Crowdsourcing of Network Data

Ding Wang
Department of Computer Science and Engineering
Michigan State University
Email: wangdin1@cse.msu.edu

Prakash Mandayam Comar
Amazon
Email: mc.prakash@gmail.com

Pang-Ning Tan
Department of Computer Science and Engineering
Michigan State University
Email: ptan@cse.msu.edu

***Abstract*—A key requirement for supervised learning is the availability of sufficient amount of labeled data to build an accurate prediction model. However, obtaining labeled data can be manually tedious and expensive. This paper examines the use of crowdsourcing technology to acquire labeled examples for classifying network data. Unfortunately, creating human intelligence tasks (HITs) to enable crowdsourcing is cumbersome for network data and may even be prohibitive for privacy reasons. To overcome this limitation, we present a novel framework called surrogate learning to transform the network data into a new representation (i.e., images) so that the labeling task can be completed even by non-domain experts. We analyze the reconstruction error of the transformation and use the theoretical insights to provide guidance on how to develop an effective surrogate learning approach for any given network and source image corpus. We also performed extensive experiments using Amazon Mechanical Turk to demonstrate the efficacy of our approach on node classification problems.**

I. Introduction

Labeled data is essential for various supervised network mining tasks such as node classification and link prediction. While for the most part the labels can be gleaned from the raw network itself, they are often incomplete and noisy, thus requiring alternative approaches to solicit more labels to augment the initial training data. Domain expertise is typically needed to manually examine the data instances before assigning them to their appropriate labels. Since this is a tedious and time consuming process, it may not always generate enough labeled examples. This paper examines the viability of using crowdsourcing for obtaining additional labeled data for network classification tasks.

Crowdsourcing [5] employs a group of human workers, who might be unskilled, to perform certain laborious tasks that cannot be reliably solved by computers. This includes tasks such as image annotation, where humans tend to perform the task more accurately than computers. The key challenge for harnessing the power of the crowd lies in converting the problem at hand into a simpler task that can be handled by humans with great ease and speed. Such tasks are known as Human Intelligence Tasks (HITs). For example, in the image annotation problem, the individual images constitute a HIT, which are displayed to workers in order to elicit their label information.

One practical advantage of utilizing the services of crowdsourcing is that once the HITs are designed, they can be solved by human workers with little domain expertise. The valuable time of domain experts can therefore be spared from performing cumbersome data labeling task. Unfortunately, not all labeling tasks are amenable to crowdsourcing. For example, *designing HITs for network mining is challenging as the raw network data does not lend itself to be easily annotated by non-domain experts.* This is because the label of a node in a network often depends on its local attributes (if available) as well as its relationships to other nodes in the network. *In addition, there may be privacy concerns that prohibit sharing of the network data to a third party.* This makes it difficult to design HITs that are simple for the average humans to solve without disclosing potentially sensitive information to the workers.

This paper presents a novel framework called *surrogate learning* to transform the network data into a representation that can be more easily annotated by the crowd. The proposed approach does not require the workers to have *a priori* knowledge or expertise on how to correctly label the nodes or links of the network. To illustrate this approach, consider the toy example shown in Figure 1. Here the nodes in the target network are assigned to four distinct classes (i.e., communities), labeled as A, B, C and D, respectively. The solid circles correspond to the labeled nodes, whereas the unfilled ones represent the unlabeled nodes. The surrogate learning framework selects a *surrogate image* to represent each labeled node in the target network. It also learns a transformation matrix to map the nodes in the network to their corresponding images. The transformation matrix can then be applied to any unlabeled nodes to generate their corresponding images for labeling by the crowd workers.

The source images shown in Figure 1(b) correspond to a set of handwritten digits $1, 2, 3$ and 4. To preserve characteristics of the target network, the surrogate mapping must be done in such a way that (1) nodes from the same class should be mapped to images for the same digit. and (2) nodes that are adjacent to each other should be mapped to similar images. Figure 1(c) shows the transformed images for the unlabeled nodes. Since all the labeled nodes from class C had been mapped to images for digit 3, the images for the unlabeled nodes $C1$ and $C4$ also resemble the digit 3. However, the images for some unlabeled nodes such as $A3$ are harder to discern since they are adjacent to nodes from other classes. Furthermore, the node $D1$, which has more links to nodes from class B than to those from its own class, is transformed

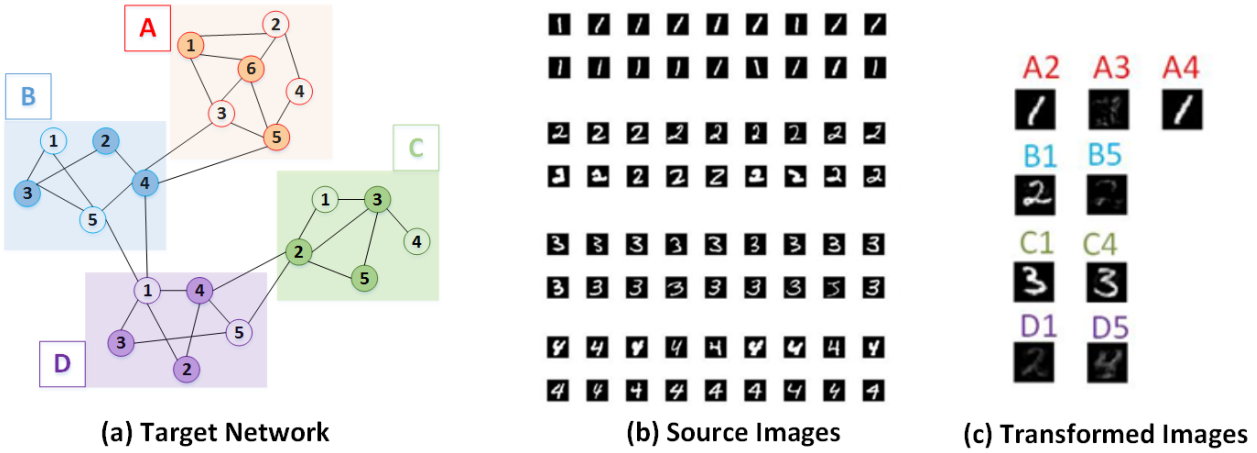


Fig. 1: A toy example consisting of a *target* network (a) and its corresponding handwritten digit images (b). There are 12 labeled (solid circles) and 9 unlabeled (unfilled circles) nodes in the network. Each labeled node is initially mapped to one of the source images. Labeled nodes from class A were mapped to images for digit 1, class B to digit 2, class C to digit 3, and class D to digit 4. A transformation function is then learned between the labeled nodes and their corresponding images. After applying the transformation to the unlabeled nodes, the transformed images will be presented to the crowd workers for labeling.

into an image that resembles digit 2 instead of digit 4.

