1 Introduction

With the advent of technology for easy generation and storage, data sources have become increasingly rich in detail. Depending on the nature of the data this poses several challenges to machine learning and consequently several classes of solutions have emerged. Our goal in this work is to bring together different strands of ideas to develop an unified framework for various transductive learning problems on partially labeled networked data. Due to the connected nature of many real-life data sources, the problem of within network classification has become an active area of research in recent times [Zhu and B.Goldberg, 2009]. In this setting, the data instances are treated as nodes in a graph and the links represent relations between the nodes. Given a small labeled set, the goal is to infer labels for the other nodes in the graph. This is an instance of transductive inference, since the labeled and unlabeled data together with the graph structure is used in the inference procedure [Chakrabarti *et al.*, 1998; Castillo *et al.*, 2007; Domingos and Richardson, 2001].

Many of the learning approaches assume a pair-wise relation between the nodes which translate to edges in the graph. In this work we are interested in looking at data that have multi-way relations between the instances. For e.g., the *co-author* relation is naturally a multi-way relation. Such multi-way relations can be modeled using *hypergraphs* in which the edges are subsets of nodes. The use of hypergraphs enables several extensions to the basic within network classification setting and such extensions are the key contributions of this work.

Hypergraph based modeling for machine learning has garnered some interest recently [Yu *et al.*, 2012; Gao *et al.*, 2011; Sun *et al.*, 2008]. In particular, Zhou and Schölkopf in 2006 [Zhou *et al.*, 2006] extended spectral clustering methods for graphs to hypergraphs and further developed a transductive inference setup for embedding, i.e., labelling a partially labeled hypergraph. In this approach the hyperedge which is being cut is considered as a clique with weight of the hyperedge being distributed uniformly over all sub-edges of clique. The spectral formulation then tries to minimize the total number of these sub-edges across the cut, using a normalized hypergraph cut objective that penalises unbalanced cuts.

In the case of many sources of connected data, such as online social networks and biological networks, in addition to the relational structure there is rich attribute information as well. This has lead to the development of *collective learning and inference* approaches that work with such attributed graphs [Desrosiers and Karypis, 2009; Sen *et al.*, 2008]. Collective Classification approaches like Iterative Classification Algorithm (ICA) [Sen *et al.*, 2008] use an augmented description of the data where the class-distribution in the neighbourhood of a node are treated as additional features. These work well in situations where there is sufficient labeled data to train a classifier well. Such methods can also be generalized beyond a transductive setting, but that is not of relevance to this work.

Another source of richness in data is the availability of multiple-descriptions of the same data. For example, to classify videos in YouTube we can construct multiple views, such as attributes from video, attributes from speech/sound in video, text corpus from text description of video etc., and several methods have been proposed to take advantage of the same [Xu *et al.*, 2013; Sun, 2013]. Multi-view methods have been used extensively in a semi-supervised setting with partially labeled training data [Blum and Mitchell, 1998; Sindhwani *et al.*, 2005]. However, handling multi-view data for transductive inference on graphs has not received much attention and there are only a few results such as [Zhou and Burges, 2007; Shi *et al.*, 2012; Vijayan *et al.*, 2014]. Similarly, the same entities could have different kinds of relations between themselves. “Follows” and “retweets” on Twitter is an example of multiple relations.

One over-arching problem that spans the different settings described above and in general inductive learning from data, is that of class imbalance. In many real settings, the different classes are seldom distributed uniformly. There have been different approaches proposed for handling class imbalance (e.g. [Cieslak *et al.*, 2012]) but there are none that are satisfactory in the networked data context.

In this work we propose a unified method to address the problems discussed above by extending the discriminative random walk framework (DRW) [Callut *et al.*, 2008; Mantrach *et al.*, 2011]. DRW is one of the most successful approaches for within network classification and is based on transit times in a limited random walk. The method works very well even when the fraction of labeled nodes on the graph is very small. In this work we extend the DRW framework in several significant ways.

