An Iterative Learning Algorithm for Within-Network Regression in the Transductive Setting

Annalisa Appice, Michelangelo Ceci, and Donato Malerba

Dipartimento di Informatica, Università degli Studi di Bari
via Orabona, 4 - 70126 Bari - Italy
{appice,ceci,malerba}@di.uniba.it

Abstract. Within-network regression addresses the task of regression in partially labeled networked data where labels are sparse and continuous. Data for inference consist of entities associated with nodes for which labels are known and interlinked with nodes for which labels must be estimated. The premise of this work is that many networked datasets are characterized by a form of autocorrelation where values of the response variable in a node depend on values of the predictor variables of interlinked nodes. This autocorrelation is a violation of the independence assumption of observation. To overcome to this problem, the lagged predictor variables are added to the regression model. We investigate a computational solution for this problem in the transductive setting, which asks for predicting the response values only for unlabeled nodes of the network. The neighborhood relation is computed on the basis of the node links. We propose a regression inference procedure that is based on a co-training approach according to separate model trees are learned from both attribute values of labeled nodes and attribute values aggregated in the neighborhood of labeled nodes, respectively. Each model tree is used to label the unlabeled nodes for the other during an iterative learning process. The set of labeled data is changed by including labels which are estimated as confident. The confidence estimate is based on the influence of the predicted labels on known labels of interlinked nodes. Experiments with sparsely labeled networked data show that the proposed method improves traditional model tree induction.

1 Introduction

A data network (also called networked data) consists of entities, generally of the same type such as web-pages or telephone accounts, which are associated with the nodes of the network and which are interlinked with other nodes via various explicit relations (or edges) such as hyperlinks between web-pages or people calling each other. Over the past few years data networks such as sensor networks, communication networks, financial transaction networks and social networks have become ubiquitous in everyday life. This ubiquity of data networks motivates the recent focus of research in data mining to extend traditional inference techniques in order to learn in data networks.

Several issues challenge the task of learning in network data, the most important being the consideration of various forms of *autocorrelation* which may affect data networks. Different definitions of autocorrelation are in use depending on the field of study which is being considered and not all of them are equivalent. Here autocorrelation is defined as the property that a value observed at a node depends on the values observed at neighboring nodes in the network. Autocorrelation has been justified in several ways, such as Tobler's first law of geography [20] and the homophily's principle [14], according to the specific application domain.

The major difficulty due to the autocorrelation is that the independence assumptions, which typically underlies machine learning methods, are no longer valid. For example, the violation of the instance independence has been identified as the main responsible of poor performance of traditional machine learning methods [16]. To remedy the negative effects of the violation of independence assumptions, autocorrelation has to be explicitly accommodated in the learned models.

In predictive models, where response variables depend on both predictor variables and an error term, autocorrelation can be expressed in three different ways, by correlating: 1) the error terms of neighboring nodes; 2) the response variables of neighboring nodes; 3) the response variable with the predictor variables of neighboring nodes. In spatial data analysis, these three types of predictive models are respectively known as spatial error models, spatial lag models and spatial cross-regressive models [18]. As observed in [2], the first two types of models are *global* in scope, in the sense that an error term or a dependent variable at a location (node of a network) has a spillover effect on all other locations, while cross-regressive models are *local* in scope, since the effects are confined to the neighbors of each observation. In spatial data analysis, cross-regressive models make more sense from a theoretical point of view [1] and present the additional advantage of being easier to use. In this paper we face the problem of learning predictive (regression) models in data networks and we deal with the autocorrelation issue by considering cross-regressive models.

The consideration of partially labeled data networks, where labeled entities are possibly interlinked to unlabeled entities and vice-versa, adds a further degree of complexity, since it is difficult *to separate data into training and test sets.* Indeed, labeled data would serve as training data and subsequently as background knowledge necessary for labeling entities in the test set. This consideration motivates the investigation of the learning problem in a setting different from the classical inductive one, where the prediction model is built by considering only a finite set of labeled data (training set) and it is then used to make prediction on any possible instance. In this work, we consider the *transductive* setting [21], where both labeled and unlabeled data are used to build the model and predictions are confined to unlabeled data available when learning starts. More precisely, the idea behind *transductive inference* (or *transduction*) is to analyze both the labeled data L and the unlabeled data U to build a model which predicts (exlusively) data in U as accurately as possible. Therefore, in the transductive setting, difficulties due to the separation of training and test set are overcome.

