Transductive Learning for Spatial Regression with Co-Training

Annalisa Appice, Michelangelo Ceci, Donato Malerba
Dipartimento di Informatica
Universita' degli Studi di Bari
via Orabona 4, Bari, Italy
{appice,ceci,malerba}@di.uniba.it

ABSTRACT

Many spatial phenomena are characterized by positive autocorrelation, i.e., variables take similar values at pairs of close locations. This property is strongly related to the smoothness assumption made in transductive learning, according to which if points in a high-density region are close, corresponding outputs should also be close. This observation, together with the prior availability of large sets of unlabelled data, which is typical in spatial applications, motivates the investigation of transductive learning for spatial data mining. The task considered in this work is spatial regression. We apply the co-training technique in order to iteratively learn two separate models, such that each model is used to make predictions on unlabeled data for the other. One model is built on the set of attribute-value observations measured at specific sites, while the other is built on the set of aggregated values measured for the same attributes in nearby sites. Experiments prove the effectiveness of the proposed approach on spatial domains.

Categories and Subject Descriptors

H.2 [**Data Management**]: Database applications—*Data mining, Spatial databases and GIS*

General Terms

Design, Algorithms, Experimentations

Keywords

Spatial Regression, Transductive Learning, Co-training

1. INTRODUCTION

Recent advances in positioning technology and location-based services have led to a rapid accumulation of spatial data for both research and commercial purposes. For the automatic analysis of such data, spatial data mining methods have received increased attention in statistics, database and data mining [18, 21].

The main issue faced by spatial data mining methods is *spatial autocorrelation*. Formally, spatial autocorrelation (or *spatial dependence*, as it is called in statistics) is the property of attributes taking values at pairs of locations a certain distance apart (neighborhood) to be more similar (positive autocorrelation) or less similar (negative autocorrelation) than expected for randomly associated pairs of observations [14]. Informally, spatial autocorrelation is observed by picturing the spatial variation of some observed attributes in a map: in the positive case, similar values tend to cluster, while in the negative case similar values scatter throughout the map. Positive spatial autocorrelation is common in ecological and environmental datasets, in many social phenomena as well as in geographical analysis, where it is known as Tobler's first law of geography, according to which "everything is related to everything else, but near things are more related than distant things" [22]. This justifies the focus of this work on such a widely found form of spatial autocorrelation.

Spatial autocorrelation violates a basic assumption made by traditional data mining methods, namely the independence of observations in the training sample, and it is responsible of their poor performance. As observed by LeSage and Pace [15], "anyone seriously interested in prediction when the sample data exhibit spatial dependence should consider a model which takes spatial autocorrelation into account".

Another issue, which is usually met in predictive spatial data mining tasks, is the *scarcity of labeled data*, since manual annotation of large data sets can be very costly. In this situation, it is important to exploit the large amount of information potentially conveyed by unlabeled data to better estimate the data distribution and to build more accurate predictors. Different learning settings have been proposed in the literature: the active setting [9] the semi-supervised setting [19], the transductive setting [13]. The active setting is an observation based learning paradigm where learner explicitly seeks for new training examples of high utility. The semi-supervised setting is a type of inductive learning, since the learned function is used to make predictions on any possible observation. The transductive setting asks for less, since it is only interested in making predictions for a set of unlabeled data known at the learning time. Actually, in many spatial domains observations to be predicted are already known in advance. This is the case of spatially referenced objects on maps already available in a Geographical Information System (GIS). This justifies the focus of this work on the transductive setting.

Transduction is based on the *smoothness assumption* according to which if two points in a high-density region are

SAC'10 March 22-26, 2010, Sierre, Switzerland.

close, then the corresponding outputs should also be close [6]. Interestingly, in spatial domains, where closeness of points corresponds to some spatial distance measure, this assumption is implied by positive spatial autocorrelation.

These considerations motivate the investigation of learning a spatial model in a transductive setting in order to deal with both spatial autocorrelation and scarcity of labeled data. The specific task of interest in this work is spatial regression. Although this task has been studied in spatial statistics and spatial data mining, at best of our knowledge, the present is the first work which explores the spatial regression in a transductive setting by proposing a method which is experimentally proved to be effective.

To formulate a solution for this spatial regression problem we face two main issues. First, we have to design a learner which accounts for positive spatial autocorrelation. More precisely, the learner should take into account the spatial continuity over attribute values when the response value is predicted and, at the same time, it should disaggregate the regression surface in different areas of homogeneous dependence. Second, we have to devise a suitable technique for the prediction of spatial unlabelled data.