- First, we extend the DRW framework to accommodate inference on hypergraphs. We introduce a new random walk operator on hypergraphs and modify the DRW procedure appropriately.
- Second, we modify the random walk operator to handle multiple relations and multiple views. This is accomplished by modeling the attribute descriptions of the data as a hypergraph.
- Third, we account for class imbalance in the network data to a limited extent by appropriately reweighting the hyperedges with a preponderance of minority class points. Such a re-weighting is made possible naturally by the use of a hypergraph based representation.

This *extended DRW* yields a single algorithm that can handle multi-way multi-relational multi-attribute data with class imbalance. We extensively compare the proposed extended DRW approach with a variety of existing algorithms on synthetic as well as many real data sets and empirically establish that our approach does better than existing methods, especially in the presence of class skew and limited availability of labeled data. For example, even when only 10% of the nodes in the graph are labeled we are able to achieve up to 35% improvement when the class ratios are very skewed.

2 Background

In this section we will look into basic formulations involving hypergraphs. We will define formal representations for hypergraphs and D-Random Walk.

Hypergraphs

Let $G = (V, E)$ be a hypergraph, where V represents a finite set of objects and E the set of hyperedges such that for any $e_i \in E, e_i \subseteq V$. Each edge is associated with a weight $w(e)$. For a vertex v, degree of vertex $d(v) = \sum_{e \in E \& v \in e} w(e)$. For a hyperedge $e \in E$, $\delta(e)$ represents the degree of the edge i.e. $\delta(e) = |e|$. Let H be a hypergraph incidence matrix with $h(v, e) = 1$ if vertex v is in edge e. Let W denote the diagonal weight matrix containing weights of the hyperedges, D_v denote the diagonal vertex degree matrix containing the degrees of vertices and D_e denote the diagonal edge degree matrix containing the degrees of edges. Also, let $n = |V|$ be the total number of instances.

For an attribute view, let X be $n \times d$ categorical attribute matrix of instances where x_i represents an attribute vector in the dataset, i.e., a column containing the values of this attribute for all elements of the dataset. Let L be a set of labeled instances, assigned to a category from a discrete set Y. The label of each instance $v \in L$ is written as y_v and L_y denotes the set of nodes in class y, with $n_y = |L_y|$.

D-Walks

As proposed in [Callut *et al.*, 2008], bounded random D-Walks are a very effective way of classification in a partially labeled graph. For a given set of states $v_0, v_1, ..., v_N$ and a class $y \in Y$, a D-Walk is a sequence of states $v_0, v_1, ..., v_l$, such that $y_{v_0} = y_{v_l} = y$ and $y_{v_t} \neq y$ for all $0 < t < l$. Let D_l^y denote the event of a D-walk of exactly length l starting and ending on a node labeled y. For a given unlabeled node $v \in V$, we define $E[pt(v) \mid D_l^y]$, the expected length-limited passage time ($pt(v)$), as the number of times the random walk process reaches node v in a walk of length exactly l as follows:

$$E[pt(v)|D_l^y] = \sum_{t=1}^{l-1} P[X_t = v|D_l^y] = \sum_{t=1}^{l-1} \frac{P[X_t = v \wedge D_l^y]}{P[D_l^y]} \tag{1}$$

Now, the D-walk betweenness function for a node v and class y and some maximum walk length L is defined as:

$$B_L(v, y) = \sum_{l=1}^{L} E[pt(v)|D_l^y] \tag{2}$$

The above betweenness can be computed very efficiently using forward and backward variables, as explained in [Callut *et al.*, 2008]. An unknown node is classified based on its betweenness in the walk.

$$P[v \mid y] = \frac{B_L(v, y)}{\sum\limits_{y' \in Y} B_L(v, y')} \tag{3}$$

$$y_v = argmax_{y \in Y} P[v|y]P[y] \tag{4}$$

where $P[y]$ is estimated as the proportion of nodes belonging to class y.