The data mining task considered in this work is *transductive within-network regression*, which is a variant of the classification task recently investigated in [11] for categorical labels. Given a fully described network (nodes and edges) for which continuous labels are provided for only some of the nodes, the goal is to determine labels of the rest of the nodes in the network. We propose a learning algorithm, named ITL (Iterative Transductive Learner), which capitalizes on the strengths of both model tree induction and transductive learning to effectively solve the given problem when labels of data networks are originally sparse and possibly scarce. The specific contributions of this work are highlighted as follows:

1. The combination of iterative transductive learning with the co-training paradigm in order to both generate cross-regressive models and bootstrap from a small set of labeled training data via a large set of unlabeled data.
2. Prediction of continuous labels is based on model trees [3], which do not impose any *a priori* global structure (e.g., linear) of the regression surface. Model trees are build on two views of data, as required by the co-training paradigm. Each model tree labels the unlabeled data for the other during the learning process.
3. The use of co-training paradigm allows us to learn two different model trees: a model tree that identifies the correlation between the label of an entity and the attribute values of the same entity and a model tree that identifies the correlation between the label of an entity and the attribute values in the neighborhood of the entity. Each model tree labels the unlabeled data for the other during the learning process.
4. We present some procedures to estimate the confidence of predicted label(s) through consulting the influence of the labeling of unlabeled entities in a model tree based re-prediction of the labeled entities which are interlinked to the unlabeled one(s).
5. We demonstrate that our approach is robust to both sparse labeling and low label consistency, performing well consistently across a range of data network where traditional model tree induction fails.

The rest of the paper is organized as follows. In Section 2, we review related work. We present the formal definition of the task in Section 3 and our proposed method in Section 4. Section 5 describes the experimental methodology and results. Finally, Section 6 concludes the work.

2 Related Work

Regression inference in data network is still a challenging issue in machine learning and data mining. Due to the recent efforts of various researchers, numerous algorithms have been designed for modeling a partially labeled network and providing estimates of unknown labels associated with nodes. Anyway, at the best of our knowledge, these algorithms address the prediction problem only in the classification case, that is, when labels are categorical.

Currently, the main research in this area is in the thrust of network learning and graph mining. Network learning assumes that data for inference are already in the form of a network and exploits the structure of the network to allow the collective inference. Collective inference allows to infer various interrelated values simultaneously. It is used in network learning since it permits to estimate neighboring labels which influence one another [12,9,19]. Since exact inference is known to be an NP-hard problem and there is no guarantee that data network satisfy the conditions that make exact inference tractable for collective learning, most of the research in collective learning has been devoted to the development of approximate inference algorithms.

Some of the popular approximate inference algorithms are the iterative inference, the Gibbs sampling, the loopy belief propagation and the mean-field relaxation labeling. An outline of strengths and weakness of these algorithms is reported in [19]. In general, one of the major advantages of collective learning lies in its powerful ability to learn various kinds of dependency structures (positive vs. negative autocorrelation, different degrees of correlation and so on) [10]. However, as pointed out in [15], when the labeled data is very sparse, the performance of collective classification might be largely degraded due to the lack of sufficient neighbors. This is overcome by incorporating informative “ghost edges” into the networks to deal with sparsity issues [13,15].

Interestingly learning problems similar to the tasks addressed in network learning have been recently addressed outside the areas of network learning and graph mining. This second area of work has not been cast as a network learning problem, but rather in the area of semi-supervised learning in a transductive setting [21] where a corpus of data without links is given. The basic idea is to connect data into a weighted network by adding edges (in various ways) based on the similarity between entities and to estimate a function on the graph which guarantees the consistency with the label information and the smoothness over the whole graph [23]. The constraint on smoothness implicitly assumes positive autocorrelation in the graph, that is, nearby nodes tend to share the same class labels (i.e., homophily).

A prominent achievement in semi-supervised learning is represented by the co-training paradigm [4] where independent views, i.e., distinct sets of attributes, of labeled and unlabeled data are available for deriving separate learners. Predictions of each learner of unlabeled data are then used to augment the training set of the other within an iterative learning process. Co-training is already used to design regression algorithms in semi-supervised learning. Brefeld et al. [5] use co-training to formulate a semi-supervised least square regression algorithm, where co-training is casted as a regularized risk minimization problem in Hilbert spaces. Several data views are obtained for learning from different instance descriptions, views, and/or different kernel functions. Zhou and Li [22] apply co-training to learn k-NN regression by adopting a single attribute set but considering distinct distance measures for the two hypotheses.

3 Problem Definition and Notations

A network is a set of entities connected by edges. Each entity is called node of the network. A number (which is usually taken to be positive) called weight is associated with each edge. In a general formulation, a network can be represented as a (weighted) graph that is a set of nodes and a ternary relation which represent both the edges between nodes and the weight associated to each edge. Formally,

Definition 1 (Data Network). *A data network N is a pair (V, E), where:*

1. *V is a set of nodes, and*
2. *E is a set of weighted edges between nodes, that is,*

$$E = \{\langle u, v, w\rangle | u, v \in V, w \in \mathbb{R}^+\}.$$

In this work, each node of the network is associated with a data observation $(\mathbf{x}, y) \in \mathbf{X} \times Y$. $\mathbf{X}$ is a feature space spanned by m predictor variables X_i with $i = 1, \ldots, m$ while Y is the possibly unknown response variable (or label) with a range in $\mathbb{R}$. Additionally, labels are typically sparse in the network, that is, nodes for which labels are known may be interlinked with nodes for which labels must be estimated. In several real cases, labels are also scarce since the manual annotation of large data sets can be very costly. In this data context, the problem of regression consists in predicting the labels of unlabeled nodes as accurate as possible. The regression problem is formulated in network learning as follows.