The specific contribution of this work is to provide answers to both issues by generating models which both deal with spatial autocorrelation and reliably bootstrap from a small set of labeled training data via a large set of unlabeled data. We propose a novel algorithm, named SpReCo (Spatial Regression with Co-Training), which resorts to the co-training paradigm [3] in order to separately learn regressors from two different views of the same data. One view is defined only on original attributes (explanatory variables), whose values are measured at some specific spatial positions. The other view, which accounts for the possible spatial dependence, is based on aggregate attributes, whose values are derived by aggregating measurements of the explanatory attributes in the neighborhood of those spatial positions. According to the co-training paradigm, the regression model learned from a view is used to predict unlabeled data for the other during the learning process. However, only some unlabeled data are considered, namely the most reliable. The final prediction of unlabeled observations is the weighted average of the regression estimates generated by both learners. In this work, regressors considered at each iteration of SpReCo are model trees learned by the algorithm reported in [1].

This paper is organized as follows. The next section introduces a formalization of the transductive learning problem. Section 3 presents both the background of this work and the most relevant related literature. Section 4 describes the proposed solution. A theoretical analysis of the computational complexity of SpReCo is reported in Section 5. Experimental results are reported in Section 6. Finally, Section 7 concludes and presents ideas for further developments.

2. PROBLEM STATEMENT

In this work, spatial information is modeled according to the *field* model [20], i.e., the space is seen as a continuous surface over which features vary, and the spatial variation is defined by a number of functions $f : \mathbb{R}^2 \mapsto$ *Attribute domain*. The set D of observations is a set of tuples $(id, u, v, \mathbf{x}, y) \in (ID, U, V, \mathbf{X}, Y)$, where ID is a primary key, i.e., *id* identifies the position $\langle u, v\rangle \in U \times V = \mathbb{R}^2$, $\mathbf{X}$ is the vector of explanatory attributes ($\mathbf{X} = (\mathbf{X_1}, \ldots \mathbf{X_d})$) and $\mathbf{x}$ is the d-tuple of their values observed at the position $\langle u, v\rangle$, and $y \in Y = \mathbb{R}$ is the (possibly unknown) response value observed at $\langle u, v\rangle$ for $\mathbf{x}$.

In the transductive setting, the learner receives both full information (including responses) on the observations in the *training set* $T \subset D$ and partial information (without responses) on the observations in the *working set* $W = D - T$, and it is asked to predict the response values of the observations in W. Formally, *Given i*) the training set $T \subset D$; *ii*) the projection of the working set W on $ID \times U \times V \times \mathbf{X}$; *iii*) a *distance* function used to define the neighborhood of a position $\langle u, v\rangle$; *Find* predictions of the unknown response values of observations in W which are as accurate as possible.

The original formulation of the problem of function estimation in a transductive setting is *distribution-free* and requires that both T and W are sampled from D without replacement. This means that, unlike the standard inductive setting, the observations in the training (and working) set are supposed to be mutually dependent. Vapnik has also introduced a 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 [23](Theorem 8.1), error bounds for learning algorithms in the distribution-free setting also apply to the more popular distributional transductive setting. This justifies the focus of this work on the original distribution-free setting.

3. BACKGROUND AND RELATED WORKS

In recent years, several data mining and statistical methods have been investigated for the spatial regression task. A brief survey on regression techniques (e.g., k-NN, geographically weighted regression, kriging) which are appositely developed in order to take into account some form of spatial autocorrelation is reported in [18]. Anyway, they work with labeled data only, and, to the best of our knowledge, no method for spatial regression also consider unlabelled data.

Several algorithms have been devised for learning from labeled and unlabeled data simultaneously. Chapelle et al. [7] present an algorithm to estimate the values of a function at a set of unlabeled observations by minimizing the leave one-out error of Ridge Regression on the joint labeled and unlabeled set. Belkin et al. [2] propose semi-supervised learning algorithms which are based on a regularization that exploits geometry of marginal distribution. Cortes and Mohri [10] give explicit VC-dimension generalization bounds for transductive regression which are employed to design a regression algorithm that exploits local estimates of unlabeled observations and then performs a global optimization to ensure robust predictions. Wang et al. [24] extend a kernel regression approach which leverages labeled and unlabeled data. Unlabeled observations are predicted by assuming that their estimates are close to the actual values.

