Annalisa Appice, Corrado Loglisci, Donato Malerba, Active learning via collective inference in network regression problems, Information Sciences, Volumes 460–461, 2018, Pages 293-317, ISSN 0020-0255,

The published version is available on

https://doi.org/10.1016/j.ins.2018.05.028

Active Learning via Collective Inference in Network Regression Problems

Annalisa Appice[a,b,c,*], Corrado Loglisci[a,b], Donato Malerba[a,b,c]

[a] *Department of Informatics, Università degli Studi di Bari Aldo Moro, via Orabona, 4 - 70125 Bari - Italy*
[b] *Consorzio Interuniversitario Nazionale per l'Informatica - CINI, Italy*
[c] *Centro Interdipartimentale di Logica e Applicazioni - CILA, Italy*

Abstract

Active learning is a promising machine learning paradigm for querying oracles and obtaining actual labels for particular examples. Its goal is to decrease the number of labels needed, in order to learn a predictive model able to achieve a high level of accuracy. It may turn out to be advantageous in several regression problems where scarce labels can be acquired. A novel active learning algorithm for regression problems in network data is defined. This algorithm performs active learning by taking into account explicitly the correlation property of network data, which makes the labels of linked nodes related to each other. Specifically it resorts to collective inference, in order to accommodate the data correlation in the active selection of the network nodes labeled by oracles. The empirical study proves that the proposed combination of active learning and collective inference can actually boost regression performances in various network domains.

Keywords: Network Regression, Active Learning, Collective Inference, Correlation Analysis

1. Introduction

Artificial intelligence (AI) has been considerably developed during the last thirty years. Thus far, a wide number of artificial intelligence systems has provided various machine learning algorithms for accurate predictive modeling (e.g. [2, 3, 22, 33]). They commonly find a model in data and predict unseen values using that model. As computer systems continue to become more and more powerful and complex by collecting huge volumes of data - often with a complex structure - the capabilities of artificial intelligences will also increase. Network data (e.g. sensor networks, communication and financial networks, web and social networks) are becoming an increasingly important challenge in AI [37]. Regardless of where we encounter them in day-to-day life, network data consist of nodes, which may be connected to each other by edges. The nodes of a network arising from a peculiar domain are, generally, of the same type, i.e. they are described by a vector of fixed properties. The nodes of networks arising from different domains, however, may be associated with different properties. For example, the nodes of a social network (e.g.

*Corresponding author (Tel: +39 (0)805443262 Fax: +39(0)805443269)
Email addresses: annalisa.appice@uniba.it (Annalisa Appice), corrado.loglisci@uniba.it (Corrado Loglisci), donato.malerba@uniba.it (Donato Malerba)

Twitter) are associated with social properties (e.g. number of tweets, number of followers), while the nodes of a spatial network (e.g. a solar photovoltaic grid) are associated with geospatial properties (e.g. solar radiation, temperature). The edges between the nodes may express an explicit relation, which reflects the dependence between the properties of the nodes. Friendship is an example of relation in social networks, while geographical closeness is an example of relation in spatial networks. This study proposes a machine learning algorithm for predictive modeling in a data network. The model found in a training data network can be used to predict unseen data of a new testing network that arises from the same training domain (i.e. both training and testing nodes are described by a vector with the same properties; they are linked according to the same type of relation).

Predictive modeling of network data is made complex due to the presence of *correlation*. This is a deterministic or probabilistic dependence between the values of a property on linked nodes [18]. It is apparent in the positive form in a wide variety of network domains like social and spatial domains [29, 49]. In social data analysis, correlation can be recognized in the homophily principle, that is, the tendency of nodes with similar values to be linked with each other [32]. In spatial data analysis, correlation can be recognized in Tobler's first law of geography, that is, the tendency of a geophysical attribute to take a value at a given location that may be similar to the values of that attribute in nearby locations [27]. Recent studies [6, 12, 30, 36, 45, 49] have shown that taking label correlations into account may contribute to improving the accuracy of predictive inferences in network data domains [36, 49]. In this context, *collective inference* algorithms, that reason collectively by predicting labels of linked examples simultaneously, offer a unique opportunity to accommodate label correlations in the learned models [36]. Although most work on collective inference is defined for classification problems, a few collective inference algorithms have recently been proposed for regression problems [6, 29]. In any case, these studies do not pay any attention to the procedure to acquire a representative labeled set from the data network, while this is often the first step towards performing accurate predictive inference, even when few labels are acquired [20].

The widely-used paradigm for label collection is called passive learning, where training samples are randomly selected from the underlying distribution and manually annotated by an oracle (e.g. human experts). However, due to the high cost associated with the above label collection process, it often happens that there are not enough labeled samples to train a high quality predictive model. An important research question is to develop algorithms that learn an accurate model with minimal labeling effort required in such tasks. One promising learning strategy is to use *active learning*. In this strategy, rather than being presented with a labeled training set from the start, the learner is allowed to request labels for particular examples, with the goal of decreasing the number of labels needed to achieve the desired level of accuracy. At present, various active learning algorithms have been investigated for network classification [7, 26, 31, 53]. Few active learning algorithms have already been developed for regression [9–11, 16, 39]. A seminal study [25] combines active learning and correlation analysis of univariate linked numeric data. However, to the best of our knowledge, active learners that take direct advantage of correlation in linked multivariate data have still not been considered for network regression problems.

The main contribution of this study is the description of a novel holistic strategy, where collective inference is used to drive the active learning for network regression. A novel algorithm, called CoNeRa (Collective Network Regression via Active learning) is described. It performs predictive modeling for a numeric target (also called label) in an initially unlabeled training data network. Specifically it iteratively selects a budget of training nodes to be labeled by the oracle, so that a final accurate regression model can be learned from this partially oracle-labeled network. This model can be used to predict unseen data in a testing data network that is acquired in the same domain condition of training.[1] This algorithm uses both the descriptive information (node properties) and the network structure (correlation property) during the training procedure. According to the collective inference theory, the algorithm learns a collective regression model by accounting for descriptive data associated with nodes, as well as collective data yielded by handling the property of label correlation throughout the network. According to the active learning theory, the algorithm requests labels for particular examples, which are selected based upon a disagreement measure. Specifically, the disagreement quantifies the correlation of the (potential) labels surrounding a (potential) target value. It is noteworthy that this algorithm is based on regression, as the final goal is performing predictive modeling of a numeric target. However, it implements an active learning procedure that also uses clustering, that is a form of unsupervised classification. Specifically, the active learning component integrates a new constraint-based clustering solution. This component is able to discover a cluster structure that depicts the correlation observed throughout the network in the descriptive data (i.e. each cluster represents a region of connected nodes associated with similar descriptive data). Assuming a dependence between the descriptive variables and the target variable, this cluster knowledge, discovered as a model of the descriptive data correlation, should implicitly provide some information on the correlation property of the target. Under this assumption, the active learning procedure considers this cluster knowledge, in order to guarantee diversity in the label acquisition and avoid over-investing in descriptive areas of the training data, which have already been explored. Finally, the collective regression model, induced from the final set of examples labeled by the active learning, can be used in a testing procedure involving unseen nodes.

The paper is organized as follows. Section 2 summarizes the main research contribution of this study, while Section 3 reports relevant related work. Section 4 illustrates the proposed collective active learning algorithm and its time complexity. Section 5 describes the datasets, the experimental methodology and reports the results. Finally, in Section 6 some conclusions are drawn and future work is outlined.

2. Research Contribution

The idea of combining collective inference with active learning is not new in machine learning research; indeed, it has already been used for the classification of network data [7]. Theoretically, the collective active

[1] The presented algorithm is formulated, in order to process data acquired at a specific time without accounting for mechanisms of incremental modeling of historical data. This is now out of the scope of this paper, although it may represent an interesting future direction of investigations.

learning strategy defined for the classification can also be adapted to the regression of network data by learning a regression model instead of a classification model, synthesizing collective information on numeric labels instead of collective information on categorical labels, defining disagreement in the numeric domain instead of disagreement in the categorical domain. In any case, apart from the innate difference between learning a regression model (with a numeric target) and learning a classification model (with a categorical variable), this study extends this state-of-the-art in machine learning in several directions. Specifically, it gives the following research contributions.

First, a collective active learning strategy is fully defined for network regression problems. This formalization describes mechanisms which synthesize collective information on numeric labels. It also defines a disagreement measure working in the regression scenario. Second, a contiguity constraint clustering algorithm is illustrated, in order to learn a model of the correlation property throughout the network. Clustering is simultaneously performed both in the (descriptive) data space and in the (edge) network structure. It partitions network nodes in clusters, where each cluster depicts a connected region of correlated data. The active learning mechanism accounts for this cluster knowledge to guarantee the diversity of the active sample. In particular, the active examples for the oracle labeling are selected cluster per cluster. Under the assumption that similar nodes are grouped in the same cluster, while different nodes are grouped in separate clusters, this cluster-based selection of active examples will avoid the problem of selecting "similar" examples for labeling and contribute to diminishing the loss in the generalization of the learned model. Third, a new disagreement measure is formulated, in order to select optimal unlabeled nodes whose labels will be acquired by the oracle. This measure quantifies the scarcity of correlation among the target values over linked nodes. For every unlabeled node, it measures the average deviation between labels associated with neighbor (linked) nodes and the label associated with the node under consideration. To perform this analysis, the target values, which are currently unknown in the training data network, are predicted by the currently learned collective regression model. The scarcity of the target correlation suggests the existence of a high number of errors in the predicted labels and the opportunity of acquiring ground truth of these labels, in order to appropriately correct the induced regression model. This approach using correlation of numeric labels in the active selection phase is novel for regression. As an additional contribution, the formulation of the presented correlation-aware disagreement measure is appropriately strengthened by the consideration of the knowledge enclosed in the cluster structure of the network. Fourth, an extensive evaluation of the effectiveness of the proposed algorithm is performed; indeed, regression problems in real network data from various social and spatial domains are being considered. The evaluation also includes the investigation of the correlation property of the target data, in order to support the decision of performing collective inference. The experimental results show that the presented algorithm generally outperforms several regression baselines defined in active learning that disregard the network structure of data, as well as collective active state-of-the-art learning algorithms, that account for the label correlation through the induction of a collective regression model (Section 5.4.1). This algorithm also proves that all the components of our proposal contribute to its efficacy (Section 5.4.2). It analyzes the sensitivity of the performance to the parameter

set-up (Section 5.4.3) and confirms the effectiveness of the suggested clustering solution along its peculiar algorithmic aspects (Section 5.4.4). Fifth, a preliminary investigation of the impact of the temporal dimension on the viability of the regression model learned by the proposed algorithm is illustrated (Section 5.5). This analysis highlights that models learned from historical data can sometimes be performed appropriately in the future when training and testing data behave similarly. Finally, results recently reported in the active learning literature and achieved by addressing a few network regression problems are discussed (Section 5.6).