Given:

1. the labeled node set $L \subset \mathbf{X} \times Y$;
2. the projection of the unlabeled (working) node set $U = V - L$ on $\mathbf{X}$;
3. the ternary relation $E \subset V \times V \times \mathbb{R}^+$;
4. the neighborhood function $\eta_E : V \longmapsto 2^{V \times \mathbb{R}^+}$ such that:

$$\eta_E(u) = \{(v_1, w_1), \ldots, (v_k, w_k)\} \text{ with } (u, v_i, w_i) \in E, i = 1 \ldots k$$

Find an estimate $\hat{Y}$ for the unknown value of response variable Y for each node $u \in U$ such that $\hat{y_u}$ is as accurate as possible.

An algorithmic solution to this problem is reported in the next Section. The learner receives full information (including labels) on the nodes of L and partial information (without labels) on the nodes of U as well as weighted edges in E and is asked to predict the labels of the nodes of U. The algorithm is formulated in the original distributional-free transductive setting [21] and requires that both L and U are sampled from the node set V without replacement. This means that, unlike the standard inductive setting, the nodes in the labeled (and unlabeled) set are supposed to be mutually dependent based on the existence of a link (transitively) connecting them. Vapnik introduced an alternative transductive setting which is distributional, since both T and W are assumed to be drawn independently and identically from some unknown distribution. As shown in [21](Theorem 8.1), error bounds for learning algorithms in the distribution-free setting apply to the more popular distributional transductive setting. This justifies our focus on the distributional-free setting.

Algorithm 1. Top-level description of the Iterative Transductive Learner in Co-training style.

```
1: **ITL**(L, U, E)
2: **Input**
3: the labeled node set $L \subset \mathbf{X} \times Y$;
4: the projection of the unlabeled node set $U = V - L$ on $\mathbf{X}$;
5: the ternary relation $E \subset V \times V \times \mathbb{R}^+$;
6: **Output**
7: an estimate of unknown labels of nodes in $U$;
8: **begin**
9: $\overline{L} \leftarrow$ laggedPredictorVariables$(L, V, E)$; $\overline{U} \leftarrow$ laggedPredictorVariables$(U, V, E)$;
10: $L_0 \leftarrow L$; $U_0 \leftarrow U$; $L_1 \leftarrow \overline{L}$; $U_1 \leftarrow \overline{U}$;
11: $i \leftarrow 1$;
12: **repeat**
13:    $change \leftarrow$ false;
14:    $t_0 \leftarrow$learn$(L_0)$; $t_1 \leftarrow$learn$(L_1)$;
15:    $P_0 \leftarrow$predictConfidentLabels$(t_0, U_0, L_0, E)$; $P_1 \leftarrow$predictConfidentLabels$(t_1, U_1, L_1, E)$;
16:    **if** $P_0 \neq \oslash$ or $P_1 \neq \oslash$ **then**
17:       $change \leftarrow$ true;
18:       **for** $e \in P_0$ **do**
19:          $L_1 \leftarrow L_1 \cup \{\langle$instance$(e, U_1), \hat{y_e}\rangle\}$; $U_1 \leftarrow U_1 - \{$instance$(e, U_1)\}$;
20:       **end for**
21:       **for** $\overline{e} \in P_1$ **do**
22:          $L_0 \leftarrow L_0 \cup \{\langle$instance$(\overline{e}, U_0), \hat{y_{\overline{e}}}\rangle\}$; $U_0 \leftarrow U_0 - \{$instance$(\overline{e}, U_0)\}$;
23:       **end for**
24:    **end if**
25: **until** not$(++i \geq$ MAX_ITERS AND $change)$;
26: $U \leftarrow$label$(t_0, L, U, t_1, \overline{L}, \overline{U})$;
27: **end**
```

4 The Algorithm

The Iterative Transductive Learner (ITL) addresses the problem of predicting the unknown continuous labels of nodes which are distributed in a sparsely labeled network. ITL iteratively induces two distinct model trees in a co-training style. Labels predicted from one model tree which are estimated as confident extend the labeled set to be used by the other model tree learner for the next iteration of the learning process. The model trees which are induced in the last iteration of ITL are used to predict the unknown network labels. Details on the learning in co-training style, the evaluation of the confidence of predicted labels and the labeling of unlabeled nodes are reported in the next subsections.

4.1 Iterative Learning in Co-training Style

The top level description of ITL is reported in Algorithm 1. Let $N(V, E)$ be the sparsely labeled network whose unknown labels have to be predicted, ITL takes as input: the attribute-value observations (with labels) associated with the labeled node set $L \subset V$, the attribute-value observations (without labels) associated with the unlabeled node set $U \subset V$ ($U = V - L$), and the relation E, and predicts the unknown labels for the nodes of U. ITL is iterative and keeps with the main idea of co-training by inducing at each iteration, two distinct regression models from different views of the attribute-value data associated with the node set. The former is a regression model which includes the predictor variables measured at the currently labeled nodes of the network, the latter is a cross-regressive model which includes the *lagged* predictor variables measured in the neighborhood of the nodes.