A key question that must be answered is whether the proposed framework can be effectively applied to any target network and source images. To answer this question, we analyze the reconstruction error of the transformation and use the theoretical insights to provide guidance on how to develop an effective approach for any given network and source image corpus. We also empirically showed the effectiveness of the framework in providing additional labeled data for node classification and link prediction tasks.

The main contributions of this paper are as follows:

1) We investigated the problem of applying crowdsourcing to network data.
2) We developed a novel framework called surrogate learning to transform network data into an image representation to enable labeling by the crowd.
3) We offer practical guide on how to effectively apply the framework to any network and image data.
4) We demonstrated the efficacy of the framework for various network classification problems.

II. Preliminaries

Let $\mathcal{S}$ and $\mathcal{T}$ denote the source and target domains. The source domain corresponds to an image corpus for which labels can be easily acquired through crowdsourcing whereas the target is a large network for which obtaining labels is expensive.

Let $\mathbf{X}^{(s)}$ be an $n_s \times d_s$ data matrix for the source images and $\mathbf{Y}^{(s)}$ be the corresponding $n_s \times c_s$ class membership matrix, where n_s is the number of labeled images, d_s is the number of attributes (pixels), and c_s is the number of classes. Each element $y^s_{ij} \in \mathbf{Y}^{(s)}$ is equal to 1 if the labeled image $\mathbf{x}^{(s)}_i$ belongs to class j and zero otherwise. Similarly, let $\mathbf{X}^{(t)} = [\mathbf{X}^{(tl)}; \mathbf{X}^{(tu)}]$ denote an $(n_t + r) \times d_t$ data matrix for the target network and $\mathbf{Y}^{(tl)}$ be its corresponding $n_t \times c_t$ class membership matrix, where n_t is the number of labeled examples, r is the number of unlabeled examples, d_t is the number of attributes, and c_t is the number of classes in the target network. For node classification, each example corresponds to a node in the network, whereas for link prediction, each example corresponds to a node pair. In addition, let $\mathbf{A}$ denote the adjacency matrix of the network. For brevity, we focus only on undirected networks in this study.

Source Image Corpus The example shown in Figure 1 uses images of handwritten digits as the source data. In practice, there are many other types of image corpus that can potentially be used to generate the surrogate images for crowdsourcing. The choice of source data should satisfy several criteria. First, the images must be clear and easy to label. For instance, we had investigated transforming the data using principal component analysis, but found that it does not guarantee the transformed data can be easily interpreted by humans. Second, the images for different classes should be well-separated. Third, the number of classes of images should be at least as large as the number of classes in the target data to enable label mapping. Finally, as will be discussed in Section IV-B, the number of attributes in the target data should be comparable to the number of pixels in the image data to minimize information loss.

III. Surrogate Learning Framework

Our proposed surrogate learning framework consists of the following three steps:

I. **Surrogate Mapping.** Given $\mathcal{S} = (\mathbf{X}^{(s)}, \mathbf{Y}^{(s)})$ and $\mathcal{T} = (\mathbf{X}^{(tl)}, \mathbf{Y}^{(tl)})$, we need to learn a transformation matrix $\mathbf{U}$ that maps each target example $\mathbf{x}^{(tl)}_i \in \mathbf{X}^{(tl)}$ to its surrogate

image $\mathbf{x}_j^{(s)} \in \mathbf{X}^{(s)}$ in such a way that satisfies the following two criteria:

- *Label Consistency:* Target examples that belong to the same class should be mapped to images from the same class.
- *Link Similarity:* If there is a link between node i and j in the network, their corresponding surrogate images should be highly similar.

II. **Surrogate Labeling.** The transformation matrix $\mathbf{U}$ will be applied to the unlabeled examples of the target network $\mathbf{X}^{(tu)}$ to generate their corresponding images $\hat{\mathbf{X}}^{(su)}$ that will be labeled by the crowd workers. Since each image can be labeled by more than one worker, a consensus on the class label must be made for each target example. Let $\mathbf{Y}^{(tu)}$ denote the consensus labels obtained for the unlabeled target examples.

III. **Model Building.** The newly labeled target examples $(\mathbf{X}^{tu}, \mathbf{Y}^{tu})$ are augmented to the original training set. A classifier is then trained on the expanded training set to generate a new model.

IV. Surrogate Mapping

Our goal is to choose a surrogate image for each target example of the network data and learn the transformation matrix $\mathbf{U}$. The mapping does not have to be a bijection, i.e., multiple target examples can be mapped to the same surrogate image. However, it should preserve the label consistency and link similarity requirements. Let $\mathbf{P}$ be an $n_t \times n_s$ matrix, where $\mathbf{P}_{ij} = 1$ if the source image $\mathbf{x}_j^{(s)}$ is the surrogate for the target node $\mathbf{x}_i^{(t)}$. The objective function for the surrogate mapping task is given below

$$\begin{aligned}\min_{\mathbf{U},\mathbf{P},\mathbf{Q}} \; & \parallel \mathbf{PX}^{(s)} - \mathbf{X}^{(tl)}\mathbf{U} \parallel_F^2 + \parallel \mathbf{PY}^{(s)}\mathbf{Q}^T - \mathbf{Y}^{(tl)} \parallel_F^2 \\ & + \alpha \sum_{i<j} \mathbf{A}_{ij} \parallel (\mathbf{PX}^{(s)})_i - (\mathbf{PX}^{(s)})_j \parallel^2 + \beta \|\mathbf{U}\|_F^2 \\ \text{s.t.} \quad & \forall i,j: \; \mathbf{P}_{ij} \in \{0,1\}, \quad \mathbf{P}\mathbf{1}_{n_s} = \mathbf{1}_{n_t}, \\ & \forall i,j: \; \mathbf{Q}_{ij} \in \{0,1\}, \quad \mathbf{Q}\mathbf{1}_{c_s} = \mathbf{1}_{c_t}, \end{aligned} \tag{1}$$