3 Extended DRW

The DRW algorithm as proposed in [Callut *et al.*, 2008] was defined on simple graphs, and used an edge weighting scheme to accommodate instance similarity based on attributes. We extend their algorithm in several ways:
(1) We define DRW for hypergraphs to model multi-way relations by defining a random walk operator appropriate for both graphs and hypergraphs;
(2) We accommodate multiple relations in the DRW framework by extending the random walk operator to multiple hypergraphs;
(3) We include attribute information also as a hypergraph; and
(4) We introduce an edge reweighting scheme to handle class-imbalance.
In this section, we explain the extensions and modifications that were carried out on DRW and propose our algorithm.

Random Walk in hypergraphs

The random walk probability matrix P for a hypergraph $G = (V, E)$ can be computed in the following way. Let $u \in E, v \in E$ be two vertices in a hypergraph connected by a hyperedge e' then the transition rules are defined as :

- Choose e' from vertex u over all edges incident on u with a probability proportional to the weight of e', i.e.,
$$p_1 = \frac{w(e')}{\sum\limits_{e \in E \& u \in e} w(e)} \tag{5}$$
- Choose a vertex v (excluding u) in e' uniformly at random, i.e.,
$$p_2 = \frac{1}{\delta(e') - 1} \tag{6}$$
- Probability of transition from u to v is summation over all such hyperedges that connect u and v,
$$p(u, v) = \sum_{\forall e \in E, \{u,v\} \subset e} (p_1 * p_2) \tag{7}$$

As $\left[\sum\limits_{e \in E \& v \in e} w(e) \right]$ is $d(v)$, we can the write the transition stochastic matrix P as,

$$P = D_v^{-1} H W (D_e - I)^{-1} H^T \tag{8}$$

The diagonal entries of P are ensured to be 0, to preclude self-loops. The above stochastic matrix P is row-stochastic and a non-symmetric matrix. In our framework, we do not need a symmetric matrix and hence do not use techniques used in graph learning literature (e.g., [Shi and Malik, 2000]) to approximate the nearest symmetric matrix for random walk. This preserves the accuracy of the random walk derived from the transition probability matrix. Our formulation for random walk is also different from the random walk operator defined in [Zhou *et al.*, 2006] - for a given edge e we choose a vertex v uniformly at random excluding the current vertex, whereas, in [Zhou *et al.*, 2006] for a given edge e a vertex v is chosen uniformly at random including the current vertex. Their formulation reduces the probability of transitioning between nodes by a significant amount for hyperedges with a small degree. Crucially, this modification brings the random walk operator more in line with the definition on graphs and it can be used generically for both hypergraphs and graphs.

For this random walk operator, the stationary or steady state probability of a node j is given by

$$\pi_j = \frac{d(j)}{\sum\limits_{v \in V} d(v)} \tag{9}$$

Based on this definition of hypergraph random walk, DRW can be extended for hypergraphs by replacing the probabilities in the forward and backward operator computation, by the transition probabilities computed for the hypergraph. There would be a corresponding change in the betweenness measures, with the expected path lengths now being dependent on the hypergraph.

Random Walk for Multiple Relations

We extend the random walk operator proposed in the previous section, and propose a multi-graph random walk operator that can be used for learning with multiple relations, along the lines of [Zhou and Burges, 2007]. We model multiple relations as a set of graphs and hypergraphs, which can be viewed as different layers. A random walker can move from one graph (or layer) to another with the probability distribution defined over the different layers being dependent on the steady state probabilities of the nodes, which in turn is influenced by the differential weights α that are used to weight the different graphs and hypergraphs. Once the layer is chosen, the random walker can move following the transition probability matrix defined in the earlier section.