A prominent achievement in transductive and semi-supervised learning is the *co-training* paradigm. A recent survey of works related to co-training can be found in [26]. In co-training, independent views, e.g., distinct sets of attributes, of labeled and unlabeled data are available for deriving separate learners. Predictions of each learner on unlabeled observations are then used to augment the training set of the other in an iterative learning process. In regression, Brefeld et al. [5] resort to co-training in order to formulate a semi-supervised least square regression algorithm, where co-training is formulated as a regularized risk minimization problem in Hilbert spaces. The optimal solution in the

Hilbert space is described by a linear combination of kernel functions centered on the set of labeled and unlabeled observations. To achieve the multiple view learning, distinct functions are searched from different Hilbert spaces. Zhou and Li [25] apply co-training to k-NN regression. Instead of using two disjoint attribute sets they consider distinct distance measures for the two hypotheses.

To conclude, regression methods developed in transductive and semi-supervised learning, with or without co-training, do not deal, at least explicitly, with positive autocorrelation as a spatial regression method does. A recent study on transductive learning with applications in spatial domains [16] is focused on the classification task, while transduction in spatial regression remains almost untouch.

4. ALGORITHM

A top-level view of SpReCo is shown in Algorithm 1. Following the co-training paradigm, SpReCo operates with two views of the same data. The first data view is $D = T \cup W$ and includes the original values of explanatory attributes measured at some specific spatial positions. The second data view is $\overline{D} = \overline{T} \cup \overline{W}$ and includes the aggregate attribute values derived by aggregating measurements of the explanatory attributes in the neighborhood of those spatial positions.

The algorithm is iterative. SpReCo uses two sets (T_1 and T_2) of labeled observations, and two sets (W_1 and W_2) of unlabeled observations. Initially, $T_1 = T$, $T_2 = \overline{T}$, $W_1 = W$ and $W_2 = \overline{W}$. At each iteration, observations are moved from W_1 (W_2) to T_1 (T_2). This is done according to two regression models, denoted as t_1 and t_2, which are learned from T_1 and T_2 respectively (see the calls of the function *learn*). The learner is the same for both regression models. The model t_1 (t_2) is used to predict labels ($\widehat{y}$) for the observations falling in W_1 (W_2). Labels which are reliably predicted (see the calls of *predictReliableLabels*) are assigned to the corresponding observations in W_2 (W_1). These observations are then moved from W_2 to T_2 (W_1 to T_1). The function *instance* retrieves the description of an observation in the alternative view. The learning process stops either when the maximum number M of learning iterations is reached, or when no unlabeled observation has been moved from W_i to T_i, $i = 1, 2$. Finally, regression models learned in the last iteration of the learning process are used to definitely label observations in the working set (see the call of *label*).

Details on the aggregation of the explanatory variables in a neighborhood, the estimation of reliability of predicted labels, and the final labeling of observations in the working set are reported in the next Sub-sections.

4.1 Aggregation in a Neighborhood

The function *neighborhoodBasedDescription* returns, for each observation $e = (id, u, v, \mathbf{x}, y) \in D$, an alternative view $\overline{e} = (id, u, v, \overline{\mathbf{x}}, y) \in \overline{D}$. The computation of $\overline{x}$ is based on a neighborhood $N_h(e, D)$, i.e., the set of the h observations in D whose spatial positions are closest to $\langle u, v\rangle$. Closeness is computed by means of the function $distance(\cdot, \cdot)$, which corresponds to the Euclidean distance in this work.

The vector $\overline{\mathbf{x}} = (\overline{x_1}, \ldots, \overline{x_d})$ is computed as follows:

$$\overline{x_i} = \frac{\sum\limits_{e^j \in N_h(e,D)} (x_i^j \cdot w(e, e^j))}{\sum\limits_{e^j \in N_h(e,D)} w(e, e^j)} \qquad i = 1, \ldots, d, \qquad (1)$$

where x_i^j denotes the value taken by the explanatory variable X_i for e^j, while $w(e, e^j) \in \mathbb{R}^+$ is a weight proportional to the spatial proximity of e^j to e. We follow the weighting schema adopted in [8], according to which the weight is defined as a continuous function of the Euclidean distance, i.e.,:

$$w(e, g) = e^{-\frac{distance(e,g)^2}{b_e^2}} \qquad (2)$$

with $b_e = \max_{g \in N_h(e,D)} distance(e, g)$. If the UV coordinates of both e and g coincide, $w(e, g)$ is set to 1.