Before detailing these contributions through the illustration of both the proposed strategy (Section 4) and its empirical evaluation (Section 5), the state-of-the-art will be described (Section 3), in order to clarify the research gap identified by this study with respect to the current related literature.

3. Related work

This Section presents a state-of-the-art review in active learning for regression (see Section 3.1) and network data (see Section 3.2): they are the key elements of the algorithm presented in this study.

3.1. Active learning for regression

The active learning strategy is mainly investigated for classification problems (see [20] for a survey). However, few active learning algorithms are defined for regression problems.

Cai et al. [10] propose a regression algorithm that bases active learning on Stochastic Gradient Descent. It estimates the expected regression model change, in order to select the active examples that lead to the largest changes of the parameters of the learned model. Ceperic et al. [11] define a similar approach for a multi-kernel support vector regression (SVR) algorithm. The labeled training data set is incrementally built, based on the influence of each new labeled example on the accuracy of the learned regression model.

Burbidge et al. [9], Pasolli et al. [39] and Douaka et al. [16] define regression algorithms that apply a query-by-committee mechanism [19], in order to select examples for labeling. This mechanism uses a committee of regression models and selects the active examples that achieve the maximal disagreement among labels predicted by the committee. The regression committee includes several regression models computed from subsets of the training data. Pasolli et al. [39] and Douaka et al. [16] also describe algorithms that account for the hidden resemblance between labeled and unlabeled examples. They use the covariance measure to select the active examples that are the most dissimilar from the current labeled training data.

Demir and Bruzzone [15] illustrate a two-stepped cluster-based algorithm that selects active examples for a Support Vector regression learner. In the first step, a clustering algorithm is applied to the training set composed of both unlabeled and labeled examples, which are not support vectors (SVs). Clusters that have non-SVs inside are filtered out, whereas all the unlabeled examples of the remaining clusters are taken as the most relevant examples. In the second step, the clusters with the highest example density are considered and one example is selected per cluster for the label request. This idea of a cluster-based sample selection

criterion is oriented to avoid the problem of selecting "similar" examples for the labeling. In fact, over-investing on similar examples would cause a loss in the generalization of the learned models. This problem may occur especially when the "best" examples are naively selected. The sample *diversity* is a crucial aspect in active learning. Reitmaier and Sick [44] describe a sophisticated sampling selection criterion to guarantee the sample diversity in a generative active learner. This criterion is based on the data properties and the distance of the examples from the decision boundary. It also depends on the density throughout the regions, where the examples are selected, as well as the diversity of the examples in the query set chosen for labeling the unknown label distribution of the examples. Reitmaier et al. [43] extend this study by exploring the use of a probabilistic generative model. Therefore, both these two studies formulate solutions dealing with sample diversity in classification problems, without exploring the same problem in the regression scenario.

Son and Lee [47] focus on the formulation of a sample selection criterion that is based on *uncertainty sampling*. In particular, they integrate the uncertainty sampling mechanism into a Bayesian-inspired regression approach, by accounting for the relevance vector machine regression algorithm. Their proposal is based on the rationale that the relevance vectors are located at the local maximal points of the predictive variance. These maximal points would correspond to the examples with maximum uncertainty.

The afore-mentioned studies describe various regression algorithms for performing active learning in independent and identically distributed data. They all apply active learning mechanisms without explicitly accounting for the correlation property of data connected throughout the network.

3.2. Active learning for network data

Regardless of the peculiar learning paradigm, networks are inherently complex data; indeed, they are characterized by a structure that makes the examples (nodes) dependent on each other. This phenomenon is a violation of the assumption that data are independently and identically distributed and makes the traditional learning algorithms poorly accurate when applied to network data [35]. The ubiquity of network data in day-to-day life is the new emergent challenge for developing novel machine learning approaches capable of exploiting the network information and improving accuracy of inferences about linked examples. Concentrating on active learning paradigm, the challenge of dealing with network information has received little attention for classification problems. To the best of our knowledge, the state-of-the-art of current research in machine learning has not investigated active learning for network regression problems in depth.

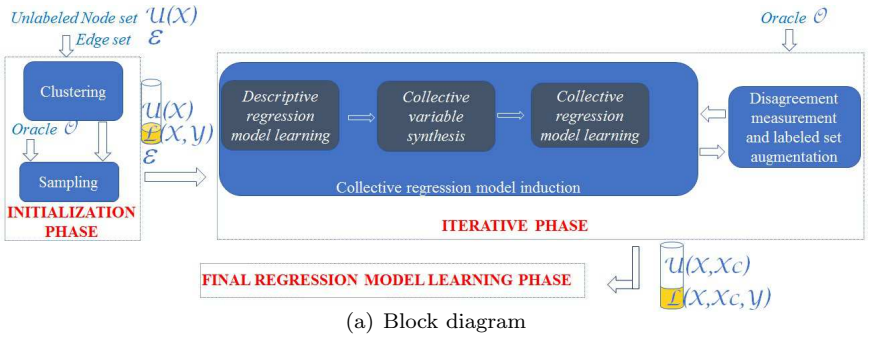
Macskassy [31] authors a seminal study that embeds the network structure in the sample selection criterion of an active classification learner. He uses graph-based metrics (e.g., clustering coefficient and betweenness centrality), in order to identify a pool of informative examples and applies an empirical risk minimization technique to select the most informative example from the pool. However, in this study, the predictive model is still learned neglecting the correlation property of data.

Bilgic et al. [7] propose a classification algorithm that accounts for the network structure through node communities, built as graph-based clusters. They rely on collective inference and perform active learning via query-by-committee. Collective inference [36, 45] accounts for the presence of label correlations in

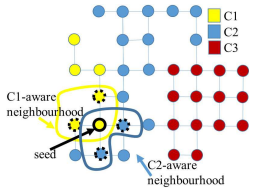
network data. In particular, descriptive properties associated with network nodes are augmented with new collective properties, which are constructed by summarizing label values in a neighborhood. The committee is composed of three classifiers induced from the labeled nodes: the classifier spanned on the descriptive properties, the classifier spanned on the collective and descriptive properties and the majority label associated with a cluster. Procedurally, the algorithm initially derives a network structure by performing graph-based clustering, in order to partition the network into communities. The nodes for the initial labeled set are picked from these communities. Then, the algorithm iteratively computes the entropy of the committee of three classifiers for each unlabeled node and selects examples with the highest entropy from the clusters with the highest average entropy. Labels are acquired for these selected examples. The classifiers are, therefore, trained on the augmented labeled set, in order to perform new inferences on unlabeled nodes linked to labeled nodes. It is noteworthy that the kind of strategy presented in [7] takes inspiration from the same elements (collective inference, cluster knowledge in the active learning mechanism) as the ones used in the strategy presented in this study. However, there are crucial differences to be highlighted: first, strategy in [7] is defined for classification and some effort is requested to extend it, in order to address regression;[2] second, the graph-based clustering step performed in [7] applies a modularity clustering solution that accounts for edge information, whereas it neglects descriptive information. This means that it is able to discover clusters of highly connected nodes without paying attention to verifying the actual degree of the correlation (similarity) in the data associated with linked nodes grouped in the same cluster. This limit is overcome in the presented study by resorting to a contiguity constraint algorithm that derives clusters of correlated data over connected nodes by accounting for information in both the (descriptive) data space and the (edge) network structure; third, a new disagreement measure is proposed in our study: it applies to numeric targets and combines cluster knowledge and correlation-aware analysis of collective predictions.

Kuwadekar and Neville [26] presents a different strategy: the committee of classifiers is learned from descriptive data of various labeled sets. These sets are generated from the labeled part of the network by a subgraph resampling algorithm. The committee is used to predict the unlabeled nodes. A network-based score is computed, in order to choose to label nodes with low variance and low disagreement among their neighbors. This criterion is based on the idea that the most valuable unlabeled examples lie in high-density (unlabeled) regions and their predictions disagree the least with their immediate neighborhood and disagree the most with the common prediction of the committee. This algorithm both admits collective inference and accounts for the network information in the active example selection. However, it performs active learning differently from our algorithm, because it neglects the cluster structure of the network. Instead, the cluster knowledge represents an effective component of our solution, in order to guarantee the diversity of selected active examples and deal with slowly progressive network variation in the data.

[2] In the empirical study, a formulation of the collective active learning algorithm described in [7] is appropriately extended to be applied to regression problems. This is one of the competitor of the empirical evaluation. The accuracy results show that our algorithm generally outperforms this competitor.



(a) Block diagram



(b) Neighbourhood

Figure 1: Figure 1(a): The block diagram of the collective active learning strategy performed by CoNeRA includes: (1) the initialization phase (see the description in Section 4.2.1), (2) the iterative learning phase (see the description in Section 4.2.2) and (3) the final regression model learning phase (see the description in Section 4.2.3). Figure 1(b): Cluster-aware neighborhoods.