The first term in the objective function is a measure of reconstruction error when converting the target examples into images. The second and third terms in the objective function enforce the label consistency and link similarity requirements. The last term is a regularization term for $\mathbf{U}$. The image data is normalized so that each pixel has a value between 0 and 1. The constraint on matrix $\mathbf{P}$ ensures that its elements are binary-valued and that each target example is mapped to exactly one source image. The label matching matrix $\mathbf{Q}$ is a $c_s \times c_t$ binary-valued matrix that represents the mapping between the class labels of the source images and the target examples of the network. α is a user-specified parameter that controls the tradeoff between minimizing reconstruction error and violating the link similarity constraint.

Although the formulation given in (1) assumes that the target examples have both node attributes and link information, it is also applicable to non-relational data by setting $\alpha = 0$. For network classification tasks in which we have only link structure but no node attribute information, we can also set $\alpha = 0$ and assume either $\mathbf{X}^{(t)} = \mathbf{A}$ (i.e., using the link information as attributes of the target examples) or derive link-based features such as betweenness centrality, clustering coefficient, etc., to represent $\mathbf{X}^{(t)}$.

The *surrogate selection matrix* $\mathbf{P}$ allows a many-to-one assignment between the target examples and the source images. This is essential because, for any two target examples that are similar to each other and belong to the same class, $\mathbf{P}$ should assign them to images that resemble each other. If no such pair of images can be found, it would be better to map both target examples to the same surrogate image.

A. Optimization

We employ an alternating least square method to solve the optimization problem given in (1). Since the classes are unrelated, we can set $\mathbf{Q}$ to be an identity matrix.[1] The algorithm would iteratively update the matrices $\mathbf{P}$ and $\mathbf{U}$ until convergence. We begin with an initial surrogate selection matrix $\mathbf{P}^0$ obtained by randomly assigning images from a specific class to target examples of its corresponding class according to $\mathbf{Q}$ in order to satisfy the label consistency criterion, $\mathbf{P}^0\mathbf{Y}^{(s)} = \mathbf{Y}^{(t)}$. Let $\mathbf{Z} = \mathbf{P}^0\mathbf{X}^{(s)}$ denote the matrix for the selected surrogate images of the target examples. By fixing $\mathbf{P}$, the transformation matrix can be solved in closed form as follows

$$\mathbf{U} = \left(\mathbf{X}^{(tl)T}\mathbf{X}^{(tl)} + \beta\mathbf{I}\right)^{-1}\mathbf{X}^{(tl)T}\mathbf{Z}, \tag{2}$$

The regularizer β ensures that the matrix is invertible even if $\mathbf{X}^{(tl)}$ is not a full-rank matrix.

Next, we fix $\mathbf{U}$ and update $\mathbf{P}$ using a greedy approach. For each target example $\mathbf{x}_i^{(tl)}$, we first compute its transformed image $\hat{\mathbf{x}}_i^{(tl)} = \mathbf{x}_i^{(tl)T}\mathbf{U}$. Let $\mathbf{z}_i^{k-1} = (\mathbf{P}^{k-1}\mathbf{X}^{(s)})_i$ be the surrogate image selected for the i-th target example in the previous iteration. The change in the objective function if the previous surrogate image $\mathbf{z}_i^{k-1}$ is replaced by a new surrogate $\mathbf{x}_j^{(s)}$ is given by

$$\begin{aligned} & \Delta_i(\mathbf{z}_i^{k-1}, \mathbf{x}_j^{(s)}) \\ = \; & \parallel \mathbf{x}_j^{(s)} - \hat{\mathbf{x}}_i^{(tl)} \parallel^2 + \alpha \sum_{k:A_{ik}=1} \parallel \mathbf{x}_j^{(s)} - (\mathbf{PX}^{(s)})_k \parallel^2 \\ - \; & \parallel \mathbf{z}_i^{k-1} - \hat{\mathbf{x}}_i^{(tl)} \parallel^2 - \alpha \sum_{k:A_{ik}=1} \parallel \mathbf{z}_i^{k-1} - (\mathbf{PX}^{(s)})_k \parallel^2 \end{aligned}$$

We choose the surrogate $\mathbf{z}_i^k = \mathbf{x}_{l_i}^{(s)}$ that leads to the biggest decrease in Δ_i:

$$l_i \;=\; \text{argmin}_{j:y_j^{(s)}=y_i^{(tl)}} \; \Delta_i(\mathbf{z}_i^{k-1}, \mathbf{x}_j^{(s)}) \tag{3}$$

We then set $\mathbf{P}_{i,l_i}^k = 1$ and $\mathbf{P}_{i,j}^k = 0 \; (\forall j \neq l_i)$. Note that a new surrogate is selected using Equation (3) only if $\Delta_i(\mathbf{z}_i^{k-1}, \mathbf{x}_{l_i}^{(s)}) < 0$. Otherwise, $\mathbf{z}_i^k = \mathbf{z}_i^{k-1}$. We iteratively

[1] A more careful selection of $\mathbf{Q}$ would require considerations of the within-class and between-class variability of the source and target examples. We plan to pursue this in future research.