Algorithm 1 Main procedure of SpReCo.

```
SpReCo$(T, W)$
/* $T$: training set; $W$: working set; */
$\langle \overline{T}, \overline{W} \rangle \leftarrow$ neighborhoodBasedDescription$(T \cup W)$;
$T_1 \leftarrow T$; $W_1 \leftarrow W$; $T_2 \leftarrow \overline{T}$; $W_2 \leftarrow \overline{W}$;
$i \leftarrow 1$;
repeat
    change $\leftarrow$ false;
    $t_1 \leftarrow$learn$(T_1)$; $t_2 \leftarrow$learn$(T_2)$;
    $P_1 \leftarrow$predictReliableLabels$(t_1, T_1, W_1)$;
    $P_2 \leftarrow$predictReliableLabels$(t_2, T_2, W_2)$;
    if $P_1 \neq \emptyset$ or $P_2 \neq \emptyset$ then
        change $\leftarrow$ true;
        for $e \in P_1$ do
            $T_2 \leftarrow T_2 \cup \{\langle \text{instance}(e, W_2), \widehat{y_e} \rangle\}$;
            $W_2 \leftarrow W_2 - \{\text{instance}(e, W_2)\}$;
        end for
        for $e \in P_2$ do
            $T_1 \leftarrow T_1 \cup \{\langle \text{instance}(e, W_1), \widehat{y_e} \rangle\}$;
            $W_1 \leftarrow W_1 - \{\text{instance}(e, W_1)\}$;
        end for
    end if
    $i \leftarrow i + 1$;
until ($i \geq M$ OR NOT change); /* exit when true */
$W \leftarrow$label$(t_1, t_2, T, W, \overline{T}, \overline{W})$;
end
```

4.2 Prediction of Reliable Labels

Several approaches to estimate reliability of regression predictions are discussed in [4]. In SpReCo, reliability is estimated by extending to regression tasks the approach proposed in [25] for classification tasks. In particular, known labels of the nearest neighbors are considered in accordance with a k-NN based learning algorithm (see Algorithm 2). This choice is motivated by the fact that k-NN does not hold a separate training phase and naturally deals with spatial autocorrelation on the response attribute.

Intuitively, the most reliably labeled observation e should be the one which maximizes the consistency of a k-NN learner with the labeled set. More precisely, the k-NN is used to re-predict each observation p in the labelled set T_j, and SpReCo compares the squared errors made by k-NN when information on $\langle e, \widehat{y_e}\rangle$ is either considered ($knn(p, T_j \cup \{\langle e, \widehat{y_e}\rangle\})$) or not ($knn(p, T_j)$). The comparison is based on the difference:

$$\epsilon_p = (y_p - \text{knn}(p, T_j))^2 - (y_p - \text{knn}(p, T_j \cup \{\langle e, \widehat{y_e}\rangle\}))^2 \quad (3)$$

where y_p is the original label of p in T_j at the current iteration of SpReCo, while $\widehat{y_e}$ is the label predicted for e.

As k-NN learner, we adopt a version of weighted k-Nearest Neighbor algorithm, where responses in a k sized neighborhood of p are weighted according to same weighting function

Algorithm 2 Determine reliable labels.

```
1: predictReliableLabels($t_j, T_j, W_j$)
2: /* $t_j$: regression model; $T_j$: set of labeled observations;
   $W_j$: set of unlabeled observations; */
3: $P_j \leftarrow \emptyset$;
4: for $e \in W_j$ do
5:     $\widehat{y_e} \leftarrow$ response($t_j, e$);
6:     $\epsilon_+ \leftarrow 0$; $\epsilon_- \leftarrow 0$;
7:     for $p \in N_k(e, T)$ do
8:         $\epsilon_p \leftarrow (y_p - knn(p, T_j))^2 - (y_p - knn(p, T_j \cup \{\langle e, \widehat{y_e}\rangle\}))^2$;
9:         if $\epsilon_p \geq 0$ then $\epsilon_+ \leftarrow \epsilon_+ + 1$;
10:                      else $\epsilon_- \leftarrow \epsilon_- + 1$;
11:     end for
12:     if $\epsilon_+ \geq \epsilon_-$ then $P_j \leftarrow P_j \cup \{\langle e, \widehat{y_e}\rangle\}$;
13: end for
14: return $P_j$
```

defined in Equation 2. Then:

$$knn(p, T_j) = \frac{\sum\limits_{q \in N_k(p,T_j)} y_q \cdot w(p, q)}{\sum\limits_{q \in N_k(p,T_j)} w(p, q)}. \tag{4}$$

Repeatedly measuring the squared errors of the k-NN predictions for each observation in T_j is time-consuming. However, we can compute a computationally efficient approximation, by considering only the labeled observations in $N_k(e, T_j)$. This choice is coherent with the fact that a k-NN learner uses only local information. Therefore, we consider:

$$\epsilon_+ = |\{p \in N_k(e, T_j) | \epsilon_p \geq 0\}| \quad \epsilon_- = |\{p \in N_k(e, T_j) | \epsilon_p < 0\}|, \tag{5}$$

and we consider the estimated label $\widehat{y_e}$ to be reliable if $\epsilon_+ \geq \epsilon_-$, un-reliable otherwise.