Table 1: **Symbols**. Description of the frequently used symbols.

symbol	meaning	symbol	meaning	symbol	meaning
N	Network node set	$\mathcal{E}$	Network edge set	$\mathcal{X}$	Descriptive variable space
$\mathcal{X}\mathbf{c}$	Collective variable space	$\mathcal{Y}$	Target (numeric) variable	$\mathcal{L}$	Labeled node set
$\mathcal{U}$	Unlabeled node set	l	Initial labeled set size	t	Sample pool size
$\mathcal{B}$	Budget	$\mathcal{O}$	Oracle		

Komurlu and Bilgic [25] formulate an active learning strategy to update the Dynamic Gaussian Bayesian model (DGBm) of the numeric univariate sensor readings of a wireless network. The DGBm models spatio-temporal correlations of sensor readings. Active learning uses an impact-based selection criterion, in order to dynamically choose sensors for the future observations, based on their impact on predicting others.

4. Collective Network Regression via Active Learning

CoNeRa is a novel collective active learning algorithm for network regression. It combines active learning, clustering and collective inference for regression in network data. The list of frequently used symbols is reported in Table 1, while the infographic of the implemented strategy is reported in Figure 1(a). Before starting the illustration in detail, the learning task is formally introduced.

4.1. Learning problem statement

The network regression problem is here formulated by considering: (1) A (directed, weighted) training *data network* $(\mathcal{N}, \mathcal{X}, \mathcal{Y}, \mathcal{E})$ collected from a *regression* problem domain. Specifically, $\mathcal{N}$ is a node set. $\mathcal{X}$ is a vector of m descriptive variables $(X_1, X_2, \ldots, X_m)$, so that a vector of descriptive values $(x_1, x_2, \ldots, x_m)$ is associated with a node $n \in \mathcal{N}$. $\mathcal{Y}$ denotes a numeric target variable, so that each $y \in \mathcal{Y}$ is a (initially unknown) numeric label for n. $\mathcal{E}$ is a weighted edge set that describes the network structure of $\mathcal{N}$. It is defined as follows: $\mathcal{E} = \{(n_i, n_j, d_{ij}) | (n_i, n_j) \in \mathcal{N} \times \mathcal{N}, d_{ij} \in \Re_0^+\}$. d_{ij} is a numeric, positive dissimilarity.

The lower d_{ij}, the higher the dependence of n_j on n_i throughout the network.[3] (2) A training set $\mathcal{U} \subseteq \mathcal{N}$ that is the set of unlabeled nodes considered for the training procedure. (3) An oracle $\mathcal{O}$ that provides the ground truth labels for $\mathcal{Y}$. (4) An initial labeled set size l and a sample pool size t. (5) A budget B that is the maximum number of nodes that can be labeled by the oracle. The considered task is to query the oracle, in order to acquire the ground truth labels of B nodes from $\mathcal{U}$ and learn a collective regression model accounting for the acquired labels. The strategy formulated to address this task starts assuming an empty labeled node set $\mathcal{L}$. It initially selects a sample of l nodes from $\mathcal{U}$ to be labeled by the oracle and moved from $\mathcal{U}$ to $\mathcal{L}$. It iterates a series of node selections from $\mathcal{U}$ for the oracle labeling (i.e. at each iterate a pool of t unlabeled nodes is selected, labeled by the oracle and moved from $\mathcal{U}$ to $\mathcal{L}$), so that the accuracy of the collective regression model on unseen data is maximized, after training it on current $\mathcal{L}$. This regression model is induced accounting for both descriptive information and collective knowledge on labeled nodes. During the iterative phase, collective knowledge is yielded from labels acquired by the oracle and labels predicted by the regression model. At each iteration, this regression model changes, as new labels are acquired and the collected knowledge is updated accordingly. The collective regression model for the testing procedure is the one learned from final $\mathcal{L}$ once the iterative phase stops (i.e. when $|\mathcal{L}| = B$).

4.2. Collective network active regression strategy

A top-level description of the collective active regression strategy is reported in Algorithm 1. The algorithm comprises an initialization phase, an iterative phase and a final regression model induction phase. A cluster structure of the network is learned during the initialization phase. This cluster structure is data-driven by a contiguity constraint on edge set $\mathcal{E}$. Each cluster collects nodes, whose descriptive data are correlated across $\mathcal{E}$. This cluster structure is used to select initial examples, whose labels can be acquired by querying oracle $\mathcal{O}$. These examples are randomly sampled per cluster. The number of examples sampled per cluster is proportional to the cluster size. In the iterative active learning phase, a collective regression model is learned from labeled part $\mathcal{L}$ of the training set (the set of nodes, whose labels have been provided by the oracle). A pool of unlabeled examples, which are unreliably predicted by the collective regression model, are selected per cluster. Their labels are acquired by querying the oracle. The new labeled examples augment $\mathcal{L}$ of the training set. In the final inductive phase, the collective regression model is learned from the labeled part of the training data constructed by fully using budget B. Hopefully, this regression model will be used to predict labels of testing nodes. A description of the three phases is reported in the following.

4.2.1. Initialization phase (Algorithm 20, lines 1-5)

There are various studies on active learning, which compute clustered data. They are mainly for classification problems. They perform either clustering, based only on the descriptive data [13], or clustering based only on the network structure (edge information) [7]. In the strategy illustrated here, clustering is performed

[3] It is assumed that every node is linked to itself in a network with dissimilarity 0 (i.e. $\forall n_i \in \mathcal{N}$: $(n_i, n_i, 0) \in \mathcal{E}$).

Input: $\mathcal{N}$: node set; $\mathcal{U} \subseteq \mathcal{N}$: unlabeled node set $\mathcal{E}$: edge set; $\mathcal{X}$: vector of descriptive variables; $\mathcal{Y}$: target variable; B: budget, l: initial labeled set size; t query pool size; $\mathcal{O}$: oracle;

Output: $(CoNeRa-D, CoNeRa-C)$: regression model

`// initialization phase`

$\mathcal{L} \leftarrow \oslash$

$\mathcal{C} \leftarrow$clustering$(\mathcal{U}, \mathcal{E})$

$\mathcal{L} \leftarrow$clusterBasedSampling$(\mathcal{U}, \mathcal{C}, l)$

$\mathcal{Y} \leftarrow$oracle$(\mathcal{L}, \mathcal{O})$

$\mathcal{U} \leftarrow \mathcal{U} - \mathcal{L}$

`// iterative phase`

repeat

  $CoNeRa-D \leftarrow$regressionModelLearning$(\mathcal{L}, \mathcal{X}, \mathcal{Y})$

  $\hat{\mathcal{Y}_D} \leftarrow$labeling$(\mathcal{U}, \mathcal{X}, CoNeRa-D)$

  $\mathcal{X}\mathbf{c} \leftarrow$collectiveVariableConstructing$(\mathcal{L}, \mathcal{Y}, \mathcal{U}, \hat{\mathcal{Y}_D}, \mathcal{E})$

  $CoNeRa-C \leftarrow$regressionModelLearning$(\mathcal{L}, \mathcal{X}, \mathcal{X}\mathbf{c}, \mathcal{Y})$

  $\hat{\mathcal{Y}_C} \leftarrow$labeling$(\mathcal{U}, \mathcal{X}, CoNeRa-C)$

  $\mathcal{Q} \leftarrow$ activeNodeSelecting$(\mathcal{U}, \mathcal{Y}, \mathcal{L}, \hat{\mathcal{Y}_C}, \mathcal{E}, \mathcal{C}, t)$

  $\mathcal{U} \leftarrow \mathcal{U} - \mathcal{Q}$

  $\mathcal{Y} \leftarrow$oracle$(Q, \mathcal{O})$

  $\mathcal{L} \leftarrow \mathcal{L} \cup (\mathcal{Q}, \mathcal{Y})$

until *(size*$(\mathcal{L}) = B$*)*;

`// inducing the final collective regression model by fully using budget` B

$CoNeRa-D \leftarrow$regressionModelLearning$(\mathcal{L}, \mathcal{X}, \mathcal{Y})$

$\hat{\mathcal{Y}_D} \leftarrow$labeling$(\mathcal{U}, \mathcal{X}, CoNeRa-D)$

$\mathcal{X}\mathbf{c} \leftarrow$collectiveVariableConstructing$(\mathcal{L}, \mathcal{Y}, \mathcal{U}, \hat{\mathcal{Y}_D}, \mathcal{E})$

$CoNeRa-C \leftarrow$regressionModelLearning$(\mathcal{L}, \mathcal{X}, \mathcal{X}\mathbf{c}, \mathcal{Y})$

Algorithm 1: Collective network active regression

in the descriptive space and in the network structure, simultaneously. The goal is to determine a cluster structure that depicts network regions, where the distribution of descriptive data is smoothly continuous, with boundaries possibly marked by sharp discontinuities, observable in the descriptive data correlation. This cluster structure is used to initialize the sample for querying the oracle. This clustering step is done by a contiguity constrained clustering algorithm that evaluates the similarity of the descriptive data at linked nodes. The justification of this constraint-based solution is that it can fit requirements of learning under correlation and take advantage of a network contiguity constraint between nodes to reduce the number

of possible solutions. It can also force the clustering algorithm to converge fast onto largely similar areal boundaries. This point of view, originally investigated for clustering in geophysical networks [4, 5], is here investigated in the contest of an active learning strategy applied to various kinds of networks. Clustering algorithm and the cluster-based labeled set initialization are described in the following.