The following equations outline the procedure for combining multiple graphs and hypergraphs. Let $P_1, P_2, ..., P_n$ be the transition probability matrices for each graph and hypergraph and $\Pi_1, \Pi_2, ..., \Pi_n$ be the corresponding steady state probability diagonal matrices.

$$\beta_{ij} = \frac{\alpha_i \Pi_{ij}}{\sum\limits_k \alpha_k \Pi_{kj}} \tag{10}$$

$$\forall\, 0 \leq j \leq |V|\,, 0 \leq \alpha_i \leq 1, 0 \leq i < n,$$

$$\mathbb{P} \leftarrow \sum_i \beta_i P_i \tag{11}$$

$$\mathbf{\Pi} \leftarrow \sum_i \alpha_i \Pi_i \tag{12}$$

Extending DRW to multiple relations is straightforward once the combined random walk operator is defined, and can be done by replacing the probabilities in the forward and backward operator calculations by the transition probabilities given in equation 11. The betweenness score that's thus obtained signifies the node's betweenness for a particular class over all the different layers of graphs and hypergraphs and over all possible bounded D-random walks through those layers, for a given L.

Modeling Attribute Views

Categorical attributes can be converted to hypergraphs by connecting all instances having the same value for a particular attribute, with a single hyperedge. Let $x_1, ..., x_p$ be the attributes of the data. Let each x_i draw its values from the set ν_i where $\nu_i = \{\nu_i^1, \nu_i^2, ..., \nu_i^{n_i}\}$. We then construct a hyperedge for each ν_i^n. Real attributes can be discretized into multiple bins, by appropriately recursively splitting each attribute. In order to produce *purer* edges to aid in the inference process even under class skew (see next section), we used the split criterion from Hellinger distance decision trees [Cieslak *et al.*, 2012]. Each of the bins is considered to be a hyperedge.

Handling Class Imbalance

The fact that we are using hyperedges in our modeling gives us a significant advantage in exploring new ways to differentiate informative connections from the noisy ones. The random walk operator over hypergraphs gives us the freedom to weight each of the hyperedges individually, and we can define weights to be some function of the set of nodes comprising an edge. One weighting measure that could be used is the purity of the hyperedge, i.e., the fraction of the class in majority among the known labels on that edge. Crucially, to address skewness in the dataset, we use Hellinger Distance [Cieslak *et al.*, 2012] to generate the weights. Skewness in the dataset implies that instances of some classes are over represented while instances of some other classes are under-represented. In this scenario, the instances of the under-represented class are dominated by the majority class in most of the hyperedges in the graph. From the point of computing the betweenness score for the minority class, hyperedges with a slightly more than expected number of known instances of the minority class are strong indicators of the similarity between the nodes in that hyperedge. We need a weighing function that captures this similarity between minority class points. Hellinger distance addresses these requirements very well, being less sensitive to skewness in data.

For binary classification, Hellinger distance of a hyperedge is defined as follows:

$$W(e) = \left(\sqrt{\frac{|y_e^+|}{|y^+|}} - \sqrt{\frac{|y_e^-|}{|y^-|}} \right)^2 \tag{13}$$

where,

y_e^+ = set of positive instances in hyperedge e,
y_e^- = set of negative instances in hyperedge e,
y^+ = set of positive instances in entire training set,
y^- = set of negative instances in entire training set.

This weight becomes 1 when all of the instances in the hyperedge have the same label and the hyperedge covers all the known instances of that label.

For a multi-class problem, we define the Hellinger distance as follows :

- For each edge, find the edge-class ratios that is defined as the ratio of the number of known instances of each class in the edge to the number of instances of each class in the whole training data.
- Consider the class having maximum edge-class ratio as the positive class.
- Consider all other classes together as the negative class.
- Compute Hellinger distance as per Equation 13

Given that the class proportions are skewed, if a class has a high edge-class ratio, then it indicates a significant deviation from the underlying class distribution and therefore it is probably relevant to the inference. Considering the class with the maximum edge-class ratio as positive, instead of the most frequently occurring class was also empirically observed to perform better.

If none of the instances in a hyperedge have known labels then the hyperedge is assigned the mean weight of the *known* hyperedges. In the case of graphs, we set the weights of the edges to be 1.