4.3 Final Labeling of Working Observations

The regression models t_1 and t_2, which are constructed in the last iteration of the main procedure, are finally used to predict the final labels of observations in the working set W (see Algorithm 3). More precisely, for each $e \in W$, the corresponding observation $\overline{e} \in \overline{W}$ with the same identifier of e is retrieved. Then, t_1 is used to predict the label of e while t_2 is used to predict the label of $\overline{e}$ (see the calls of the function *response*). The weighted average of these two predicted labels is finally assigned to e. The weights are computed on the basis of the mean square errors (*mse*) of both t_1 and t_2 on the original training set, T, and its alternative view $\overline{T}$, respectively.

5. LEARNING TIME COMPLEXITY

The time complexity of SpReCo depends on the complexity of the basic regression learner that is used to induce both t_1 and t_2. We denote this time-complexity as α.

The construction of the second data view (Alg. 1;ln.3) is preliminary to the execution of main cycle (repeat ... until) and, in the worst case, its time complexity is $O(d(n+m)^2)$, where n is the size of the training set T, m the size of the working set W, d is the number of explanatory attributes in X ($\overline{X}$) and $(n+m)^2$ is due to the computation of distances.

Algorithm 3 Final prediction of working labels.

```
1: label($t_1, t_2, T, W, \overline{T}, \overline{W}$)
2: /* $t_1$ ($t_2$): regression model over $X$ ($\overline{X}$); $T$ ($\overline{T}$): training
   set over $X$ ($\overline{X}$); $W$ ($\overline{W}$): working set over $X$ ($\overline{X}$); */
3: $m_1 \leftarrow mse(t_1, T)$; $m_2 \leftarrow mse(t_2, \overline{T})$; $Y_W \leftarrow \emptyset$;
4: if $m_1 > m_2$ then
5:     $\omega_1 \leftarrow 1$; $\omega_2 \leftarrow m_1/m_2$;
6: else
7:     $\omega_1 \leftarrow m_2/m_1$; $\omega_2 \leftarrow 1$;
8: end if
9: for $e \in W$ do
10:     $\overline{e} \leftarrow$ instance($e, \overline{W}$);
11:     $Y_W \leftarrow Y_W \cup \{\langle e, \frac{\text{response}(t_1,e)\omega_1 + \text{response}(t_2,\overline{e})\omega_2}{\omega_1+\omega_2}\rangle\}$;
12: end for
13: return $Y_W$
```

The time complexity of the main cycle of Algorithm 1 is:

$$O\left(M(\underbrace{2\alpha}_{(Alg.1;ln.8)} + 2m \overbrace{(\underbrace{c}_{(Alg.2;ln.5)} + \underbrace{2k(n+m)}_{(Alg.2;ln.7-14)})}^{(Alg.1;ln.9-10)}) \right), \tag{6}$$

where c denotes the cost of predicting $\hat{y_e}$ given the regression model. The time complexity of main cycle of Algorithm 1 is then $O(M\alpha + Mmc + Mkm(n+m))$ in the worst case, since lines 11-22 do not affect complexity.

Finally, the complexity of labeling examples in the working set (Alg. 1; ln.24) is:

$$O(\underbrace{2nc}_{(Alg.3;ln.4-8)} + \underbrace{2mc}_{(Alg.3;ln.9-12)}) = O(c(n+m)). \tag{7}$$

Therefore, in the worst case, the time complexity of SpReCo is $O(d(n+m)^2 + M\alpha + Mmc + Mkm(n+m) + c(n+m))$. This analysis is simplified by observing that both M and k are (small) user-defined constants, and by supposing that the time required by a model t to classify one example can be neglected. Hence, the time complexity of SpReCo is $O(d(n+m)^2 + \alpha)$, i.e., it is quadratic in the size of D and linear both in the number of explanatory variables and in the complexity of the model learner.