Algorithm 1 `Surrogate Mapping Algorithm`

Input: $\mathbf{X}^{(s)}$, $\mathbf{X}^{(tl)}$, $\mathbf{Y}^{(s)}$, $\mathbf{Y}^{(t)}$
Output: Transformation Matrix $\mathbf{U}$
k = 0;
Initialize $\mathbf{P}^0$, satisfying $\mathbf{P}^0\mathbf{Y}^{(s)} = \mathbf{Y}^{(t)}$
repeat
  $k = k + 1$
  $\mathbf{U}^k = \text{argmin}_{\mathbf{U}} \parallel \mathbf{P}^{k-1}\mathbf{X}^{(s)} - \mathbf{X}^{(tl)}\mathbf{U} \parallel^2$
  for $i = 1, 2, \ldots, n$ **do**
    $\hat{\mathbf{x}}_i^{(tl)} = \mathbf{x}_i^{(tl)T}\mathbf{U}^k$
    Update the row $\mathbf{P}_i^k$ using Equation (3).
  end for
until $\mathbf{P}^{k-1} = \mathbf{P}^k$
`return` $\mathbf{U}^k$

update the matrix $\mathbf{P}^k$, starting from the first row until the last one. Due to the way the surrogates are selected, $\forall i : \Delta_i(\mathbf{z}_i^{k-1}, \mathbf{x}_{l_i}^{(s)}) \leq 0$, this guarantees that the objective function is monotonically non-increasing. A summary of our surrogate mapping approach is given in Algorithm 1. The proof of convergence of the algorithm is omitted due to lack of space.

B. Reconstruction Error Analysis

Our proposed framework is designed to transform a target network into images so that they can be easily annotated by the crowd. This begs the question, whether *it is always possible to find a transformation that preserves properties of the target data while producing images that can be easily discerned by workers*. To measure the amount of information loss due to the transformation, we compute its reconstruction error. By analyzing the theoretical properties of the error, we provide guidance on how to develop an effective surrogate learning procedure for any target network and choice of image corpus.

Let $\mathbf{Z} = \mathbf{P}\mathbf{X}^{(s)}$ be the surrogate images selected to represent $\mathbf{X}^{(tl)}$. If the reconstruction error $\|\mathbf{Z} - \mathbf{X}^{(tl)}\mathbf{U}\|_F$ is small, we expect a minimal loss of information since $\mathbf{X}^{(tl)}$ can be reconstructed from $\mathbf{Z}$ and $\mathbf{U}$ with high accuracy. To illustrate this, let $\mathbf{Z} = \mathbf{X}^{(tl)}\mathbf{U} + \epsilon$ and $\hat{\mathbf{X}}^{(tl)} = \mathbf{Z}\mathbf{U}^T(\mathbf{U}\mathbf{U}^T)^{-1}$. Thus,

$$\hat{\mathbf{X}}^{(tl)} = (\mathbf{X}^{(tl)}\mathbf{U} + \epsilon)\mathbf{U}^T(\mathbf{U}\mathbf{U}^T)^{-1} = \mathbf{X}^{(tl)} + \epsilon\mathbf{U}^T(\mathbf{U}\mathbf{U}^T)^{-1}$$

$$\implies \|\hat{\mathbf{X}}^{(tl)} - \mathbf{X}^{(tl)}\|_F \leq \|\epsilon\|_F \sqrt{\mathbf{tr}(\mathbf{U}\mathbf{U}^T)^{-1}}$$

which decreases as $\|\epsilon\|_F \to 0$. Furthermore, if the reconstruction error is small, $\mathbf{X}^{(tl)}\mathbf{U} \approx \mathbf{Z}$, which means the digits depicted in the transformed images should be discernible by human workers. The applicability of the proposed framework thus becomes a question of determining whether it is possible to obtain a low reconstruction error for any network. To determine the condition under which a low reconstruction error can be found, we examine the ranks of the data matrices:

Proposition 1. *Let* $\mathbf{A}$ *be an* $m \times n$ *matrix and* $\mathbf{B}$ *be an* $n \times k$ *matrix. If* $r(\mathbf{A})$, $r(\mathbf{B})$, *and* $r(\mathbf{AB})$ *denote the ranks of matrices* $\mathbf{A}$, $\mathbf{B}$, *and* $\mathbf{AB}$, *respectively, then it can be shown [2] that* $r(\mathbf{AB}) \leq \min[r(\mathbf{A}), r(\mathbf{B})]$.

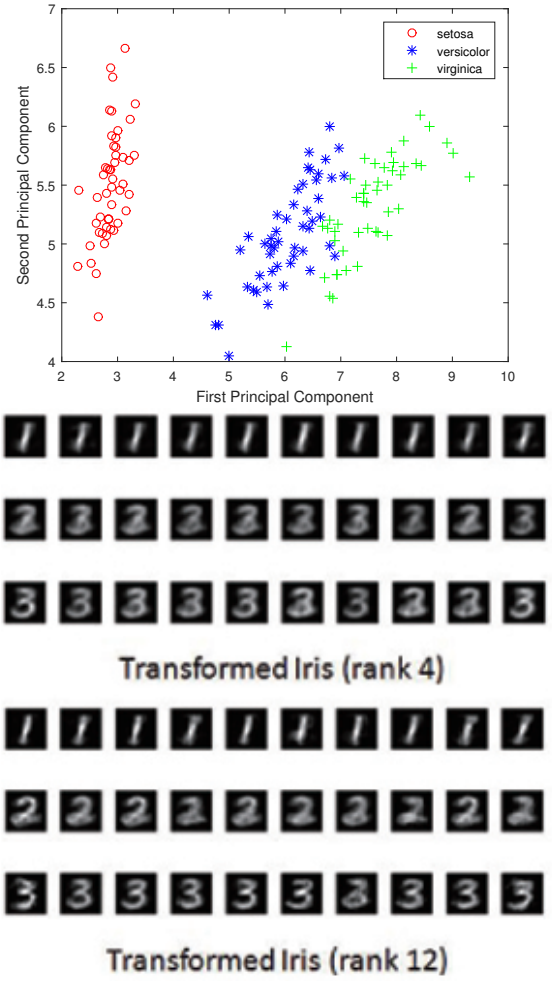


Fig. 2: Transforming iris data into handwritten digit images. The middle and bottom panels show the transformed images when the iris data contains 4 and 12 features, respectively.