The Extended DRW Algorithm

Putting all of these together, the complete EDRW procedure is as shown in Algorithm 1. First the attribute views and relations are converted into corresponding hypergraphs (lines 3 - 5, 8 - 10). Second, the Hellinger weights are computed for each of the hypergraphs (lines 6 and 11), then the random walk operator corresponding to all the views is computed (lines 12 - 14). The α's required in this step are empirically estimated through five-fold cross validation. Finally the DRW procedure is called with this random walk operator and the resulting steady state probabilities (line 16).

4 Experimental Setup

The experiments aim to show the effectiveness of the different aspects of the Extended DRW method - construction of the attribute view hypergraph, inference with very few labeled samples; with multiple views and multiple relations; and when the data exhibits high class-imbalance. Towards this, we have run experiments on synthetic and real datasets.

Algorithm 1 Extended Discriminative Random Walk

```
 1: function MVMR.PREDICT($Views$, $Relations$, $Y$)
 2:     i ← 0
 3:     for $view \in Views$ do
 4:         i←i+1
 5:         $H_i$ ← convertToHypergraph(view, $Y$)
 6:         $W_i$ ← computeHellingerWeights($H_i, Y$)
 7:                                          ▷ Using equation 13
 8:     for $relation \in Relations$ do
 9:         i ← i+1
10:         $H_i$ ← relation
11:         $W_i$ ← computeHellingerWeights($H_i, Y$)
12:     for $j = 1 : length(H)$ do
13:         [$P_j$, $\Pi_j$] ← computeStochasticMatrix($H_j$)
14:                                     ▷ Using Equations 8 and 9
15:     [$\mathbb{P}$, $\mathbf{\Pi}$] ← combineHypergraphs($P$, $\Pi$)
16:                                          ▷ Using Equation 11
17:     labels ← D-RandomWalk($\mathbb{P}$, $\mathbf{\Pi}$)
18:     return labels
```

The synthetic datasets consist of a graph and a hypergraph of 1000 nodes and two classes. In order to show the effectiveness of EDRW under different class skews, we considered 3 different class skews - 1:1, 1:5 and 1:20 - and for each skew, generated 5 sets, each consisting of a hypergraph and a graph. Our method is compared with the work of [Zhou and Burges, 2007] with modifications to handle hypergraphs, and ICA with bootstrapping, and also with a modified version of ICA that uses transductive SVM as its classifier. In order to show the effectiveness of Hellinger distance, we have also compared against the results obtained by weighting the hyperedges based on purity, and on assigning equal weights to all hyperedges. For generating synthetic hypergraphs, we needed to decide on the degree distribution of the hyperedges and the homophily factor. We say that a hypergraph exhibits homophily if the class distribution on a significant number of hyperedges is significantly different than the base class distribution of the whole dataset. We estimated both these factors from many real-world data sets. The degrees of the hyperedges were drawn from a modified power law distributions, and mean homophily was set at 0.4 for hypergraphs and 0.6 for graphs.[1]

We have also used real-world data with varying number of views and relations, to compare our method against other multi-view and collective classification approaches. The details about the various datasets used can be found in Table 1. We performed further experiments that we have not reported here due to lack of space. For a given percentage of unknowns, experiments were run over multiple partitions of the dataset into training and test data, and the average of all the runs have been reported. In all the experiments, we found that the performance was similar for walk lengths of 2, 3 and 4, reminiscent of results reported in [Callut *et al.*, 2008]. The results reported here are for $L = 2$. For data with more than one graph or hypergraph, the weights for each of the graphs and hypergraphs were found using 5-fold cross-validation.

Table 1: Dataset Description

Dataset	No. of nodes	No. of classes	Attr. views	Rel. views
WebKB	877	5	1	0
Cora	2708	7	1	1
Citeseer	3312	6	1	1
Twitter Football	248	20	2	6
Twitter Olympics	464	28	2	6

5 Results

The results shown in bold are significantly better at a p-value of 0.05 according to paired sample t-test. When all methods perform equally well in an experiment, none of the results are shown in bold. The results that are reported are the macro F1-scores, which is deemed to be better than other measures such as accuracy in the presence of class-imbalance. The same test and training data split was used for comparison between multiple algorithms, for a particular run. The reported results have been measured over the complete set of unlabeled data.