6. EXPERIMENTS

The proposed algorithm has been evaluated on four spatial databases whose description is reported below.

USA Geographical Analysis Spatial Data (GASD). This georeferenced data set [17] contains 3,107 observations on USA county votes cast in 1980 presidential election. for each county explanatory attributes are: the population of 18 years of age or older, the population with a 12th grade or higher education, the number of owner-occupied housing units, and the aggregate income. The response attribute is the total number of votes cast. For each county, the spatial coordinates (U,V) of its centroid are available.

Forest Fires. This dataset is publicly available in the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/). Details are reported in [11]. It collects 512 observations of forest fires in the period January 2000 to December 2003 in the Montesinho natural park, Portugal. The explanatory attributes are: the Fine Fuel Moisture Code, the Duff Moisture Code, the Drought Code, the Initial Spread Index, the temperature in Celsius degrees, the relative humidity, the wind speed in km/h, and the outside rain in mm/m^2. The response attribute is the burned area of the forest in ha (with 1ha/100 = 100 m^2). The spatial coordinates (U,V) refer to the centroid of the interested area in a park map.

North-West England (NWE). Data concerns the region of North West England, which is decomposed into 1011 censual wards. Both explanatory and response variables available at ward level are taken from 1998 Census. They are: percentage of mortality (response attribute) 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. Spatial coordinates (U,V) refer to the ward centroid. By removing observations including null values, only 979 observations are used in this experiments.

Sigmea-Real. This dataset [12] collects 817 measurements of the rate of herbicide resistance of two lines of plants (response attributes), that is, the transgenic male-fertile (MF) and the non-transgenic male-sterile (MS) line of oilseed rape. Explanatory attributes 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. Spatial coordinates (U,V) of the plant are available.

The experiments aim at validating the actual advantage in accuracy of the transductive algorithm over its inductive counterpart when few labeled observations are available. The baseline regression models are the model trees t_1 and t_2 which are induced from the training sets T and $\overline{T}$ respectively. Details on the model tree learner are reported in [1]. We also evaluate the advantages of a co-training implementation in transductive learning by comparing SpReCo with a simple ensemble of model trees induced from the different views of training data ($M = 5$ vs $M = 1$).

The empirical comparison is based on the mean square error (MSE). To estimate the MSE, a K-fold cross validation is performed and the average MSE (Avg.MSE) over the K-folds is computed. Due to the different size of the datasets, we set $K = 10$ in experiments on the GASD dataset, and $K = 5$ in all other cases. Differently from standard cross-validation, for each trial we use a single fold as training set, and the remaining $K - 1$ folds for the working set. This way, we simulate the scarcity of labeled data. Experiments are repeated with different values of h and k for SpReCo. h ranges from 1 (no co-training), to 5, 10, 15 and 20, while k ranges from 5, to 10, 15 and 20.

The percentage of Avg.MSE reduction ($\delta_{SpReCo\leftarrow Baseline}$) of the transductive learner with respect to the (ensemble of) baseline inductive algorithm is reported in Table 1.

$$\delta_{SpReCo\leftarrow Baseline} = (1 - \frac{Avg.MSE_{SpReCo}}{Avg.MSE_{Baseline)}}). \tag{8}$$

A positive (negative) value of $\delta_{SpReCo\leftarrow Baseline}$ is in favor of the transductive (inductive) algorithm. Moreover, the greater in absolute value, the more accurate the algorithm.

Results suggest several conclusions. First, they confirm that SpReCo performs generally better than at least one of the basic model tree learners (several times both of them) 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 the Sigmea Real dataset (MS-MF), in which case the baseline inductive learner t_1 often outperforms SpReCo. Our justification is that the worse performance of SpReCo may depend on the fact that MS (MF) dataset exhibits about 59% (65%) of observations which are labeled as zero. High percentage of zero valued labels leads to a degradation of predictive capability of the learner which operates with the aggregate data view. In fact, the accuracy of model tree t_2 is significantly worse than the accuracy of the classical model tree t_1. In this case, co-training is beneficial since it reduces the error due to t_2. Second, we observe that the co-training improves the accuracy of the transductive learner more than a simple ensemble of model trees generated from different views of data ($M = 5$ vs $M = 1$). Finally, we observe the dependence of the performance of SpReCo on both h and k. Although experiments show that best accuracy is obtained with $h > 1$, that is, when co-training does not degenerate in self training, best performance is generally obtained by considering relatively low sized neighborhood ($h = 5$ and $k = 5$). This result provides a guideline to choose h and k.