Contiguity constraint clustering (Algorithm 20, line 2). The computation starts with $\mathcal{U}$ assigning $k = 1$, where k enumerates the computed clusters. The construction of a new cluster C_k starts with a seed unclustered node n_i. This seed is randomly chosen from unlabeled nodes ($n_i \in \mathcal{U}$), which are still un-assigned to any cluster. Then n_i is added to C_k, while C_k is expanded by using n_i as the seed of the expansion process. C_k is added to the cluster structure $\mathcal{C}$. k is incremented by one and the clustering process is iteratively repeated until all the nodes are assigned to a cluster. The expansion of C_k is driven by a seed node n_i and it is recursively defined. First, the neighborhood $\eta(n_i)$ having a seed n_i is constructed by considering the unclustered nodes n_j, which are directly reachable from n_i in $\mathcal{E}$. Formally,

$$\eta(n_i) = \{n_j \in \mathcal{U} | (n_i, n_j, d_{ij}) \in \mathcal{E}\}. \tag{1}$$

Then candidate cluster $tempC = C_k \cup \eta(n_i)$ is computed. The cluster homogeneity property $h(tempC)$ is evaluated on candidate cluster $tempC$ spanned on the descriptive space $\mathcal{X}$. Formally,

$$h(tempC) = \begin{cases} true & if \max\limits_{n_i \in tempC, n_j \in tempC} scaledDistance(n_i, n_j) \leq \psi \\ false & otherwise \end{cases}, \tag{2}$$

where ψ is a user-defined threshold ($\psi \in [0, 1]$), the $scaledDistance(n_i, n_j)$ is the Euclidean distance computed between vectors $(\widetilde{x_1(n_i)}, \widetilde{x_2(n_i)}, \ldots, \widetilde{x_m(n_i)})$ and $(\widetilde{x_1(n_j)}, \widetilde{x_2(n_j)}, \ldots, \widetilde{x_m(n_j)})$ associated with n_i and n_j, respectively. In particular, $\widetilde{x(n)}$ denotes the value of a descriptive variable $X \in \mathcal{X}$, associated with node $n \in \mathcal{U}$ and scaled between 0 and 1, according to $\widetilde{x(n)} = \frac{x(n) - \min\limits_{n \in \mathcal{U}} x(n)}{\max\limits_{n \in \mathcal{U}} x(n) - \min\limits_{n \in \mathcal{U}} x(n)}$. Two cases are distinguished: in the former case, $tempC$ satisfies the homogeneity property and then, nodes of $\eta(n_i)$ are clustered into C_k. In the latter case, $tempC$ does not satisfy the homogeneity property and the addition of each node of $\eta(n_i)$ to C_k is evaluated node-by-node. In both cases, nodes newly clustered in C_k are iteratively chosen as seeds to continue the expansion process. The expansion process stops, if no new node is added to the cluster.

Labeled set initialization (Algorithm 20, lines 3-5). The initialization phase is performed for three reasons: (1) to identify an initial sample of nodes from $\mathcal{U}$, (2) to require the ground-truth labels of sampled nodes to oracle $\mathcal{O}$ and (3) to move definitely sampled nodes with their labels from $\mathcal{U}$ to $\mathcal{L}$. A cluster-based sampling procedure [40] is employed, in order to take into account cluster structure $\mathcal{C}$ for network. Procedurally, the simple random sampling is used separately in every cluster with the numbers of nodes selected from different clusters proportional to the cluster size. Let l be the total number of nodes to be sampled for the initialization phase, C_k be a cluster ($C_k \in \mathcal{C}$) that groups $|C_k|$ nodes, n be the total number of nodes

actually distributed across clusters of $\mathcal{C}$ (i.e. $n = \sum_{C_k \in \mathcal{C}} |C_k|$), the number of nodes sampled from C_k via simple random sampling is $\frac{|C_k| \times l}{n}$. The simple random sampling is done without replacement, that is, one deliberately avoids choosing any nodes of every cluster more than once.

4.2.2. Iterative phase (Algorithm 20, lines 6-16)

This phase iteratively induces a collective regression model from $\mathcal{L}$ and queries $\mathcal{U}$, in order to augment $\mathcal{L}$. At each iteration, the output of the active learning query is a pool of t unlabeled nodes which are labeled by the oracle and moved from $\mathcal{U}$ to $\mathcal{L}$. The iterative phase stops when budget B is fully used so that $|\mathcal{L}| = B$.

Collective regression model induction (Algorithm 20, lines 7-11). The induction of a collective regression model is three-stepped: (1) a descriptive regression model ($CoNeRa - D$) is induced from $\mathcal{L}$ spanned on $\mathcal{X} \times \mathcal{Y}$ and used to predict descriptive-aware labels $\hat{\mathcal{Y}_D}$ of nodes in $\mathcal{U}$; (2) labels acquired by querying the oracle and labels predicted by the descriptive regression model feed the inference to construct a vector of collective variables $\mathcal{X}\mathbf{c}$ and (3) a collective regression model ($CoNeRa-C$) is finally induced from $\mathcal{L}$ spanned on $\mathcal{X} \times \mathcal{X}\mathbf{c} \times \mathcal{Y}$, and used to predict collective-aware labels $\hat{\mathcal{Y}_C}$ of nodes in $\mathcal{U}$. Any regression algorithm can be selected as a base learner of both the descriptive and collective regression model.

The vector of collective variables is synthesized by taking into account the results of recent studies [38, 41], which investigate the computation of various summarizarization statistics to handle the correlation property of the numeric target over local neighborhoods of the network data. These statistics are formulated assuming a weakly stationary model of neighborhood data. They correspond to second order characteristics of the underlying data, averaged over linked nodes (in accordance with the hypothesis that labels which are linked manifest a coherent correlation). In particular, computed collective variables describe the typical value of the target variable (i.e. mean - μ and weighted mean - ω), as well as the variability of the variable (deviation - σ) within the neighborhood. Procedurally, for each node $n_i \in \mathcal{L} \cup \mathcal{U}$), the vector of collective variables $\mathcal{X}\mathbf{c}(n_i) = (\mu(n_i), \omega(n_i), \sigma(n_i))$ is computed as follows. Let $\eta(n_i)$ be the neighborhood with seed n_i determined according to Formula 1), $|\eta(n_i)|$ be the size of $\eta(n_i)$, $y'(n_i)$ be the label acquired by the oracle if n_i is an active example (i.e. $y'(n_i) = y(n_i)$ if $n_i \in \mathcal{L}$) and the label predicted by $CoNeRa - D$ otherwise (i.e. $y'(n_i) = \hat{y_D}(n_i)$ if $n_i \in \mathcal{U}$), the following calculations are performed:

- *Mean variable* $\mu(n_i)$ is computed as the average value of the labels grouped in the neighborhood. It is computed by the Distance Interpolate μ[4] so that $\mu(n_i) = \frac{1}{|\eta(n_i)|} \sum_{n_j \in \eta(n_i)} y'(n_i)$.
- *Weighted mean variable* $\omega(n_i)$ is computed as the weighted average of the labels grouped in the neighborhood. It is computed by the Inverse Distance Weighted Interpolate ω. This is a variant of the

[4] Distance Interpolate is a form of Nearest Neighboring predictive modeling that is here used as a mechanism to construct collective variables. These variables will be subsequently processed, in order to learn the collective regression models which are actually used to predict unseen data.

Distance Interpolate, which weights the contribution of the neighbors by the weights derived from dissimilarities associated with the neighborhood edges. Formally, $\omega(n_i) = \frac{1}{\sum_{n_j \in \eta(n_i)} w_{ij}} \sum_{n_j \in \eta(n_i)} w_{ij} y'(n_i)$.

Similarly to the Distance Interpolate, the Inverse Distance Weighted Interpolate explicitly makes the assumption that target values on nodes that are close (linked) to n_i are more alike than those that are farther apart. However, the Distance Interpolate considers that all neighbors contribute equally to the measure of the target correlation summary around n_i. Differently, the Inverse Distance Weighted Interpolate assumes that each target value has a local influence that diminishes with the link dissimilarity. It gives greater weights to nodes closest to n_i and the weights diminish as a function of link dissimilarity. A popular choice is to use the Inverse power function as a weighting function [28, 29, 41]: weights are proportional to the inverse of the dissimilarity (between the linked nodes) raised to a power value (2 by default).[5] In this study, weights are computed proportionally to the dissimilarities, enclosed in the edge structure of the network. Formally, every weight w_{ij} is computed as follows:

$$w_{ij} = \frac{1}{(d_{ij}+1)^2}, \tag{3}$$

where d_{ij} is the dissimilarity associated with the edge from n_i to n_j in $\mathcal{E}$. Correction $(d_{ij}+1)$ is done as dissimilarities may also be zero valued, in order to avoid division by zero.

- *Deviation variable* $\sigma(n_i)$ is computed, in order to capture the notion of the spread of the labels within a neighborhood. It is calculated as follows: $\sigma(n_i) = \sqrt{\frac{1}{|\eta(n_i)|} \sum_{n_j \in \eta(n_i)} (y'(n_i) - y'(n_j))^2}$.