Synthetic data

Experiments were run on 5 different sets of random graph and hypergraph for every skew ratio that was considered. On each of these sets, the experiments were run multiple times as mentioned above. The results are reported in Table 2 and are an average over a total of 50 different runs over the 5 sets of graphs for a given skew ratio. In the table, “EDRW” denotes the case where the experiments were run using Hellinger weights for the hypergraph, “Purity” the case where the edges were weighted by the purity of the hyperedge, and “Equal” the case where all hyperedges were given equal weights (unweighted). The results were compared against ICA, ICA with Transductive SVM (abbreviated as TICA in the table) as the classifier, and against the work of [Zhou and Burges, 2007] appropriately modified to handle hypergraphs (denoted as LP below).

It was observed that ICA performs well even under some amount of class-imbalance, provided the number of known training examples is sufficient. ICA with TSVM performs poorly as the number of unknowns increases, and with increasing skew. TSVM can handle partially labeled data, but under extreme class skew doesn’t seem to perform well in this setting. LP becomes worse with the increase of skew and with increase in the number of unknown data, while our method with Hellinger weights performs well even under extreme class imbalance and with lots of unknowns, achieving up to 35% improvement over ICA in one case.

Real-World Data

Skewed Dataset

WebKB [Sen *et al.*, 2008] is a skewed dataset of 5 classes with the ratio of the most frequently occurring class to the least frequently occurring one being 10:1. We combined the word corpus of all four universities into a single corpus, constructed a hypergraph and classified with different sizes of training and test set. Table 3 gives the results of comparison of EDRW with Hellinger trees [Cieslak and Chawla, 2008] and with SVMs. The use of Hellinger distance as a weight measure on a skewed dataset, along with hyperedges, helps EDRW perform better than the other methods.

One View and One Relation

For Cora and Citeseer [Sen *et al.*, 2008] datasets we constructed one higher order relationship hypergraph for word corpus of papers and another hypergraph for co-citation network. We have compared our algorithm with Iterative Classification Algorithm (ICA) [Sen *et al.*, 2008] with bootstrapping to initialize the labels. The results are shown in Table 4. For ICA, we treated the co-citation information as a binary relation and the words as node attributes. Since both the views were hypergraphs, with the ability to weight every single hyperedge of the views, we were able to better differentiate more important hyperedges from the less important ones,

[1] Due to lack of space we are not able to provide a complete description here. Along with the code of EDRW, the details and code for synthetic graph generation is available at `https://github.com/HariniA/EDRW`.

Table 2: Macro-F1 scores for synthetic datasets

Skew Ratio		Percentage of Unknowns			
		30%	50%	70%	90%
1:1	EDRW	0.9998	0.9992	0.9991	0.9880
	Equal	0.9972	0.9984	0.9965	0.9764
	Purity	0.9971	0.9971	0.9969	0.9839
	LP	0.9998	0.9997	0.9993	0.9978
	ICA	1.0000	1.0000	1.0000	**0.9999**
	TICA	0.9900	0.7721	0.5000	0.5000
1:5	EDRW	**0.9877**	**0.9602**	**0.9051**	**0.7837**
	Equal	0.9486	0.8949	0.8405	0.7305
	Purity	0.9471	0.8922	0.8607	0.7113
	LP	0.9825	0.9631	0.8391	0.5053
	ICA	0.9076	0.8772	0.8230	0.4557
	TICA	0.7751	0.6489	0.3938	0.1435
1:20	EDRW	**0.8938**	**0.8060**	**0.7589**	**0.6761**
	Equal	0.8747	0.7663	0.7553	0.6705
	Purity	0.8791	0.7664	0.7557	0.6725
	LP	0.6408	0.5103	0.4875	0.4874
	ICA	0.8455	0.7672	0.7436	0.4927
	TICA	0.5175	0.3978	0.2814	0.0824