To understand the implication of Proposition 1, let $\hat{\mathbf{X}}^{(s)} = \mathbf{X}^{(tl)}\mathbf{U}$ be the transformed images of the target examples. According to the proposition, $r(\hat{\mathbf{X}}^{(s)}) \leq r(\mathbf{X}^{(tl)})$. Thus, if the rank of the target data matrix is considerably lower than that for the original source image matrix, then $r(\hat{\mathbf{X}}^{(s)}) = r(\mathbf{X}^{(tl)}\mathbf{U}) << r(\mathbf{X}^{(s)})$, which leads to a large reconstruction error.

We illustrate this in Figure 2 using the well-known Iris data[2] containing 150 examples belonging to 3 categories (Iris versicolor, Iris virginica, and Iris setosa) [3]. Each category contains 50 examples, which are matched against 50 handwritten images of 28×28 dimensions containing the digits 1, 2, or 3. The rank of the data matrix for the handwritten images is 150, which is considerably higher than the rank of

[2]https://archive.ics.uci.edu/ml/datasets/Iris

[3]Although this is not a network data, the surrogate mapping algorithm is still applicable by setting $\alpha = 0$.

Iris data, which is equal to 4. As a result, the reconstruction error using the handwritten images is high, as shown in the middle diagram of Figure 2. Even though all the examples in the Iris versicolor class were mapped to images containing digit 2, their transformed images ($\hat{\mathbf{X}}^{(s)}$) look noisy and do not resemble digit 2. Instead, they looked like a mixture of digits 2 and 3 since it is hard to distinguish the Iris versicolor class from Iris virginica. However, if we increase the dimensionality of the Iris data from 4 to 12 (by adding quadratic and cubic terms for each of the original features), the reconstruction error reduces significantly especially for the Iris versicolor class, as shown in the bottom diagram of Figure 2). Finally, if we project the Iris data to a 150-dimensional feature space (using higher order polynomials), the reconstruction error is close to zero.

The key lesson here is that it is preferable to have a source data whose rank is smaller than that of the target data. There are two ways to achieve this. First, we can reduce the rank of the source data by applying singular value decomposition. The drawback here is that the rank reduction may damage the visual clarity of the images, which makes them hard to be labeled by humans. Alternatively, we can increase the rank of the target data by projecting them to a higher dimensional space, similar to the approach used in kernel learning. This leads to the following guide to achieve low reconstruction error:

Practical Guide: *Compare the ranks of the source images and target network data matrices. If the former is larger than the latter, increase the rank of the target (e.g., by projecting the data to a higher-dimensional space) before applying the surrogate mapping algorithm.*

The need for a lower-rank source data matrix, or equivalently, higher-rank target data matrix, can also be justified based on the following theorem:

Theorem 1. *If $\mathbf{X}^{(tl)}$ and $\mathbf{Z}$ are full-rank matrices and $\beta = 0$, then*

$$\|\mathbf{Z} - \mathbf{X}^{(tl)}\mathbf{U}\|_F \leq \sqrt{(N_t - d_t)d_s}\sigma_{\max},$$

where N_t is the number of target examples, d_s is the dimensionality of the surrogate images, d_t is the dimensionality of the target data, and $\sigma_{\max}$ is the maximum eigenvalue of $\mathbf{Z}$.

Proof. Let $\hat{\mathbf{P}} = \mathbf{X}^{(tl)}\left(\mathbf{X}^{(tl)T}\mathbf{X}^{(tl)}\right)^{-1}\mathbf{X}^{(tl)T}$. From Equation (2), $\mathbf{X}^{(tl)}\mathbf{U} = \hat{\mathbf{P}}\mathbf{Z}$ since $\beta = 0$. Therefore,

$$\begin{aligned} \| \mathbf{Z} - \mathbf{X}^{(tl)}\mathbf{U} \|_F^2 &= \| (\mathbf{I} - \hat{\mathbf{P}})\mathbf{Z} \|_F^2 \\ &\leq \| \mathbf{I} - \hat{\mathbf{P}} \|_F^2 \| \mathbf{Z} \|_F^2 \end{aligned} \quad (4)$$

It is easy to show that $\hat{\mathbf{P}}$ is a projection matrix, which satisfies the following properties: $\hat{\mathbf{P}}^2 = \hat{\mathbf{P}} = \hat{\mathbf{P}}^T$. Therefore,

$$\begin{aligned} \| \mathbf{I} - \hat{\mathbf{P}} \|_F^2 &= \mathbf{tr}\Big((\mathbf{I} - \hat{\mathbf{P}})(\mathbf{I} - \hat{\mathbf{P}})^T\Big) \\ &= \mathbf{tr}(\mathbf{I} - \hat{\mathbf{P}}) \\ &= N_t - \mathbf{rank}(\hat{\mathbf{P}}) \\ &= N_t - d_t \end{aligned} \quad (5)$$

where we have assumed $d_t < N_t$. Let $\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$ be the singular value decomposition on $\mathbf{Z}$. Hence,

$$\| \mathbf{Z} \|_F^2 = \mathbf{tr}(\mathbf{Z}\mathbf{Z}^T) = \mathbf{tr}(\mathbf{\Sigma^2}) = \sum_i \sigma_i^2 \leq d_s\sigma_{max}^2 \quad (6)$$

The proof follows by replacing Equations (5) and (6) into the inequality in (4). □

V. Surrogate Labeling and Model Building

Our goal is to acquire additional labeled examples for network data by creating HITs that can be easily solved by non-expert workers. Once the transformation matrix $\mathbf{U}$ has been estimated, it can be applied to the unlabeled target examples to generate images for the workers to label. Some of the generated images can be hard to discern, for which the workers are allowed to flag them as noise. Noisy examples will not be augmented to the training set when building the classifier. Additionally, some images can be assigned to more than one label. In this case, a consensus label must be determined based on the labels provided by the workers.

We consider two ways to obtain the consensus label. First, we take a simple majority vote of the labels. If none of the labels have a majority vote, then the target example remains unlabeled and will not be augmented to the training set. The second way is train a *crowd classifier* that takes the labels provided by workers as input features to predict the actual class. To do this, we first apply the transformation matrix $\mathbf{U}$ to the labeled examples and present their transformed images to the crowd for labeling. Since the actual class of the images are known, we use this information to create a data set for training the crowd classifier. The features of the classifier correspond to the number of workers who assigned each label to the image. We termed the first approach using majority vote as **Surrogate-CM** and the second approach using crowd classifier as **Surrogate-CC**. Once the consensus labels are found, they are augmented to the original training data and subsequently provided to a classifier for training.