Disagreement measurement and labeled set augmentation (Algorithm 20, lines 12-15). The unlabeled nodes are selected for labeling and this choice is based on a disagreement analysis that looks for the scarcity of correlation among linked labels. The disagreement measure is computed for each node of $\mathcal{U}$. This computation is performed by taking into account: (1) labels $\hat{Y}_C$ of the nodes of $\mathcal{U}$ predicted by collective regression model $CoNeRA - C$ induced in the current iteration; (2) ground truth labels Y of the nodes of $\mathcal{L}$ acquired by the oracle along both the initialization and the already performed iterations; (3) cluster structure $\mathcal{C}$ of the network data computed in the initialization phase. Let $n_i \in \mathcal{U}$ be an unlabeled node, $\eta(n_i)$ be the neighborhood with seed n_i, $\mathcal{C}|_{n_i}$ be a subset of the cluster structure ($\mathcal{C}|_{n_i} \subseteq \mathcal{C}$), defined so that every cluster $C \in \mathcal{C}|_{n_i}$ groups at least one node of neighborhood $\eta(n_i)$ (i.e. $\forall C \in \mathcal{C}|_{n_i} : \exists n_j \in \eta(n_i), n_j \in C$). Neighborhood $\eta(n_i)$ is projected onto clusters of $\mathcal{C}|_{n_i}$, in order to associate a node with its cluster-aware neighborhoods (see Figure 1(b)), that is: $n_i \mapsto \eta(n_i) \mapsto \{\eta(n_i, C)\}_{C \in \mathcal{C}|_{n_i}}$, where $\eta(n_i, C) = \{n_i\} \cup \{n_j \in \eta(n_i) | n_j \in C\}$. This cluster-aware projection of a neighborhood is interesting along

[5] Several weighting functions (e.g. Gaussian-like similarity measure or the Bisquare density function [50]) can be considered as an alternative to the Inverse power distance. All follow the same rationale: as the dissimilarities increase, the weights decrease and the influences of the neighbor labels diminish as well.

the boundary between clusters, where linked nodes, which express a significant variability in correlation dependencies, are grouped in separate clusters. In these network areas, a neighborhood will group linked nodes that describe separate local correlation phenomena simultaneously. As the disagreement of a node is here measured, in order to quantify the correlation of the labels surrounding a target label. The disagreement is actually computed on the cluster-aware neighborhoods of a node. This allows us to define a disagreement measure that naturally fits variability in correlation dependencies along the cluster boundary. Formally, $disagreement(n_i) = \min_{C \in \mathcal{C}|_{n_i}} disagreement(\eta(n_i, C))$, where $disagreement(\eta(n_i, C))$ measures the spread of labels associated with neighbors that belong to cluster C with respect to the label of seed n_i. Formally:

$$disagreement(\eta(n_i, C)) = \frac{1}{\sum_{n_j \in \eta(n_i, C))} w_{ij}} \sum_{n_j \in \eta(n_i, C))} w_{ij} |y'(n_i) - y'(n_j)|, \tag{4}$$

where $y'(n_i)$ is acquired by the oracle if $n_i \in \mathcal{L}$ (i.e. $y'(n_i) = y(n_i)$), while $y'(n_i)$ is predicted by the collective regression model induced from current $\mathcal{L}$, if $n_i \in \mathcal{U}$ (i.e. $y'(n_i) = \hat{y_C}(n_i)$). w_{ij} is computed with Formula 3.

By taking into account the disagreement information, the pool of unlabeled nodes whose labels will be acquired by the oracle are finally selected cluster by cluster. Procedurally, all clusters in $\mathcal{C}$ are initially sorted in descending order, according to a cluster disagreement score, that is, the average disagreement score in the cluster divided by the number of already labeled examples from the cluster. This cluster sorting criterion follows the rationale behind the sorting criterion described in [7], although the two strategies use a different formulation of the cluster structure and of the disagreement measure. Specifically, it allows us to avoid over-investing in the clusters that have already been explored for the label acquisition.[6] The clusters associated with equal disagreement scores are sorted randomly. Subsequently, for each cluster $C_k \in \mathcal{C}$, the unlabeled nodes currently grouped in C_k are sorted in descending order according to their disagreement value (computed with Formula 4). Again, two nodes with equal disagreement within a cluster are considered equally useful for the labeling phase, so they are randomly sorted. The clusters are repeatedly visited with the defined order. At each iteration of the visit, the unlabeled node with the highest disagreement in the cluster is selected, it is then marked as visited (so that it cannot be selected in the subsequent iterations of the visiting procedure), removed from $\mathcal{U}$ and temporally put into a temporary pool set $\mathcal{Q}$. The visiting procedure is repeated until t distinct unlabeled nodes have been added to $\mathcal{Q}$. Finally, the oracle is queried so that it can yield a label for each node in $\mathcal{Q}$. Once $\mathcal{Q}$ is fully labeled, the newly labeled nodes of $\mathcal{Q}$ are definitely added to labeled set $\mathcal{L}$. It is noteworthy that according to the illustrated procedure, labels of the entire pool set are acquired in one query only after the pool set is fully populated (i.e. $\mathcal{Q}$ contains t unlabeled nodes). This avoids the following case: if one unlabeled node gets labeled, it will produce changes in the neighborhood variation and lead to erroneous selection of nodes to be labeled. In particular, $\mathcal{Q}$ will be

[6] The cluster ordering is crucial when the number of nodes t to be sampled at each iterative phase is lower than the number of clusters in the cluster structure: only t top-ranked clusters will contribute to the sample pool with one unlabeled node selected per cluster.

fully populated without any change in the neighborhood structure. The neighborhood structure effectively changes only after that target values are acquired for all nodes in current pool set $\mathcal{Q}$.

4.2.3. Final regression model induction phase (Algorithm 20, lines 17-20)

Let us consider a final labeled set $\mathcal{L}$ as it is populated with labels acquired by the oracle during the initialization and iterative phases. The final descriptive regression model $CoNeRa-D$ is induced from $\mathcal{L}$ as it is spanned over a descriptive space $\mathcal{X}$ and a target variable $\mathcal{Y}$. It is used to predict labels $\hat{\mathcal{Y}_D}$, which are still unknown in a current unlabeled set $\mathcal{U}$. Using ground truth labels, acquired by the oracle for $\mathcal{L}$ and labels predicted by $CoNeRa-D$ for $\mathcal{U}$, the collective variables (mean μ , weighted mean ω and deviation σ) are synthesized over the entire training network. A collective regression model $CoNeRa-C$ is finally induced from $\mathcal{L}$ as it is spanned on the descriptive space $\mathcal{X}$, the collective space $\mathcal{X}\mathbf{c}$ and the target variable $\mathcal{Y}$. Regression models $CoNeRa-D$ and $CoNeRa-C$ produced in this final phase can then be used to predict unseen target values of any new testing data network that is acquired in the same domain condition as the training network (i.e. every node of the testing network is associated with a vector of variables in $\mathcal{X}$). A three-stepped prediction methodology is used: first, every node of the testing data network is initially associated with the descriptive-aware label that can be predicted by $CoNeRa-D$ using only the descriptive information associated with each node; second, the descriptive-aware predicted labels are considered to construct the variables of a collective space $\mathcal{X}\mathbf{c}$ over the testing data network. Finally, the collective-aware labels are predicted by $CoNeRa-C$ using both descriptive and collective information.

4.3. Time complexity

Basics. Let U be the number of nodes in the training set, m be the number of descriptive variables, E be the number of edges, ϵ be the average size of a neighborhood, K be the number of clusters in the cluster structure, l be the number of nodes in the labeled set populated in the initialization phase, t be the number of nodes in the sample pool populated at each iteration of the iterative phase, B be the budget, IT be the number of iterations performed in the iterative phase (i.e. $IT = (B-l)/t$) and $\lambda(\cdot)$ be the time complexity of the base regression learner. It is noteworthy that $\lambda(\cdot)$ is a function of both the labeled set size and the variable space size processed by the learner to build the regression model.

Initialization phase. The time cost of the contiguity constrained clustering is $U\epsilon m$, that is, the cost of evaluating the homogeneity property on each neighborhood along the cluster construction. The cost of sorting clusters by the descending order with respect to the cluster size is $K \log K$, while the cost of selecting l samples from the cluster-based representation of the training set is l. Therefore, the total cost of the initialization phase is $U\epsilon m + K \log K + l$, that is, $\mathrm{O}(U\epsilon m + K \log K)$, by considering that $l \leq U$.

Iterative phase. The time cost of learning the collective regression model depends on the cost of: (1) inducing the descriptive regression model (that is, $\lambda(B, m)$ in the worst case , i.e. the labeled set contains the full budget of examples), (2) constructing the collective variables (that is, $U\epsilon$ for each collective variable) and (3)

Table 2: **Data description**. For each dataset, the data domain, the number of nodes - N, the number of edges - E, the number of descriptive attributes - m, as well as the target attribute are reported.

Data	Domain	N	E	m	Data	Domain	N	E	m
Irs	spatial-remote sensing	2830	766827	28	Mf	spatial-agriculture	817	31995	4
Movies	social	415	123163	12	Ms	spatial-agriculture	817	31995	4
NCEP	spatial-environmental	2376	378276	9	PubmedMellitus	social	6170	64055	19
SCH	spatial-census	7229	21690311	6	SOILmoisture	spatial-environmental	3888	827100	33
Twitter	social	55	253	3	Vegetation	spatial-environmental	3888	827100	33
YouTube	social	13723	180976	3					

inducing the collective regression model (that is, $\lambda(B, m+3)$) . The total cost is asymptotically $\lambda(B, m)+U\epsilon$. The time cost of the disagreement measurement is $U\epsilon K$ in the worst case, that is, each neighborhood is projected on K clusters. The time cost of the labeled set augmentation depends on the cost of: (1) sorting the clusters of $\mathcal{C}$ according to the cluster disagreement score (that is, $UK+K\log K$ in the worst case, i.e. the cluster size is less than U), (2) sorting the unlabeled nodes per cluster according to the node disagreement (that is $KU\log U$ in the worst case, i.e. the cluster size is less than U), (3) selecting t examples for labeling (that is, t). Therefore, the total cost of an iteration is $\lambda(B, m)+U\epsilon+U\epsilon K+UK+K\log K+KU\log U+t$, that is, $\mathrm{O}(\lambda(B, m)+U\epsilon K+KU\log U)$, by considering that $K \leq U$ and $t \leq U$. The total cost of the iterative phase is $\mathrm{O}(IT\cdot(\lambda(B, m)+U\epsilon K+KU\log U))$.

Final regression model induction phase. The time cost of learning the collective regression model from the final labeled set is $\mathrm{O}(\lambda(B, m)+U\epsilon)$.

Total time complexity. The total time cost of the presented collective active learning strategy is:

$$\underbrace{U\epsilon m+K\log K}_{initialization\ phase}+\underbrace{IT\cdot(\lambda(B, m)+U\epsilon K+KU\log U)}_{iterative\ phase}+\underbrace{\lambda(B, m)+U\epsilon}_{final\ model\ induction},$$

that is, $\mathrm{O}(U\epsilon m+K\log K+IT\cdot(\lambda(B, m)+U\epsilon K+KU\log U))$.