7. CONCLUSIONS

In this paper we have motivated, described and empirically evaluated SpReCo, a transductive learning algorithm for regression tasks whose design is based on the co-training paradigm. Two regression models are induced from two views of the same set of labeled data. One model is built on the set of attribute-value observations measured at specific sites, while the other is built on the set of aggregated values measured for the same attributes in nearby sites. Each regression model is used to predict the response of the unlabeled data for the other model during the learning process. The type of regression models considered in this work are model trees, which combine local models computed on portions of the feature space. Experimental results generally confirm the effectiveness of our proposal and lead to the conclusion that transduction is an important direction for further research in Spatial Data Mining, since there is a clear contiguity of the concept of positive autocorrelation with the smoothness assumption in transduction. In principle, we expect that a strong spatial autocorrelation should counterbalance the lack of labeled data, if spatial autocorrelation is taken into account in the transductive setting. As future work, we plan to investigate how automatically tune several parameters of SpReCo and pick the best one.

8. ACKNOWLEDGMENT

This work is supported by the PRIN/COFIN 2007 “Multi-relational approach to spatial data analysis”. The authors thank Saso Dzeroski for kindly providing Sigmea Real dataset.

9. REFERENCES

[1] A. Appice and S. Dzeroski. Stepwise induction of multi-target model trees. In *ECML 2007*, volume 4701 of *LNCS*, pages 502–509. Springer-Verlag, 2007.

[2] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[3] A. Blum and T. M. Mitchell. Combining labeled and unlabeled data with cotraining. In *COLT 1998*, pages 92–100, 1998.

[4] Z. Bosnić and I. Kononenko. Comparison of approaches for estimating reliability of individual

Table 1: Transductive learning vs inductive learning: percentage of error reduction ($\delta_{SpReCo \leftarrow Baseline}$%).

h	k	GASD			Forest Fires			NWE			Sigmea Real-MS			Sigmea Real-MF		
M=5		t1	t2	M=1	t1	t2	M=1	t1	t2	M=1	t1	t2	M=1	t1	t2	M=1
1	5	8.9	-	8.92	-1.3	-	-1.32	1.1	-	1.14	0.1	-	0.09	0.0	-	0.01
1	10	10.9	-	10.91	-2.4	-	-2.43	2.3	-	2.27	0.1	-	0.09	-0.6	-	-0.57
1	15	11.0	-	10.97	-3.1	-	-3.06	-0.2	-	-0.23	0.0	-	0.05	-0.6	-	-0.56
1	20	10.8	-	10.82	-1.9	-	-1.89	2.0	-	1.99	0.1	-	0.09	-0.4	-	-0.39
5	5	14.7	18.5	10.13	19.8	-0.6	7.93	3.9	1.9	1.33	4.8	3.5	1.89	-1.0	7.5	2.11
5	10	14.7	18.5	10.16	20.8	0.7	9.13	3.5	1.4	0.89	4.7	3.5	1.87	-4.0	4.8	-0.74
5	15	14.5	18.3	9.99	20.1	-0.2	8.31	3.7	1.7	1.17	3.6	2.3	0.67	-2.4	6.2	0.77
5	20	14.4	18.2	9.83	20.1	-0.3	8.22	3.9	1.9	1.33	-1.4	-2.8	-4.49	-1.9	6.7	1.22
10	5	13.7	24.9	11.93	21.5	1.6	9.93	2.6	1.1	0.10	-0.3	3.7	-0.04	-3.0	3.5	-1.48
10	10	13.9	25.2	12.22	20.9	0.8	9.25	2.6	1.1	0.13	0.2	4.3	0.52	-4.1	2.4	-2.62
10	15	13.9	25.2	12.18	20.3	0.0	8.51	2.6	1.1	0.10	-0.9	3.2	-0.60	-5.6	1.0	-4.09
10	20	12.9	24.3	11.15	20.8	0.7	9.15	2.8	1.3	0.32	1.3	5.3	1.55	-3.9	2.6	-2.38
15	5	13.5	24.7	12.63	20.6	0.6	8.91	2.8	2.9	0.82	0.4	3.8	-0.08	-2.4	4.3	-1.09
15	10	11.6	23.0	10.68	20.6	0.5	8.84	3.0	3.1	0.99	-0.8	2.6	-1.28	-3.9	2.9	-2.54
15	15	13.5	24.7	12.62	20.8	0.8	9.07	2.8	2.9	0.80	2.0	5.4	1.57	-1.1	5.6	0.26
15	20	12.0	23.3	11.09	21.0	1.1	9.35	2.9	3.0	0.88	1.2	4.6	0.78	-2.7	4.0	-1.38
20	5	10.8	22.3	9.87	21.3	2.0	9.82	2.6	1.5	0.23	3.7	3.8	1.31	-3.7	5.5	-0.89
20	10	11.6	23.0	10.67	20.5	1.1	8.95	2.8	1.7	0.46	1.7	1.8	-0.75	-5.0	4.3	-2.13
20	15	12.5	23.8	11.50	20.3	0.7	8.65	2.6	1.5	0.27	1.7	1.8	-0.75	-4.8	4.5	-1.94
20	20	11.4	22.8	10.41	20.9	1.5	9.37	2.5	1.4	0.17	5.6	5.7	3.28	-3.4	5.8	-0.58