Table 3: Macro-F1 scores for WebKB dataset

	Macro F1-score		
Unknowns	H-Tree	SVM	EDRW
30%	0.6156	0.6577	**0.6860**
50%	0.4937	0.6175	**0.6665**
70%	0.5713	0.5452	**0.6314**
80%	0.5011	0.4773	**0.5864**
90%	0.4547	0.4015	**0.5077**

Table 4: Macro-F1 scores for Cora and Citeseer datasets

	Cora Macro F1		Citeseer Macro F1	
Unknowns	ICA	EDRW	ICA	EDRW
30%	0.7431	**0.8222**	0.6938	0.7094
50%	0.7336	**0.8204**	0.6760	**0.7083**
70%	0.6854	**0.7927**	0.6406	**0.6900**
90%	0.5456	**0.6385**	0.5614	**0.6174**

Table 5: Macro-F1 scores for Twitter Olympics dataset

	Macro F1		
Unknowns	CE	Jaccard Graph	EDRW
152(30%)	0.7122	0.9706	**0.9888**
241(50%)	0.6217	0.9380	**0.9525**
290(60%)	0.5381	0.9154	**0.9360**

Table 6: Macro-F1 scores for Twitter Football dataset

	Macro F1		
Unknowns	CE	Jaccard Graph	EDRW
82(30%)	0.7380	0.9078	**0.9579**
129(50%)	0.6573	0.8821	**0.9252**
181(70%)	0.5315	0.8349	**0.9007**
205(80%)	0.4422	0.7761	**0.8427**

and this led to good results for EDRW. Note that Cora dataset exhibits class imbalance with ratio of the minority class to the majority class being 1:4.5.

Multiple Views and Multiple Relations

Twitter Olympics and Twitter Football [Greene, 2013] are datasets with more than one view and one relation. We have compared EDRW to Collective Ensemble [Eldardiry and Neville, 2011]. This supports multiple relation classification, but we concatenate the multiple-views to produce a single attribute description. In order to check the efficiency of the hypergraph approach, we also constructed a graph based on the Jaccard similarity of the attributes of the nodes, with edges being present between two nodes if their similarity was above a particular threshold, and ran our algorithm using those graphs rather than the hypergraphs. As we can see from tables 5 and 6, the hypergraph model performs better, despite there not being much of a class skew. This is as expected, since the hypergraph allows us to look at similarity at individual attribute levels, while the Jaccard similarity looks at an aggregated distance measure. Collective Ensemble performs worse than EDRW in most of these experiments due to the fact that the attributes from two views are concatenated together as a single view, thus leading to a bloating up of the feature space, which might lead to issues like overfitting. One more reason for the better performance of EDRW are the weights given to different attribute and relational views, which is missing in Collective Ensemble.

6 Conclusion

In this work, we proposed Extended Discriminative Random Walks (EDRW), by defining a random walk operator on multiple graphs and hypergraphs, facilitating transductive inference on multi-view, multi-relational data. One of the key innovations of the work is a hyperedge weighing scheme based on Hellinger distance that helps improve performance in case of class-imbalance, which we empirically established, on both synthetic and real data. The advantages of this formulation are: unified representation of attribute descriptions, relational information and multi-way relations; a random walk operator that handles any number of views and relations; good performance even in the presence of class-imbalance and a small number of known labels. In order to enable this approach to scale to very large datasets we are working on graph sparsification techniques for faster inference.

Acknowledgments

This work was partly supported by a grant from Ericsson Research to Balaraman Ravindran. The authors would also like to thank Sudarsun Santhiappan of RISE Lab, IIT Madras, for his valuable feedback regarding the paper.

References

[Blum and Mitchell, 1998] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pages 92–100, New York, NY, USA, 1998. ACM.

[Callut *et al.*, 2008] Jérôme Callut, Kevin Françoisse, Marco Saerens, and Pierre Dupont. Semi-supervised classification from discriminative random walks. In *Machine Learning and Knowledge Discovery in Databases*, pages 162–177. Springer, 2008.