5. Experimental evaluation and discussion

The efficacy of CoNeRa is initially investigated considering several network regression datasets, collected at a specific time point. A brief description of these network data is reported in Section 5.1. The experimental methodology is presented in Section 5.2, the compared algorithms and their parameters are illustrated in Section 5.3, while the empirical results are discussed in Section 5.4. Subsequently, the performance of CoNeRa is explored over time (see Section 5.5), as well as on a few network regression problems considered in the recent active learning literature (see Section 5.6). All experiments are run on ReCaS cloud, CPU 1:8 @ 2Ghz 2,16.0 GB RAM, running Ubuntu 14.04.4 (GNULinux 3.13.0-39-generic x86_64).

5.1. Datasets

Eleven social and spatial network data collections are used. The characteristics of these data sets are summarized in Table 2. In each data collection, a number of descriptive attributes, as well as the numeric

label are associated with the nodes. The nodes are linked by weighted edges. Weights represent dissimilarities between nodes. A brief description of both social and spatial networks is reported in Sections 5.1.1 and 5.1.2. For each data network, the correlation property of the target variable is analyzed in Section 5.1.3.

5.1.1. Social network data

Movies dataset contains ratings, collected during the period 1997-1998, given to movies by users of the online movie recommender service Movielens [29, 49]. Each movie is described by the critics' rating (target variable), as well as the all/top/audience critics's ratings (average scores, numbers of reviews/fresh scores/rotten scores) from the Rotten Tomatoes film review aggregator. The selected users had rated at least 20 movies. Only pairs of movies ranked together by a single user are considered. The nodes represent the movies whereas the weighted edges represent the dissimilarity of the ratings given by the users.

Twitter Maternal Health dataset contains the top Twitter users who mentioned maternal mortality on August 25, 2010 [49]. It also includes the number of posts (tweets) of a user (target variable), the user's registration date on Twitter and its time zone, the number of tweets (posted by other users) that the user marked as "favorites", as well as the number of "following" and "followed" on Twitter. Only pairs of users having a "following" relation are considered. The nodes are the users, whereas the weighted edges are the "following" relation between the Twitter users. Edges are associated with 1-valued weights.

YouTube dataset refers to the set of user profiles crawled on December, 2008 from YouTube [52]. Each user is described by count of the subscribers (target variable) the count of contacts, count of the subscriptions and count of favorite videos. The nodes are the users, whereas the weighted edges represent the existence of interactions between two user profiles. Edges are associated with 1-valued weights.

PubmedMellitus dataset contains scientific publications of the collection PubMed Diabetes [45], which contain the word "mellitus". Each publication is described by a TF/IDF weighted word vector from a dictionary that consists of 20 unique stemmed words and that has no stop-word. The target is the TF/IDF value of the word "mellitus". In this study, the nodes are the scientific publications. An edge connects node n_i to node n_j, if publication n_i cites publication n_j. Edges are associated with 1-valued weights.

5.1.2. Spatial network data

Irs dataset comprises multi-temporal remote sensing data of the Kras region in Western Slovenia obtained from EEA Corine Image 2006 [48]. The target variable is the forest standing height. The descriptive variables are the minimum reflectance, the maximum reflectance, the average reflectance and the standard deviation of reflectance for each IRS sensor channel, aggregated over fine and coarse image segments. The coordinates of the centroids of the image segments are assigned to the nodes of the network structure.

Mf and Ms datasets contain measurements of pollen dispersal (crossover) rates from two lines of plants (target variables), that is, the transgenic male-fertile (MF) and the non-transgenic male-sterile (MS) line of oilseed rape [29, 49]. The coordinates of each sample point are collected. The descriptive variables are the following: the cardinal direction and the distance of the sample point from the center of the donor field,

the visual angle between the sample plot and the donor field, and the shortest distance between the plot and the nearest edge of the donor field. The nodes in the network structures of both Mf and Ms are the sampling points.

NCEP dataset refers to data collected by the NCEP/NCAR Reanalysis Project [23]. Data were recorded daily from December, 1, 2015 to March,1 2016 by 2376 sensors. Each sensor recorded the following measurements: the precipitable water content (target), the air temperature, the surface lifted index, the best lifted index, the surface pressure, the potential temperature, the relative humidity, the sea level pressure, the eastward wind and the northward wind. The sensors represent the nodes and the descriptive variables associated with the nodes are computed as the average of the geophysical measurements recorded in the period of observation.

SCH dataset concerns Southern California Housing (SCH) census data, provided by the 1990 Census and aggregated at the level of block groups that, on average, include 1425.5 individuals living in a geographically compact area [24]. The target variable is the median house value, while the descriptive variables are the median income, housing median age, total rooms, total bedrooms, population and households in a neighborhood. The coordinates of the centroids of the block groups are collected in the dataset and associated with the nodes of the neighborhood structures.

SOILmoisture and Vegetation datasets contain data studied by the Real Time Ocean Forecasting System Project [46]. The data were recorded on September, 17 2014 by 3888 sensors. Each sensor measured the values of 34 properties in the various categories (Land Surface, Oceanography, Meteorology and Hydrology). For both datasets, the nodes are the sensors. In SOILmoisture the target is "Soil moisture parameter in canopy conductance surface numeric", while in Vegetation the target is "Vegetation surface".

As suggested in [50], the edges of the network structure of these datasets are defined by assuming that every node is actually linked to the nodes falling in an influence sphere, having a radius b. In particular, node n_i is linked to node n_j, if $EuclideanDistance(n_i, n_j) \leq b$. The edge weight is the Euclidean distance between the linked nodes' spatial coordinates. The neighborhood radius is defined according to the considerations reported in [4, 6]. Specifically, radius b is fixed as $b = .1 \times \max_{n_i, n_j \in \mathcal{N}} EuclideanDistance(n_i, n_j)$.

5.1.3. Target correlation analysis

The correlation analysis is performed, in order to quantify whether the linked nodes tend to have correlated values of the target variable throughout the network. The Global Moran's I [34] is computed, in order to summarize the overall pattern of network dependence of target data into a single indicator (see results in Table 3). This measure is borrowed from spatial data analysis, but also fits the general structure of network data [49]. It uses the weight matrix that reflects the strength of the connection between linked nodes. The Global Moran's I is defined as $I_Y = \frac{1}{W \sum_{n_i} (y(n_i) - \overline{Y})^2} N \sum_{n_i} \sum_{n_j} w_{ij} (y(n_i) - \overline{Y})(y(n_j) - \overline{Y})$, where N is the number of nodes indexed by i and j; Y is the variable of interest; $y(n_i)$ and $y(n_j)$ are the values of the variable Y for the nodes n_i and n_j, respectively; $\overline{Y}$ is the overall average of Y in the entire network; and

Table 3: **Target data correlation analysis**. For each dataset, both the Global Moran's I and the expected value of Moran's I - $E(I)$ under the null hypothesis of no network correlation are reported for the target. Values $I \geq E(I)$ confirm the presence of positive in-network correlation of the target.

Data	I	$E(I)$	Data	I	$E(I)$	Data	I	$E(I)$
Irs	0.543455	-0.0003533	Mf	0.255617	-0.0012254	Movies	-0.078004	-0.0024154
Ms	0.347508	-0.0012254	NCEP	0.949875	-0.0004210	PubmedMellitus	0.253763	-0.0001621
SCH	0.104382	-0.0001383	SOILmoisture	0.766608	-0.0002572	Twitter	0.067496	-0.0185185
Vegetation	0.704051	-0.0002572	YouTube	0.078236	-0.0000728			

$W = \sum_{i,j} w_{ij}$ is the sum of weights, where the higher each weight w_{ij}, the closer the nodes n_i and n_j across the network. The dissimilarity decay function considered for assigning weights during collective inference is used to define the weighting matrix (i.e. w_{ij} is computed according to Formula 3). The values of the Global Moran's I generally range from -1 to +1.[7] The expected value of the Global Moran's I (calculated assuming the values of Y distributed randomly) is $E(I_Y) = \frac{-1}{N-1}$. Dubin [17] highlights that $I \geq E(I)$ indicates positive correlation (i.e. similar values measured at linked nodes). Based on this theory, the values of both I and $E(I)$, reported in Table 3, confirm the validity of the collective hypothesis concerning the positive correlation of the target data throughout a network. Although the correlation property is, generally, observed both in social and spatial networks, the degree of correlation is higher in spatial networks (see IRS, NCEP, SOILmoisture and Vegetation) than in social networks. In spatial networks, this quantitative analysis can also be supported by a visual inspection of the correlation property. Figures 2(a)-2(c) show the target values measured for NCEP, Vegetation and SCH, plotted along the spatial coordinates of the network nodes. The maps show that the target values that are close to each other in space tend to have similar values. When plotted on a map, they form a nice smooth continuous surface without sharp edges and discontinuities.This phenomenon is stronger in NCEP and Vegetation, where I is 0.949875 and 0.704051 and is, respectively, weaker (but still visible) in SCH where I is 0.104382.

5.2. Experimental methodology

This experimental study is carried out adopting the evaluation methodology that is introduced in [7]. For each dataset, the five-fold cross validation (5-CV) is performed, by randomly partitioning the dataset into five folds. During each learning phase, one fold of the cross-validation is used as a testing set (20% of dataset), while the hold-out folds are used as a training set (80% of dataset). The regression model learned from the training set is used to predict the testing set. As promoted in [7], the training phase process is performed after removing the edges from the nodes in the testing fold, in order to avoid a contamination of the training phase. The samples from the remaining four folds (training set) have their labels initially hidden. They are treated as the unlabeled set $\mathcal{U}$ from which the active learner selects l samples, which

[7] This case can be guaranteed by resorting to the row-standardization of the weights during the calculus of the Global Moran's I [17].