VI. Experimental Evaluation

We applied our surrogate learning framework to the node classification task using the following two data sets.

Wikipedia Biology Corpus: We sampled Wikipedia articles from 4 sub-categories in biology—genetics, zoology, anatomy and cell-biology. There are altogether 2128 links in the Wikipedia biology network. The TRAIN set consists of 2000 articles with 500 articles from each topic and the TEST set consists of 800 articles with 200 articles from each topic. We apply principal component analysis to reduce dimensionality of the data from 6015 words to 550 features.

Cora data set: The Cora data set contains 2708 computer science articles, categorized into one of seven classes. The citation network between articles has 5429 links. Each article is described by a binary-valued vector indicating the absence or presence of a word. There were altogether 1433 unique words, which were reduced to 550 features after applying

principal component analysis. 70% of the data were reserved for training, while the remaining 30% were used for testing.

We use handwritten digit images from [9] as our source corpus. There are roughly 5000 images for each digit from 0 through 9. Each image is of size 28×28 and is represented as a feature vector of length 784. Although the rank of the source image $\mathbf{X}^{(s)}$ is initially larger than the rank for target data $\mathbf{X}^{(tl)}$, the surrogate mapping algorithm iteratively reduces the rank of the selected surrogate image matrix ($\mathbf{PX}^{(s)}$) until its rank falls below that of $\mathbf{X}^{(tl)}$ upon convergence.

We employ nonlinear support vector machine (SVM) as our classifier. We compared the performance of nonlinear SVM trained on the original training set against the same classifier trained on the expanded training set, which includes the newly labeled examples acquired via crowdsourcing. Note that the expanded training set includes only examples that were not labeled as noise by the workers.

A. Results for Node Classification

The images generated by the surrogate learning framework for the unlabeled nodes in the network are presented to a crowd of five workers for labeling. We set $\alpha = 0.2$ and $\beta = 0.1$ as parameters of the surrogate mapping algorithm. The results do not appear to change significantly when α and β are varied in the range between 0 and 1.

1) Wikipedia Biology Corpus.: Table I shows the node classification results for the Wikipedia biology corpus. We first trained a nonlinear SVM model using Gaussian kernel with parameter $\sigma = 0.1$ as our baseline. The same parameter is used to train the SVM models for our surrogate learning framework. SVM gave a baseline accuracy of 63.50% on the TEST set. The average classification accuracy of Amazon MTurk workers on the TEST SET is around 73.37%. This suggests that the surrogate mapping algorithm was able to transform the nodes into crisp images that can be easily labeled by the workers. On average, the workers labeled 111 (14%) of the TEST examples as noise or having more than one class. After augmenting the training set with labeled examples acquired through crowdsourcing, both `SURROGATE-CM` and `SURROGATE-CC` boosted the baseline SVM accuracy to 77.50% and 77.63%, respectively. The F-measure for all classes also improved significantly. Furthermore, the accuracy of `Surrogate-CC` is slightly higher than `Surrogate-CM`, which suggests that the crowd classifier approach is slightly more effective than simple majority voting when combining the labels provided by the crowd.

To understand why the surrogate learning framework can improve classification accuracy, we examine the images that were wrongly classified by the baseline SVM model. Figure 3(a) shows images for a sample of test examples that were misclassified by SVM but labeled correctly by human workers. The images are sorted into rows based on their true labels (i.e., images on the first row correspond to class 1, those in second row correspond to class 2, and so on). Although the images are not as crisply clear, they are still distinctive enough for humans to identify their correct labels. Out of the 292 test images that were misclassified by the baseline SVM, human workers were able to label correctly 144 such images, which explains the improvement in the classification accuracy. Figure 3(b) shows examples of test images that were mislabeled by both baseline SVM and the majority of the workers. Although the images are quite clear for humans to label, they do not resemble the digits associated with their true classes. For example, the images shown in the first row resemble digits 2, 3, and 4 more than their true class, which is digit 1. This suggests that the feature vectors of these hard-to-classify examples are more similar to the feature vectors of other classes than to their own classes, which is why both SVM and human workers fail to classify them correctly. Nevertheless, there was a subset of images that were mislabeled by the baseline SVM and labeled as noise by one or more workers but were correctly predicted by the revised SVM after augmented with the expanded training set. Examples of such images are shown in Figure 3(c). There are 71 such images, which are significant enough to improve the overall accuracy.

One surprising finding is that the SVM model trained on the augmented training set is not that sensitive to the choice of kernel parameter σ. This is useful because finding the optimal kernel parameter for nonlinear SVM is a challenging problem. Figure 4 compares the accuracy of baseline SVM against both implementations of the surrogate learning framework when σ is varied. As can be seen from this plot, the performance of the baseline SVM is highly sensitive to the choice of kernel parameter unlike surrogate learning.

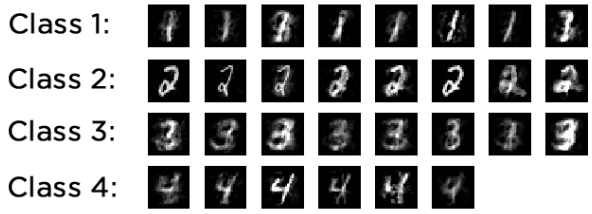
2) Cora Data Set.: Table II shows the results for the Cora data set. Unlike the Wikipedia biology corpus, the average

TABLE I: Comparison between baseline SVM, average performance of crowd workers, and surrogate labeling for node classification on Wikipedia biology corpus.