[Castillo *et al.*, 2007] Carlos Castillo, Debora Donato, Aristides Gionis, Vanessa Murdock, and Fabrizio Silvestri. Know your neighbors: Web spam detection using the web topology. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 423–430. ACM, 2007.

[Chakrabarti *et al.*, 1998] Soumen Chakrabarti, Byron Dom, and Piotr Indyk. Enhanced hypertext categorization using hyperlinks. In *ACM SIGMOD Record*, volume 27, pages 307–318. ACM, 1998.

[Cieslak and Chawla, 2008] David A Cieslak and Nitesh V Chawla. Learning decision trees for unbalanced data. In *Machine learning and knowledge discovery in databases*, pages 241–256. Springer, 2008.

[Cieslak *et al.*, 2012] David A. Cieslak, T. Ryan Hoens, Nitesh V. Chawla, and W. Philip Kegelmeyer. Hellinger distance decision trees are robust and skew-insensitive. *Data Min. Knowl. Discov.*, 24(1):136–158, 2012.

[Desrosiers and Karypis, 2009] Christian Desrosiers and George Karypis. Within-network classification using local structure similarity. In *Machine Learning and Knowledge Discovery in Databases*, pages 260–275. Springer, 2009.

[Domingos and Richardson, 2001] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66. ACM, 2001.

[Eldardiry and Neville, 2011] Hoda Eldardiry and Jennifer Neville. Across-model collective ensemble classification. In *AAAI*, 2011.

[Gao *et al.*, 2011] Yue Gao, Meng Wang, Huanbo Luan, Jialie Shen, Shuicheng Yan, and Dacheng Tao. Tag-based social image search with visual-text joint hypergraph learning. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1517–1520. ACM, 2011.

[Greene, 2013] D. Greene. http://mlg.ucd.ie/aggregation/. 2013.

[Mantrach *et al.*, 2011] Amin Mantrach, Nicolas Van Zeebroeck, Pascal Francq, Masashi Shimbo, Hugues Bersini, and Marco Saerens. Semi-supervised classification and betweenness computation on large, sparse, directed graphs. *Pattern recognition*, 44(6):1212–1224, 2011.

[Sen *et al.*, 2008] Prithviraj Sen, Galileo Mark Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[Shi and Malik, 2000] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[Shi *et al.*, 2012] Xiaoxiao Shi, Jean-Francois Paiement, David Grangier, and S Yu Philip. Learning from heterogeneous sources via gradient boosting consensus. In *SDM*, pages 224–235. SIAM, 2012.

[Sindhwani *et al.*, 2005] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *Proceedings of ICML workshop on learning with multiple views*, pages 74–79. Citeseer, 2005.

[Sun *et al.*, 2008] Liang Sun, Shuiwang Ji, and Jieping Ye. Hypergraph spectral learning for multi-label classification. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 668–676. ACM, 2008.

[Sun, 2013] Shiliang Sun. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7-8):2031–2038, 2013.

[Vijayan *et al.*, 2014] Priyesh Vijayan, Shivashankar Subramanian, and Balaraman Ravindran. Multi-label collective classification in multi-attribute multi-relational network data. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pages 509–514. IEEE, 2014.

[Xu *et al.*, 2013] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

[Yu *et al.*, 2012] Jun Yu, Dacheng Tao, and Meng Wang. Adaptive hypergraph learning and its application in image classification. *Image Processing, IEEE Transactions on*, 21(7):3262–3272, 2012.

[Zhou and Burges, 2007] Dengyong Zhou and Christopher J. C. Burges. Spectral clustering and transductive learning with multiple views. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 1159–1166, New York, NY, USA, 2007. ACM.

[Zhou *et al.*, 2006] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems (NIPS) 19*, page 2006. MIT Press, 2006.

[Zhu and B.Goldberg, 2009] Xiaojin Zhu and Andrew B.Goldberg. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–130, 2009.