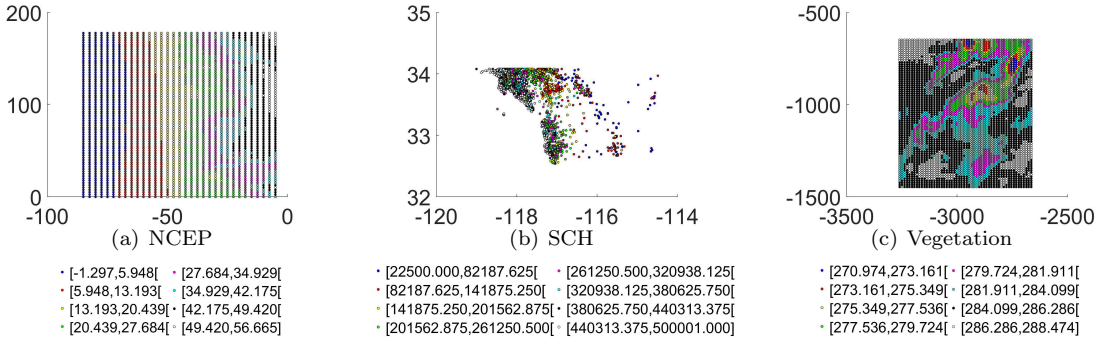


Figure 2: Target data visualized along the spatial coordinates (axis X and axis Y) of the nodes.

are labeled by the oracle, in order to initialize the labeled set $\mathcal{L}$. During the iterative learning process, a regression model is learned from the labeled set constructed at each iteration. This regression model can be used to predict labels of the testing set. As promoted in [7], data labeled during the learning phase are available during testing. However, to ensure that all algorithms are tested on the same set of samples, they are evaluated only on the held-out test set after the edges between the testing nodes and the rest of the data are restored. As the cluster-based sampling procedure employed during the initialization phase of the collective active learning strategy resorts to a simple random sampling algorithm, clustering may output different initializations of $\mathcal{L}$. This may happen due to the random selection also even if several trials are repeated with identical training set and identical cluster structure. To account for this, the experiment is repeated five times, for each algorithm, for each fold (by changing the initialization of $\mathcal{L}$):[8] thus each point of the learning curves described in this study is an average on 25 runs. In particular, the active learning strategy is analyzed along its error curve. Each point of the accuracy curve is the average of the Root Mean Squared Error (RMSE) computed on testing sets at each iteration of the active learning process.

5.3. Compared algorithms and parameters

The algorithms used for the comparison are here listed. (1) Random, DAL [16, 39] and PAL [16, 39] which consider node information, but disregard edge information. They use ad-hoc defined sampling criteria, which abstain from computing collective information. (2) ALFNET-R, which accounts for network-aware collective information. It modifies the active learning strategy presented in [7] for network classification problems, in order to comply with network regression problems. (3) CoNeRa, which includes network-aware collective information. It implements the active learning strategy illustrated in Section 4. Specifically, Random, DAL and PAL resort to a random selection of the initial $l-$sized labeled set, while ALFNET-R and CoNeRa resort

[8] There is no special restriction imposed on the sampling procedure so it may also happen that few examples are shared between two distinct initializations of the same training set. In any case, the random selection contributes to output initializations which should differ in the most part of the examples.

to a cluster-based definition of the initial labeled set. In particular, ALFNET-R uses the graph-clustering model of the network, as described in [7], while CoNeRa uses the constraint-clustering model of the network, as described in Section 4.2. Random is the naive baseline competitor that randomly selects each sample pool from the training data. DAL [16, 39] and PAL [16, 39] are two state-of-the-art active learners, designed for regression problems without a network structure. In DAL, the sample selection criterion identifies the pool of unlabeled examples, which are the farthest from the examples in the current labeled set. These distances are computed in the descriptive variable space (see details in Section 3). In PAL, the sample selection criterion identifies the pool of unlabeled examples, which are characterized by the greatest disagreements among the pool of the regression hypotheses learned from the datasets, obtained by sampling the labeled set regularly (see details in Section 3). ALFNET-R is a variant of ALFNET [7] (see details in Section 3). It uses a base regressor instead of a base classifier. It computes the collective variables of a numeric target, instead of the collective variables of a categorical target. It computes the standard deviation, instead of the entropy, in order to measure the disagreement among labels predicted by the collective regressor, the descriptive regressor and the graph-based clustering model. Finally, Random, DAL and PAL use a regression model learned from labeled data spanned on the descriptive space, in order to predict labels of testing sets, while ALFNET-R and CoNeRa use a regression model learned from labeled data, spanned on both the descriptive and collective space, in order to predict labels of testing sets.

The compared algorithms are run with $l = 50$, $t = 10$ and $B = 190$ on Irs, Mf, Ms, Movies, NCEP/NCAR, SOILmoisture and Vegetation; $l = 10$, $t = 3$ and $B = 50$ on Twitter; $l = 500$, $t = 50$ and $B = 1200$ on SCH and PubmedMellitus; $l = 1000$, $t = 100$ and $B = 2400$ on YouTube. For SCH, SOILmoisture and YouTube, the sensitivity of both accuracy and scalability of CoNeRa is evaluated, by varying l between 300, 400, 500, 600 and 700 and t between 25, 50 and 100 in SCH, l between 10, 30, 50, 70 and 90 and t between 5, 10 and 20 in SOILmoisture. An experiment is also performed by varying l between 600, 800, 1000, 1200 and 1400 and t between 50, 100 and 200 in YouTube.[9] The contiguity constraint clustering is run with $\psi = 0.3$. The inductive M5' [51][10] is used as a base regressor of the compared active learning strategies. This choice is motivated by several studies reported in the literature (e.g. [1]), which show that M5' can be applied to several regression problems with great success, outperforming several other inductive regression algorithms, in terms of accuracy and efficiency. However, this choice does not exclude the possibility of using any other regression algorithm as a base learner of our algorithm. Software and network data used in the experimental study are available for download at `http://www.di.uniba.it/~appice/software/CoNeRa/index.htm`. The implementation of CoNeRa, as well as of its competitors, is written in Java except for the clustering step of ALFNET-R that is written in R.

[9] For scalability investigation, the considered datasets are: SCH, that is, the network with the highest number of edges, SOILmoisture, that is, the network with the highest number of descriptive variables and YouTube, that is, the network with the highest number of nodes in this study.

[10] The Java implementation of M5' included in the WEKA toolkit [51] is used. The default configuration setup is considered together with the pruning option enabled.

5.4. Results and discussion

The experimental study is structured as follows. First, CoNeRa is compared to various competitive baselines. Second, an ablation study is performed, in order to test the importance of different aspects of CoNeRa (i.e. collective inference, clustering and disagreement). Third, the performance of CoNeRa is analyzed along the query pool size and the initial labeled set size. Fourth, the contiguity-constraint cluster model is evaluated along the selection of the initial labeled set, the seed selection and the distance measure.

5.4.1. Comparative study

In this set of experiments, the error curve of CoNeRa is compared to that of four baselines, namely, Random, DAL, PAL and ALFNET-R. The error curves for all the datasets are shown in Figures 3(a)-3(k). They demonstrate that CoNeRa has an often better performance than its baselines, although certain particular results can be observed. For example, the error curve of CoNeRa indicates several fluctuations in SCH (see Figure 3(g)). They may originate from the presence of several target outliers in this data network (see box plot of SCH labels in Figure 4(a)). This interpretation is also supported by the verification that CoNeRa has a constantly better performance than its baselines when the labels are distributed without significant outliers (see, for example, the error curves of SOILMoisture in Figure 3(h) with respect to the box plot of SOILMoisture targets in Figure 4(b)). This behavior may indicate that the proposed contiguity constraint clustering procedure is not sufficiently robust for the presence of outliers thus, suggesting further investigations in this direction. Beyond these considerations, the general trend highlighted by these results shows that the collective strategy, described in this study, basically has a positive impact on the error curves. Indeed, CoNeRa commonly gains accuracy with respect to the naive random strategy (Random), as well as the state-of-the-art active regression algorithms that neglect the network structure (DAL and PAL), and the baseline collective active learning strategy that is already defined in the literature (ALFNET-R). The performance of CoNeRa can be ascribed to its peculiar components, namely collective inference, the contiguity constraint clustering solution and the cluster-based correlation-aware disagreement measure. The actual impact of each component on the accuracy of the active learning strategy is investigated in the ablation study (see Section 5.4.2). The conclusions of this comparative analysis can be strengthened by the results of a statistical pairwise comparison of the error curves. The performed test is the two-sided Wilcoxon signed rank test [21]. Given two populations x and y (here the error curve of CoNeRa - x and the error curve of a baseline - y), the test returns the p-value of a paired, two-sided test for the null hypothesis that $x - y$ comes from a distribution with median zero. This is tested against the left-tailed alternative hypothesis stating that the data in $x - y$ come from a distribution with the median less than 0 (i.e. there is enough statistical evidence to assess that the median error applying CoNeRa is less than the median error applying the considered baseline). The $p-$values of the Wilcoxon rank sum tests are collected in Table 4 for each baseline in this study. The null hypothesis is commonly rejected with $p \leq 0.05$. Therefore, this statistical analysis shows that the error curve of CoNeRa is "statistically" better than the curve of Random in 8 out of 11 datasets, the curve of DAL in 9 out of 11 datasets, the curve of PAL in 6 out of 11 datasets

and the curve of ALFNET-R in 10 out of 11 datasets. Based on this analysis, the collective active strategy implemented by CoNeRa commonly results in a statistically significant error improvement (at p-value 0.05) observable in several datasets. This comparative study is completed by analyzing the average learning time (in milliseconds) spent performing the initial clustering step (contiguity constraint clustering - CC for CoNeRa and graph clustering - GC for ALFNET-R), as well as completing the iterative active learning step in CoNeRa, ALFNET-R, Random, DAL and PAL. The analysis of the times, reported in Table 5, deserves several considerations. First, CC is more time-consuming than GC. This is coherent with the fact that CC uses both the network structure and the descriptive data, while GC considers the network structure only. In any case, the ablation study (see Section 5.4.2) will prove that the additional complexity of CC will be counter-balanced by the observed gain in accuracy. Second, the active learning process with collective inference (CoNeRa and ALFNET-R) is more complex than the active learning step of PAL that bases the active example selection on the decision of an ensemble of descriptive regressors learned from various subsets of the training data. However, it is not necessarily more complex than the active learning step of DAL that bases the active example selection on the distance of each unlabeled example from labeled ones. The cost of the distance computation depends on the descriptive vector size m. This explains the verification that CoNeRa and ALFNET-R are less time-consuming than DAL in datasets with high size m (e.g. see the learning times with Irs where m=28, Vegetation and SOIL moisture where m=33, PubMellitus where m=19). Finally, the learning times of CoNeRa and ALFNET-R have roughly the same order of magnitude, although a deeper analysis shows that CoNeRa is slightly less efficient than ALFNET-R. This mainly depends on the active selection criterion adopted by CoNeRa that looks for the disagreement among linked labels. As this criterion explores the network structure, it appears less efficient (even if commonly more accurate) than the criterion adopted by ALFNET-R that looks for the disagreement among labels predicted by the collective, descriptive and graph-based clustering models. This interpretation is supported by the observation that the gap between the computation times of CoNeRa and ALFNET-R augments as the complexity (number of nodes and edges) of the network increases (see the learning times with Irs, SCH and YouTube).