In Algorithm 1, the function $laggedPredictorVariables()$ is in charge of constructing $\overline{L}$ ($\overline{U}$) that is the lagged view of the data associated with the node set L (U). This lagged view of a dataset is obtained by projecting data over the lagged predictor variables in place of the original predictor variables. Formally,

Definition 2 (Lagged variable). *Let $N = (V, E)$ be a network and X be a variable that is measured at the nodes of V. For each node $u \in V$, the lagged variable $\overline{X}$ is assigned with the aggregate of the values which are measured for X at nodes falling in the neighborhood $\eta_E(u)$.*

In particular, by considering the case that X_i is continuous, then,

$$\overline{x_{i_u}} = \frac{\sum\limits_{(v,w)\in\eta_E(u)} (x_{i_v} \times w)}{\sum\limits_{(v,w)\in\eta_E(u)} w}, i = 1 \dots m. \quad (1)$$

Further extension of ITL would allow to deal with discrete predictor variables.

By initially assigning $L_0 = L, U_0 = U, L_1 = \overline{L}$ and $U_1 = \overline{U}$, ITL learns the regression model t_0 from L_0 and the regression model t_1 from L_1 (see the function $learn()$ in Algorithm 1). The basic learner employed to induce both t_0 and t_1 is the model tree learner presented in [3]. The choice of a model tree learner is motivated by the capability of model trees of do not imposing any a-priory defined global form of regression surface, but assuming a functional form at local level. t_0 and t_1 are then used to predict the unknown labels ($\widehat{y}$) of the nodes falling in U_0 and U_1, respectively.

Labels which are confidently predicted (see function $predictConfidentLabels()$ in Algorithm 1) are assigned to the corresponding nodes in U_1 (U_0). New labeled nodes are then moved from U_1 to L_1 (U_0 to L_0). The function $instance()$ is in charge of passing from the original data view to the lagged data view of a node, and vice-versa. In particular, if u belongs to L (U), $instance(u, \overline{L})$ ($instance(u, \overline{U})$) returns $\overline{u}$ in $\overline{L}$ ($\overline{U}$). Similarly, if $\overline{u}$ belongs to $\overline{L}$ ($\overline{U}$), instance($\overline{u}, L$) (instance($\overline{u}, U$)) returns the corresponding u in L (U).

The learning process stops when the maximum number of learning iterations, MAX_ITERS, is reached, or there is no unlabeled node which is confidently moved from the unlabeled set to the labeled set. Model trees which are learned in the last iteration of the learning process are used to definitely label working observations (see function $label()$ in Algorithm 1). Details of $predictConfidentLabels()$ and $label()$ are provided in the next subsections.

Algorithm 2. Predict and estimate confidence of labels according to Single Label Confidence Estimate.

```
1: determineConfidentlyLabeledNodes($(U_i, L_i, E, t_i) \Longrightarrow P_i$)
2: Input
3: the node set $U_i \subset \mathbf{X} \times Y$ and the node set $L_i \subset \mathbf{X} \times Y$;
4: the ternary relation $E \subset U_i \cup L_i \times U_i \cup L_i \times \mathbb{R}^+$;
5: the model tree $t_i$ induced from $L_i$;
6: Output
7: $P_i \subseteq U_i$ such that $P_i$ includes only the nodes of $U_i$ whose predicted labels are
   estimated as confident;
8: begin
9: $P_i \leftarrow \oslash$;
10: for $u$ in $U_i$ do
11:    $\hat{y_u} \leftarrow$response$(t_i, u)$;
12:    $t'_i \leftarrow$learn$(L_i \cup \{\langle u, \hat{y_u}\rangle\})$;
13:    $pos \leftarrow 0$; $neg \leftarrow 0$;
14:    for $v$ in $\eta_E(u)|_{L_i}$ do
15:       if $(y_v$-response$(t'_i, v))^2$-$(y_v$-response$(t_i, v))^2 \geq 0$ then
16:          $pos \leftarrow pos + 1$;
17:       else
18:          $neg \leftarrow neg + 1$;
19:       end if
20:    end for
21:    if $pos \geq neg$ then
22:       $P_i \leftarrow P_i \cup \{\langle u, \hat{y_u}\rangle\}$;
23:    end if
24: end for
25: end
```

4.2 Evaluating the Confidence of Predicted Labels

A model tree is used to predict the unknown labels in the network. The confidence of each estimated label is evaluated in order to identify the most confident labels. Intuitively, confident labels are with the following property. The error performed by a model tree in re-predicting the labeled node set should decrease the most if the most confidently labeled nodes are utilized in the learning process. According to this property, we have designed two alternative mechanisms, named Single Label Confidence Evaluation (SLCE) and Multi Label Confidence Evaluation (MLCE), which provide an estimate of the confidence of the labels which are predicted in ITL.