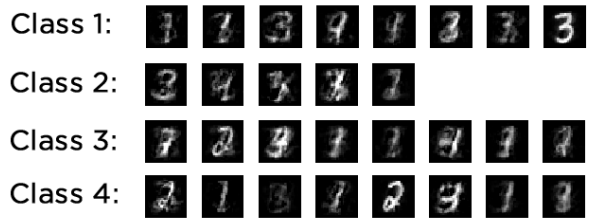
Class	SVM	$\text{Worker}_{\text{avg}}$	Surrogate-CM	Surrogate-CC
	F measure			
Zoology	0.5639	0.6388	0.6829	**0.6848**
Cell Biology	0.7250	0.8406	0.8818	**0.8861**
Anatomy	0.6267	0.7008	0.7269	**0.7284**
Genetics	0.6285	0.7333	**0.7980**	0.7960
All Classes	0.6360	0.7284	0.7724	**0.7738**
	Accuracy			
All Classes	0.6350	0.7337	0.7750	**0.7763**

TABLE II: Comparison between baseline SVM, average performance of crowd workers, and surrogate labeling for node classification on Cora dataset.

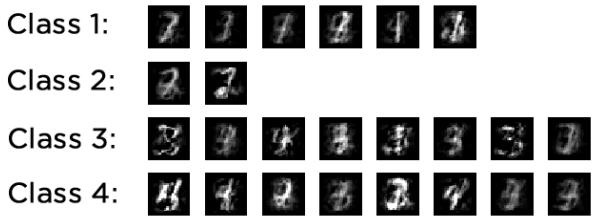
Class	SVM	$\text{Worker}_{\text{avg}}$	Surrogate-CM	Surrogate-CC
	F measure			
Case Based	0.6269	0.5394	0.6483	**0.6622**
Genetic Algo.	**0.8333**	0.7426	0.7982	0.7982
Neural Nets	0.6734	0.6968	0.6958	**0.6981**
Probabilistic	0.6794	0.6306	**0.7231**	**0.7231**
Reinforcement	0.6471	0.5469	0.6727	**0.6847**
Rule Learning	0.3636	0.3981	**0.4865**	**0.4865**
Theory	**0.6162**	0.4668	0.5990	0.6051
All Classes	0.6638	0.6144	0.6824	**0.6864**
	Accuracy			
All Classes	0.6716	0.6192	0.6875	**0.6912**



(a) Test images predicted incorrectly by baseline SVM but labeled correctly by the majority of human workers.



(b) Test images that were labeled incorrectly by both baseline SVM and the majority of workers.



(c) Test images misclassified by the baseline SVM and labeled as noise by workers but classified correctly after the SVM is re-trained using the expanded training set.

Fig. 3: Test examples mislabeled by baseline SVM and human workers for the Wikipedia biology corpus.

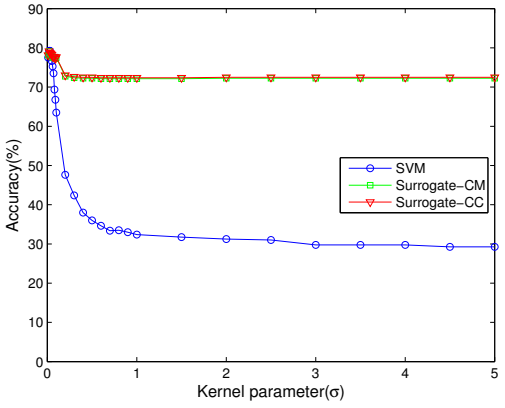


Fig. 4: Effect of varying kernel parameter on SVM and the surrogate learning framework.

accuracy for Amazon MTurk workers is less than the accuracy obtained using the baseline SVM. Nevertheless, there is still some improvement in the accuracy of SURROGATE-CC and SURROGATE-CM. More importantly, the F-measure for the majority of the classes show substantial improvements. For example, the F-measure for Rule Learning increases from 0.3636 to 0.4865 and for Reinforcement Learning, from 0.6471 to 0.6847. Furthermore, similar to the results for the Biology article corpus, the performance of SVM trained on the augmented data set does not change significantly when the kernel width parameter is varied.

B. Choice of Source Image Corpus

In this section, we investigated how the proposed framework is affected by the choice of source image corpus.

Separability of Image Classes: As noted in Section II, separability of the image classes is an important factor to consider when choosing the source image corpus. If the classes are well-separated, it would be easier (1) for the surrogate mapping procedure to select the appropriate images to represent the target examples and (2) for the crowd workers to label the images.

In this experiment, we examined each of the 126 possible combinations of 4 digit images from the handwritten image corpus to be used as source images for classifying the Wikipedia biology corpus. Specifically, for each combination of classes, we randomly chose 1000 examples from each class to form its training set. We applied the 1-nearest neighbor (1-NN) classifier to each image and check whether its nearest neighbor has the same label as the image itself. We use the accuracy of 1-NN to estimate the separability of the classes. If the classes are well-separated, then the accuracy of its 1-NN classifier should be high. Conversely, if the classes of images are harder to distinguish, the 1-NN classification accuracy should be low. We computed the class separability for all combinations of 4-digit classes in the handwritten image corpus and plot the results in Figure 5. The class separability values were found to be in a range between 0.95 to 0.99. The combination of classes {1,2,3,4} appears to be more well-separated compared to the combination of classes {4,5,8,9}. This is because, upon examining the confusion matrix for the latter combination, digits 4 and 9 as well as 5 and 8 are harder to be distinguished from each other.

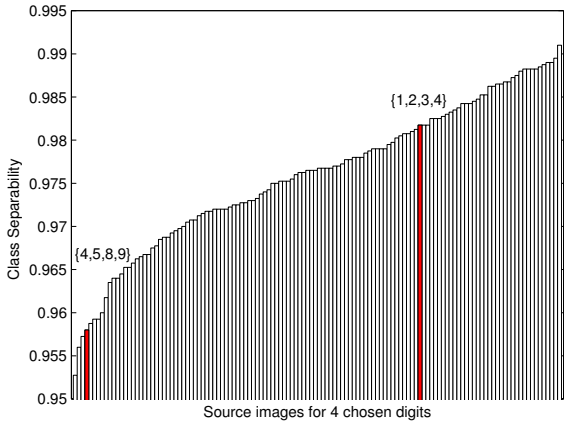


Fig. 5: Class separability values for all combinations of 4 classes chosen from the handwritten digit corpus.