regression predictions. *Data Knowledge Engineering*, 67(3):504–516, 2008.

[5] U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel. Efficient co-regularised least squares regression. In *ICML 2006*, volume 148, pages 137–144. ACM, 2006.

[6] O. Chapelle, B. Scholkopf, and A. Zien. *Semi-supervised learning.* MIT Press, 2006.

[7] O. Chapelle, V. Vapnik, and J. Weston. Transductive inference for estimating values of functions. In *NIPS 1999*, pages 421–427. The MIT Press, 1999.

[8] M. Charlton, S. Fotheringham, and C. Brunsdon. Geographically weighted regression. In *NCRM Methods Review Papers*, 2005.

[9] D. A. Cohn, L. E. Atlas, and R. E. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[10] C. Cortes and M. Mohri. On transductive regression. In *NIPS 2006*, pages 305–312. MIT Press, 2006.

[11] P. Cortez and A. Morais. A data mining approach to predict forest fires using meteorological data. In *EPIA 2007*, pages 512–523. APPIA, 2007.

[12] D. Demšar, M. Debeljak, C. Lavigne, and S. Džeroski. Modelling pollen dispersal of genetically modified oilseed rape within the field. In *Annual Meeting of the Ecological Society of America*, page 152, 2005.

[13] A. Gammerman, K. S. Azoury, and V. Vapnik. Learning by transduction. In *UAI 1998*, pages 148–155. Morgan Kaufmann, 1998.

[14] P. Legendre. Spatial autocorrelation: Trouble or new paradigm? *Ecology*, 74:1659–1673, 1993.

[15] J. P. LeSage and K. Pace. Spatial dependence in data mining. In R. Grossman, C. Kamath, P. Kegelmeyer, V. Kumar, and R. Namburu, editors, *Data Mining for Scientific and Engineering Applications*, pages 439–460. Kluwer Academic Publishing, 2001.

[16] D. Malerba, M. Ceci, and A. Appice. A relational approach to probabilistic classification in a transductive setting. *Eng. Appl. Artif. Intell.*, 22(1):109–116, 2009.

[17] P. Pace and R. Barry. Quick computation of regression with a spatially autoregressive dependent variable. *Geographical Analysis*, 29(3):232–247, 1997.

[18] S. Rinzivillo, F. Turini, V. Bogorny, C. Körner, B. Kuijpers, and M. May. Knowledge discovery from geographical data. In *Mobility, Data Mining and Privacy*, pages 243–265. Springer-Verlag, 2008.

[19] B. Shahshahani and D. Landgrebe. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Transactions on Geoscience and Remote Sensing*, 32:1087–1095, 1994.

[20] S. Shekhar and S. Chawla. *Spatial databases: A tour.* Prentice Hall, 2003.

[21] S. Shekhar, R. Vatsavai, and S. Chawla. Spatial classification and prediction models for geospatial data mining. In H. Miller and J. Han, editors, *Geographic Data Mining and Knowledge Discovery (second edition)*, pages 117–147. Taylor & Francis, 2009.

[22] W. Tobler. Cellular geography. In *Philosophy in Geography*, 1979.

[23] V. Vapnik. *Statistical Learning Theory.* Wiley, 1998.

[24] M. Wang, X.-S. Hua, Y. Song, L.-R. Dai, and H. Zhang. Semi-supervised kernel regression. In *ICDM 2006*, pages 1130–1135. IEEE, 2006.

[25] Z.-H. Zhou and M. Li. Semisupervised regression with cotraining-style algorithms. *IEEE Transaction in Knowledge Data Engineering*, 19(11):1479–1493, 2007.

[26] Z.-H. Zhou and M. Li. Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 2009.