Single Label Confidence Evaluation

The SLCE estimates the confidence of predicted labels one by one (see Algorithm 2). The confidence is estimated by a model tree that is learned from a training set consisting of the nodes which are currently labeled in the network and the unlabeled node whose predicted label has to be estimated. The confidence of

this label corresponds to the confidence of this model tree in re-predicting the labeled nodes which are interlinked (as neighbors) to the unlabeled one.
Formally, let:

1. t_i (with $i = 0, 1$) be the model tree induced from the labeled node set L_i,
2. $u \in U_i$ be a node falling in the unlabeled set U_i (with $i = 0, 1$),
3. $\hat{y_u}$ the label predicted from t_i for u,
4. $\eta_E(u)|_{L_i} = \{v \in L_i | (u, v, w) \in E\}$ be the set of labeled nodes v which are neighbors of u in L_i according to E, and
5. t'_i (with $i = 0, 1$) be the model tree induced from $L_i \cup \{\langle u, \hat{y_u}\rangle\}$,

the confidence of $\hat{y_u}$ is evaluated according to the influence of $\hat{y_u}$ on the known labels of nodes falling in $\eta_E(u)|_{L_i}$. In particular, for each $v \in \eta_E(u)|_{L_i}$, ϵ_v is the result of subtracting the squared error performed by t'_i in determining the label of v from the squared error performed by t_i in determining the label of v,

$$\epsilon_v = (y_v - \text{response}(t'_i, v))^2 - (y_v - \text{response}(t_i, v))^2 \tag{2}$$

with y_v be the response that originally labels v in L_i at the current iteration of ITL. The function $response(t_i, v)$ returns the label predicted for v by t_i, while $response(t'_i, v)$ returns the label predicted for v by t'_i, respectively.
By defining:

$$Pos = |\{v \in \eta_E(u)|_{L_i} | \epsilon_v \geq 0\}| \quad Neg = |\{v \in \eta_E(u)|_{L_i} | \epsilon_v < 0\}|, \tag{3}$$

with $|\cdot|$ the cardinality of a set $\cdot$, the label $\widehat{y_v}$ is estimated as confident if $Pos \geq Neg$, un-confident otherwise.

Multi Label Confidence Evaluation

The MLCE firstly groups unlabeled nodes of the network in possibly overlapping clusters and then estimates the confidence of the entire *cluster* of predicted labels, cluster by cluster (see Algorithm 3). For each labeled node $v \in L_i$, a cluster, $\eta_E(v)|_{U_i}$ is constructed by including the unlabeled neighbors of v which are determined in U_i according to E. By using t_i to assign a label to the nodes falling $\eta_E(v)|_{U_i}$, the labeled node set $\widehat{\eta_E(v)|_{U_i}}$ is constructed from $\eta_E(v)|_{U_i}$ (see function $assignLabel()$ in Algorithm 3), as follows:

$$\widehat{\eta_E(v)|_{U_i}} = \{\langle u, \text{response}(t_i, u)\rangle | u \in \eta_E(v)|_{U_i}\} \tag{4}$$

where $response(t_i, u)$ is the label predicted for u by t_i. Let t'_i be the model tree induced from $L_i \cup \widehat{\eta_E(v)|_{U_i}})$, predicted labels of $\widehat{\eta_E(v)|_{U_i}}$ are estimated as confident iff:

$$(y_v - \text{response}(t'_i, v))^2 - (y_v - \text{response}(t_i, v))^2 \geq 0, \tag{5}$$

un-confident, otherwise.

Algorithm 3. Predict and estimate confidence of labels according to Multi Label Confidence Estimate.

```
1: determineConfidentlyLabeledNodes($U_i, L_i, E, t_i$)$\Longrightarrow P_i$
2: Input
3: the node set $U_i \subset \mathbf{X} \times Y$ and the node set $L_i \subset \mathbf{X} \times Y$;
4: the ternary relation $E \subset U_i \cup L_i \times U_i \cup L_i \times \mathbb{R}^+$;
5: the model tree $t_i$ induced from $L_i$;
6: Output
7: $P_i \subseteq U_i$ such that $P_i$ includes only the nodes of $U_i$ whose predicted labels are
   estimated as confident;
8: begin
9: $P_i \leftarrow \oslash$;
10: for $v$ in $L_i$ do
11:     $\widehat{\eta_E(v)|_{U_i}} \leftarrow$assignLabel($T_i, \eta_E(v)|_{U_i}$);
12:     $t'_i \leftarrow$learn($L_i \cup \widehat{\eta_E(v)|_{U_i}}$);
13:     if (response($t'_i, v$)-y($v$))$^2$-(response($t_i, v$)-y($v$))$^2 \geq 0$ then
14:         $P_i \leftarrow P_i \cup \widehat{\eta_E(v)|_{U_i}}$;
15:     end if
16: end for
17: end
```

4.3 Predicting the Unlabeled Nodes in the Network

Model trees t_0 and t_1 which are learned in the last iteration of ITL are used to predict the final labels $\hat{Y}$ to be associated with originally unlabeled nodes of U. Let $\overline{u} \in \overline{U}$ be the lagged data view of the unlabeled node $u \in U$, then:

$$\hat{y_u} = \frac{\omega_0 \times \text{response}(t_0, u) + \omega_1 \times \text{response}(t_1, \overline{u})}{\omega_0 + \omega_1} \text{ with } u \in U. \tag{6}$$

where ω_0 and ω_1 are computed on the basis of the mean square error (mse) of each model tree on the original labeled set (L and $\overline{L}$, respectively). Details are provided in Algorithm 4.

5 Experiments

We demonstrate that ITL is robust to both sparse labeling and low label consistency and it improves traditional model tree induction across a range of several geographical data networks.