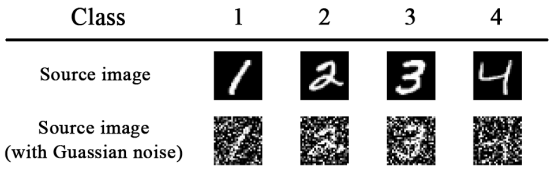


Fig. 6: Distortion of images using Gaussian noise.

TABLE III: Node classification results for the Biology article corpus using digits {4,5,8,9} and perturbed images of digits {1,2,3,4} as source image corpus.

Class	SVM	Images for {4589}		Noisy images for {1234}	
		$\text{Worker}_{\text{avg}}$	Surrogate-CM	$\text{Worker}_{\text{avg}}$	Surrogate-CM
	F measure				
Zoology	0.5639	0.6065	0.5906	0.5642	0.6103
Cell Biology	0.7250	0.8165	0.8592	0.7519	0.7701
Anatomy	0.6267	0.6772	0.6951	0.6441	0.6640
Genetics	0.6285	0.6425	0.6937	0.6608	0.7457
All Classes	0.6360	0.6857	0.7097	0.6553	0.6975
	Accuracy				
All Classes	0.6350	0.6919	0.7175	0.6580	0.6975

Next, we repeated the node classification experiment on the Wikipedia biology corpus using source images from digits {4,5,8,9}. The results are shown in Table III. The average accuracy of the workers is 69.19% while SURROGATE-CM increases the accuracy to 71.75%. Both values are lower than the previous results using digits {1,2,3,4} as source images, though they are still higher than the accuracy for baseline SVM. This result shows that class separability is an important criterion for choosing the source image corpus.

Clarity of Source Images Another criteria for choosing the source corpus is clarity of the individual images. To validate this, we repeated the node classification experiment on the Wikipedia biology corpus using perturbed images of digits {1,2,3,4}, as shown in Figure 6. The class separability value decreases from 0.9818 to 0.8113 after adding Gaussian noise to the source images. We presented the perturbed images selected by surrogate mapping to the workers for labeling. The results shown in Table III suggest that, although the classification accuracy is still higher than the baseline SVM, it is much lower than the original accuracy when the images were not distorted. This shows the importance of using images that are clear as the source domain for surrogate learning. Another point worth noting is that as the noise level increases, the images become harder for humans to label. In the worst case scenario, all images will be classified as noise by the workers, which means no new labeled examples will be augmented to the training set. In this case, the performance of surrogate learning should be equivalent to the baseline SVM.

VII. Related Work

Crowdsourcing is a distributed problem-solving model that aims to outsource costly or time consuming tasks to an undefined large group of individuals known as the crowd. The group of individuals who execute the task provided their services in exchange for micro-payments, social recognition, or for their own personal satisfaction. Crowdsourcing has been successfully employed to annotate data from Twitter streams [1], news stories [10], [11], and videos [13]. Crowdsourcing has also been adapted to active learning [8] and interactive learning [3] frameworks. However, there has been very little work on applying crowdsourcing to annotate network data for node classification and link prediction tasks.

The surrogate learning framework proposed in this study is an *out-of-domain feature transformation* approach, involving two distinct domains (*source* and *target*). Our target domain corresponds to a network whereas the source domain is an image corpus. This work departs from previous research on transfer learning [12] and domain adaptation [6], [4], which assume that the source and target domains share some commonalities in terms of their attributes, classes, or underlying structures. Instead, we consider the situation where each domain has a unique set of attributes, classes, and probability distributions.

VIII. Conclusions

This paper presents a novel framework to convert network data into images to enable the use of crowdsourcing technology. Experimental results demonstrate its effectiveness in producing visual images that can be easily annotated by the crowd. Despite its promise, the framework can still be improved in several ways. First, a more sophisticated algorithm [7] can be used to generate the consensus labels. The effect of noisy labels provided by unreliable workers also need to be investigated. Finally, the framework can be extended to consider a nonlinear transformation between the source and target domains.

References

[1] O. Alonso, C. Carson, D. Gerster, X. Ji, and S. U. Nabar. Detecting Uninteresting Content in Text Streams. In *Proc. of the ACM SIGIR 2010 on Crowdsourcing for Search Evaluation*, pages 39–42.

[2] S. Banerjee and A. Roy. *Linear Algebra and Matrix Analysis for Statistics*. CRC Press, 2014.

[3] A. Bernstein and J. Li. From active towards interactive learning: using consideration information to improve labeling correctness. In *Proc. of the Workshop on Human Computation*, pages 40–43, 2009.

[4] M. Chen, K. Q. Weinberger, and J. Blitzer. Co-training for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 2456–2464, 2011.

[5] R. Das and M. Vukovic. Emerging theories and models of human computation systems: a brief survey. In *Proc. of the Workshop on Ubiquitous Crowdsourcing*.

[6] H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. *JAIR*, 26(1):101–126, 2006.

[7] R. Gomes, P. Welinder, A. Krause, and P. Perona. Crowdclustering. In *NIPS*, pages 558–566, 2011.

[8] T. Graepel. The Smarter Crowd: Active Learning, Knowledge Corroboration, and Collective IQs. In *Proc of the Workshop on Crowdsourcing for Search and Data Mining*, 2011.

[9] Y. Le Cun, P. Haffner, L. Bottou, and Y. Bengio. Object recognition with gradient-based learning. In *Feature Grouping*. Springer Verlag, 1999.

[10] R. McCreadie, C. Macdonald, and I. Ounis. Crowdsourcing a News Query Classification Dataset. In *Proc. of the Workshop on Crowdsourcing for Search Evaluation*, pages 31–38.

[11] R. McCreadie, C. Macdonald, and I. Ounis. Crowdsourcing Blog Track Top News Judgments at TREC. In *Proc. of the Workshop on Crowdsourcing for Search and Data Mining*, pages 23–26.

[12] S. J. Pan and Q. Yang. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2010.

[13] M. Soleymani and M. Larson. Crowdsourcing for Affective Annotation of Video: Development of a Viewer-reported Boredom Corpus. In *Proc. of the Workshop on Crowdsourcing for Search Evaluation*, pages 4–8.