Dataset Description

GASD (USA Geographical Analysis Spatial Dataset) [17] contains 3,107 observations on USA county votes cast in 1980 presidential election. Specifically, it contains the total number of votes cast in the 1980 presidential election per county (response attribute), the population in each county of 18 years of age

Algorithm 4. Assigning a final label to the unlabeled nodes of the network.

```
1: **label**$(t_0, L, U, t_1, \overline{L}, \overline{U})$
2: **Input**
3: the model tree $t_0$ induced on the feature space $\mathbf{X}$;
4: the set $L \subseteq \mathbf{X} \times Y$;
5: the unlabeled set $U \subseteq \mathbf{X}$;
6: the model tree induced on the feature space $\overline{\mathbf{X}}$;
7: the labeled set $\overline{L} \subseteq \overline{\mathbf{X}} \times Y$;
8: the unlabeled set $\overline{U} \subseteq \overline{\mathbf{X}}$;
9: **Output**
10: $U' = \{\langle u, \hat{y_u}\rangle | u \in U, \hat{y_u}$ is the final label predicted for $u\}$ ;
11: **begin**
12: $U \leftarrow \oslash$;
13: $m_0 \leftarrow \mathrm{mse}(t_0, L)$; $m_1 \leftarrow \mathrm{mse}(\ t_1, \overline{L})$;
14: **if** $m_0 > m_1$ **then**
15:    $\omega_0 \leftarrow 1$; $\omega_1 \leftarrow m0/m1$;
16: **else**
17:    $\omega_0 \leftarrow m1/m0$; $\omega_1 \leftarrow 1$;
18: **end if**
19: **for** $u \in U$ **do**
20:    $\overline{u} \leftarrow \mathrm{instance}(u, \overline{U})$; $\hat{y_u} \leftarrow \frac{\mathrm{response}(t_0,u)\times\omega_0+\mathrm{response}(t_1,\overline{u})\times\omega_1}{\omega_0+\omega_1}$;
21:    $U' \leftarrow U \cup \{\langle u, \hat{y_u}\rangle\}$;
22: **end for**
23: **end**
```

or older, the population in each county with a 12th grade or higher education, the number of owner-occupied housing units, the aggregate income, the XCoord and YCoord spatial coordinates of the county. **Forest Fires** is public available for research at UCI Machine Learning Repository[1]. The details are described in [7]. It collects 512 forest fire observations from the Montesinho natural park in the northeast region of Portugal. The data, collected from January 2000 to December 2003, include the burned area of the forest in ha[2] (response variable), the Fine Fuel Moisture Code (FFMC), the Duff Moisture Code (DMC), the Drought Code (DC), the Initial Spread Index (ISI), the temperature in Celsius degrees, the relative humidity, the wind speed in km/h, the outside rain in mm/m^2, the XCoord and YCoord spatial coordinates within the Montesinho park map. **NWE** (North-West England) contains census data collected in the European project SPIN![3]. Data are census data concerning North West England area that is decomposed into censual sections or wards for a total of 1011 wards. Census data provided by 1998 Census is available at ward level. We consider percentage of mortality (response variable) and measures of deprivation level in the ward according to index scores such as, Jarman Underprivileged Area Score, Townsend score, Carstairs score and the Department of the Environments Index,

[1] http://archive.ics.uci.edu/ml/
[2] 1ha/100 = 100 m^2.
[3] http://www.ais. fraunhofer.de/KD/SPIN/project.html

the XCoord and YCoord spatial coordinates of the ward centroid. By removing observations including null values, only 979 observations are used in this experiments. Finally, **Sigmea-Real** [8] collects 817 measurements of the rate of herbicide resistance of two lines of plants (response variables), that is, the transgenic male-fertile (MF) and the non-transgenic male-sterile (MS) line of oilseed rape. Predictor variables of this study are the cardinal direction and distance from the center of the donor field, the visual angle between the sampling plot and the donor field, and the shortest distance between the plot and the nearest edge of the donor field, the XCoord and YCoord spatial coordinates of the plant.

Experimental Setting

Each geo-referenced dataset D is mapped into a data network $N = (V, E)$ that includes a node $u \in V$ for each observation $(x_1, \ldots, x_n, y, xCoord, yCoord) \in D$ and associates u with $(x_1, \ldots, x_n, y)$. Let u and v be two distinct nodes in V, there is an edge from u to v labeled with w in N (i.e., $(u, v, w) \in E$) iff v is one of the k nearest neighbors of u. The Euclidean distance is computed to determine neighbors. Notice that the neighboring relation defined above is not necessarily symmetric, v may be a k nearest neighbor of u, but not necessarily vice-versa. Additionally, u is a neighbor of u. In this paper, several data networks are constructed from the same dataset by varying $k = 5, 10, 15$. They are denoted as N_5, N_{10} and N_{15}. In each data network, the weight w is defined according to a continuous function of Euclidean distance [6] as follows:

$$w = e^{-\frac{dist(u,v)^2}{b_u^2}} \text{ with } b_u = \max_{v \in k-nearestNeighbors(u,V)} dist(u,v), \tag{7}$$

If u and v are associated with observations taken at the same geographical site, the weighting of observations collected at this site would be unity. The weighting of other observations will decrease according to a Gaussian curve as the Euclidean distance between u and v increases.

For each data network, experiments are performed in order to: 1) validate the actual advantage of the iterative transductive learner over the basic model tree learner in labeling the unlabeled nodes of a sparsely and scarcely labeled network (ITL vs t_0 and ITL vs t_1), 2) evaluate the advantages of a co-training implementation in the transductive learning (ITL vs ITL*), and 3) compare performance of SLCE and MLCE in estimating the confidence of labels (SLCE vs MLCE). t_0 (t_1) denotes the model tree which is induced from the original set of labeled data by considering the predictor variables (lagged predictor variables) only, ITL* is the iterative transductive learner without co-training, that is, no lagged view of data is considered in the learning process, ITL is the iterative transductive learner with co-training.

The empirical comparison is based on the mean square error (MSE) that is estimated according to a K-fold cross validation. K is set to 10 in experiments performed with GASD dataset, and $K = 5$ in experiments performed with Forest Fires, NWE and Sigmea Real. For each trial, algorithms to be compared are trained on a single fold and tested on the hold-out K - 1 folds, which form the working set. The comparative statistics is computed by averaging the MSE error over the K-folds (Avg.MSE). It is noteworthy that, unlike the standard cross-validation approach, here only one fold is used for the training set. In this way we can simulate datasets with a small percentage of labeled cases (the training set) and a large percentage of unlabeled data (the working set), which is the usual situation for a transductive learning. ITL is run with $MAX\_ITERS = 5$.

Results

The Avg. MSE performed by both the transductive learner and the inductive learner is reported in Table 1. Results suggest several conclusions. First, they confirm that ITL performs generally better than the basic model tree learners (ITL improves both t_0 and t_1 in accuracy) by profitably employing a kind of iterative learning to bootstrap from a small set of labeled training data via a large set of unlabeled data. The exception is represented by Sigmea Real (MF) that is the only dataset where the baseline inductive learner t_0 always outperforms ITL. Our justification is that the worse performance of ITL may depend on the fact that this dataset exhibits about 65% of observations which are labeled as zero which leads to a degradation of both predictive capability of the learner that operates with the aggregate data view in the co-training and capability of identifying confident labels. This is confirmed by the fact accuracy of

Table 1. Avg.MSE: Inductive learners (t0 and t1) vs. iterative transductive learner without co-training (ITL*) and with co-training (ITL)

Avg.MSE		N_5		N_{10}		N_{15}	
		SLCE	MLCE	SLCE	MLCE	SLCE	MLCE
GASD	t0	0.15174	0.15174	0.15174	0.15174	0.15174	0.15174
	t1	0.15879	0.15879	0.17453	0.17453	0.17419	0.17419
	ITL*	0.13582	0.13239	0.13643	0.13387	0.13606	0.13468
	ITL	**0.13006**	**0.12965**	**0.13156**	**0.13162**	**0.13387**	**0.13128**
Forest Fires	t0	81.16599	81.16599	81.16599	81.16599	81.16599	81.16599
	t1	64.68706	64.68706	64.71256	64.71257	64.80331	64.80332
	ITL*	81.45897	82.33336	81.04362	74.48984	81.30787	80.89551
	ITL	**64.44121**	**63.88140**	**64.73077**	**64.28176**	**63.92594**	**64.41118**
NWE	t0	0.00255	0.00255	0.00255	0.00255	0.00255	0.00255
	t1	0.00250	0.00250	0.00252	0.00252	0.00256	0.00256
	ITL*	0.00253	0.00252	0.00254	0.00258	0.00252	0.00253
	ITL	**0.00245**	**0.00244**	**0.00248**	**0.00248**	**0.00247**	**0.00248**
SigmeaReal (MF)	t0	2.35395	2.35395	2.35395	2.35395	2.35395	2.35395
	t1	2.57045	2.57045	2.51061	2.51061	2.51991	2.51991
	ITL*	2.36944	2.36336	2.36579	2.35532	2.37024	2.35966
	ITL	**2.44036**	**2.43278**	**2.40851**	**2.42544**	**2.46547**	**2.43397**
SigmeaReal(MS)	t0	5.87855	5.87855	5.87855	5.87855	5.87855	5.87855
	t1	5.80157	5.80157	6.12569	6.12569	6.08601	6.08601
	ITL*	5.87364	5.87389	5.87374	5.87781	5.87346	5.87340
	ITL	**5.75021**	**5.61658**	**5.85322**	**5.85275**	**5.74338**	**5.87403**

cross-regressive model tree t_1 is significantly worse than the accuracy of the classical model tree t_0. Second, the co-training improves the accuracy of the iterative transductive learner (ITL vs ITL*) by combining cross-regressive models with traditional regression models. Finally, the comparison between MLCE and SLCE suggests that the accuracy of ITL is improved by the use of MLCE when the data network includes nodes with a low number of neighbors ($k = 5$), while the accuracy of ITL is generally improved by the use of SLCE when the data network includes nodes with higher number of neighbors ($k = 15$).

6 Conclusions

In this paper we investigate the task of regression in labeled networked data where labels are sparse and continuous. We assume that data present a form of autocorrelation where the value of the response variable in a node depends on the values of the predictor variables of interlinked nodes. For this reason, we consider the contribution of lagged predictor variables in the regression model. We investigate a computational solution in the transductive setting, which asks for predicting the response values only for unlabelled nodes of the network. The neighborhood relation used in the transductive setting is computed on the basis of the node links. The solution is based on co-training, since separate model trees are learned from attribute values of labeled nodes and attribute values aggregated in the neighborhood of labeled nodes, respectively. Two distinct procedures have been proposed to evaluate confidence of the predicted labels. Experiments with several sparsely labeled networked data are performed. Results show that the proposed method improves accuracy of traditional model tree induction.

Acknowledgment

This work is fully supported by the Project "Scoperta di conoscenza in domini relazionali" funded by the University of Bari. The authors thank Saso Dzeroski for kindly providing Sigmea Areal dataset.

References

1. Abreu, M., de Groot, H., Florax, R.: Space and growth: A survey of empirical evidence and methods. Region and Development, 12–43 (2005)
2. Anselin, L.: Spatial externalities, spatial multipliers and spatial econometrics. International Regional Science Review (26), 153–166 (2003)
3. Appice, A., Dzeroski, S.: Stepwise induction of multi-target model trees. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 502–509. Springer, Heidelberg (2007)
4. Blum, A., Mitchell, T.M.: Combining labeled and unlabeled data with co-training. In: COLT, pp. 92–100 (1998)

5. Brefeld, U., Gärtner, T., Scheffer, T., Wrobel, S.: Efficient co-regularised least squares regression. In: Cohen, W.W., Moore, A. (eds.) 23th International Conference on Machine Learning, ICML 2006. ACM International Conference Proceeding Series, vol. 148, pp. 137–144. ACM, New York (2006)
6. Charlton, M., Fotheringham, S., Brunsdon, C.: Geographically weighted regression. In: ESRC National Centre for Research Methods NCRM Methods Review Papers NCRM/006 (2005)
7. Cortez, P., Morais, A.: A data mining approach to predict forest fires using meteorological data, pp. 512–523. APPIA (2007)
8. Demšar, D., Debeljak, M., Lavigne, C., Džeroski, S.: Modelling pollen dispersal of genetically modified oilseed rape within the field. In: Abstracts of the 90th ESA Annual Meeting, The Ecological Society of America, p. 152 (2005)
9. Gallagher, B., Tong, H., Eliassi-Rad, T., Faloutsos, C.: Using ghost edges for classification in sparsely labeled networks. In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008, pp. 256–264. ACM, New York (2008)
10. David, J., Jennifer, N., Brian, G.: Why collective inference improves relational classification. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2004, pp. 593–598. ACM, New York (2004)
11. Macskassy, S.A., Provost, F.: A Brief Survey of Machine Learning Methods for Classification in Networked Data and an Application to Suspicion Scoring. In: Airoldi, E.M., Blei, D.M., Fienberg, S.E., Goldenberg, A., Xing, E.P., Zheng, A.X. (eds.) ICML 2006. LNCS, vol. 4503, pp. 172–175. Springer, Heidelberg (2007)
12. Macskassy, S., Provost, F.: Classification in networked data: a toolkit and a univariate case study. Machine Learning 8, 935–983 (2007)
13. Macskassy, S.A.: Improving learning in networked data by combining explicit and mined links. In: Proceedings of the 22nd Conference on Artificial Intelligence, AAAI 2007, pp. 590–595. AAAI Press, Menlo Park (2007)
14. McPherson, M., Smith-Lovin, L., Cook, J.: Birds of a feather: Homophily in social networks. Annual Review of Sociology 27, 415–444 (2001)
15. Jennifer, N., David, J.: Relational dependency networks. Journal of Machine Learning Research 8, 653–692 (2007)
16. Neville, J., Simsek, O., Jensen, D.: Autocorrelation and relational learning: Challenges and opportunities. In: Proceedings of the Workshop on Statistical Relational Learning (2004)
17. Pace, P., Barry, R.: Quick computation of regression with a spatially autoregressive dependent variable. Geographical Analysis 29(3), 232–247 (1997)
18. Rey, S.J., Montouri, B.D.: U.s. regional income convergence: a spatial econometric perspective. Regional Studies (33), 145–156 (1999)
19. Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classification in network data. AI Magazine 29(3), 93–106 (2008)
20. Tobler, W.: Cellular geography. In: Gale, S., Olsson, G. (eds.) Philosophy in Geography (1979)
21. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
22. Zhou, Z.-H., Li, M.: Semisupervised regression with cotraining-style algorithms. IEEE Transaction in Knowledge Data Engineering 19(11), 1479–1493 (2007)
23. Zhu, X., Ghahramani, Z., Lafferty, J.D.: Semi-supervised learning using gaussian fields and harmonic functions. In: Fawcett, T., Mishra, N. (eds.) Proceedings of the 20th International Conference on Machine Learning, ICML 2003, pp. 912–919. AAAI Press, Menlo Park (2003)