5.4.2. Ablation study

The contribution of each of CoNeRa's components is tested by: studying the collective and descriptive regressors; investigating the constraint and graph clustering; exploring the cluster-based disagreement.

Collective regression model vs Descriptive regression model. An analysis is performed on the actual contribution of the collective knowledge to the accuracy of the regression models used to predict the labels of testing sets. To this aim, the error curve of the regression models learned from the variable space, composed of both descriptive and collective variables (collective regression model - CoNeRa-C), is compared to the curve of the regression models learned from the variable space, composed of descriptive variables only (collective regression model - CoNeRa-D). The error curves are shown in Figures 5(a)-5(k). These results show that CoNeRa-C is stacked stably on top of CoNeRa-D in six out of eleven datasets, namely in Irs, Ms, NCEP,

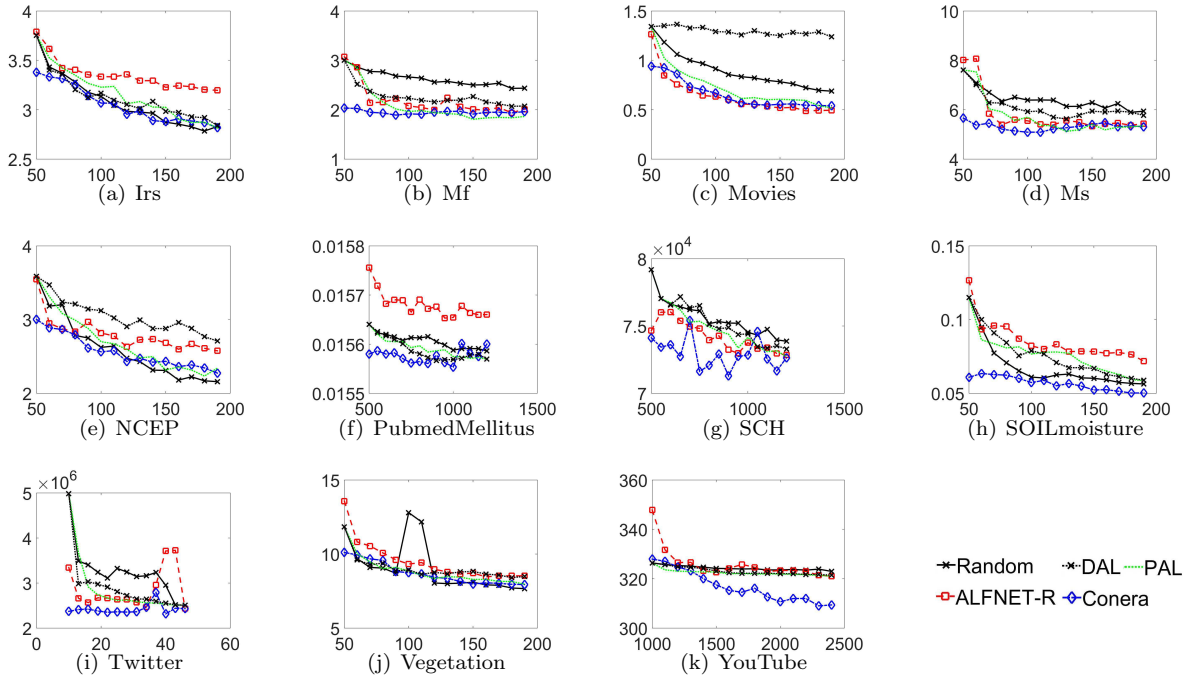


Figure 3: Error curves (RMSE (axis Y) computed on the testing data along the labeled set size at consecutive iterations (axis X)): Random, DAL, PAL, ALFNET-R and CoNeRa

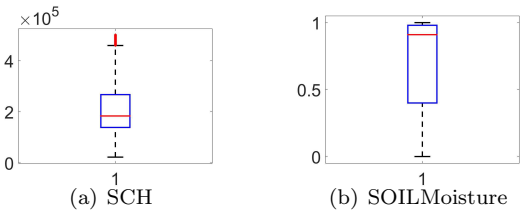


Figure 4: Box plot of the target data: SCH and SOILMoisture.

PubMedMellitus, SCH and YouTube. This also confirms that the injection of collective knowledge into the regression model used, in the end, for the testing procedure, clearly stands out as the winning strategy in the majority of the datasets which have been considered. On the other hand, the error curve of CoNeRa-C is approximately close to the error curve of CoNeRa-D in three out of eleven datasets, namely in Movies, Twitter and Vegetation. This verification can be strengthened by resorting to the pairwise Wilcoxon rank sum test, in order to compare the error curves yielded by both CoNeRa-C and CoNeRa-D. The test is performed with the null hypothesis that the error curves of CoNeRa-C and CoNeRa-D are samples from continuous distributions, which perform equally, against the alternative that they do not work equally. The test does not provide enough evidence to reject the equal performance hypothesis, as it returns $p-$value=1 for Movies and Vegetation and $p-$value=0.8777 for Twitter. A deeper discussion is opened analyzing the error curves

Table 4: **Pairwise Wilcoxon rank sum test**. Evaluation of the hypothesis of an increase in the median of pairwise compared error curves of CoNeRa, Random, DAL, PAL and ALFNET-R. $p - value \leq 0.05$ is in bold. This indicates that there is evidence to reject the null hypothesis 0 (hypothesis of equal performance) and concludes that there is a positive shift in the median of the observed error in the same curve from CoNeRa to Random/DAL/PAL/ALFNET-R at the 0.05 significance level.

Data	CoNeRa vs Random	CoNeRa vs DAL	CoNeRa vs PAL	CoNeRa vs ALFNET-R
Irs	0.53306	0.14042	0.17004	**0.00033545**
Mf	**1.6959e-06**	**1.6959e-06**	0.58215	**4.067e-05**
Movies	**0.0006076**	**1.6959e-06**	0.13138	0.86862
Ms	**1.6959e-06**	**2.0716e-06**	**0.057494**	**0.0006076**
NCEP	0.66085	**4.067e-05**	0.30933	**0.0050609**
PubmedMellitus	**2.01e-05**	0.099253	**0.011266**	**1.6959e-06**
SCH	**2.8685e-05**	**0.00015393**	**0.00070204**	**0.002105**
SOILmoisture	**0.0023974**	**2.01e-05**	**1.6784e-05**	**1.6959e-06**
Twitter	**1.3047e-05**	**3.9285e-05**	**7.386e-05**	**0.00010017**
Vegetation	0.62999	**0.06243**	0.26691	**0.019044**
YouTube	**0.0018453**	**0.0064108**	**0.011266**	**0.0018453**

Table 5: **Learning times**. The average learning time (in millisecs) is computed on the trials performed for each dataset.

Data	clustering		active learning				
	CC	GC	CoNeRa	ALFNET-R	Random	DAL	PAL
Irs	102483	881	13686	9864	4560	21764	3967
Mf	1721	115	3256	2954	2056	2148	2088
Movies	2185	243	4513	3565	2290	2673	2445
Ms	1692	134	3300	2923	2061	2137	2154
NCEP	17265	581	8204	6273	3334	7420	3131
PubmedMellitus	73153	129	10793	10798	5854	17054	9257
SCH	8821386	377400	230924	181114	62734	78400	15347
SOILmoisture	95983	1474	11623	9840	4253	32283	4974
Twitter	171	108	1693	1343	807	1068	1324
Vegetation	93229	1474	14216	10066	4406	32332	4955
YouTube	563020	4688	218580	123490	16428	145276	29288

with Mf dataset. For this dataset, CoNeRa-C is stacked on top of CoNeRa-D when labels have been acquired for less than 120 nodes, while CoNeRa-D is stacked on the bottom of the CoNeRa-D when labels are acquired for more than 120 nodes. This means that a labeled set represented in both the descriptive and collective space easily can counterbalance the learning limits (e.g. overfitting), caused by the scarceness of known labels. Agreeably, the positive impact of the injection of the collective knowledge into the regression models is observed in the initial iterations of the learning process when fewer labels have been acquired. It can also be observed that CoNeRa-C is stacked stably on top of CoNeRa-D in the SOILmoisture dataset only. This is the only dataset in this study, where there is a clear evidence that a benefit can be achieved in the accuracy of the testing procedure, by preferring the descriptive regression model to the collective regression model. Based on this analysis, the considerations already formulated in [7] can be confirmed in general term: the considered experiments provide empirical evidence that the collective regression learner is generally better than (or equally well to) the descriptive regression learner in the testing procedure.

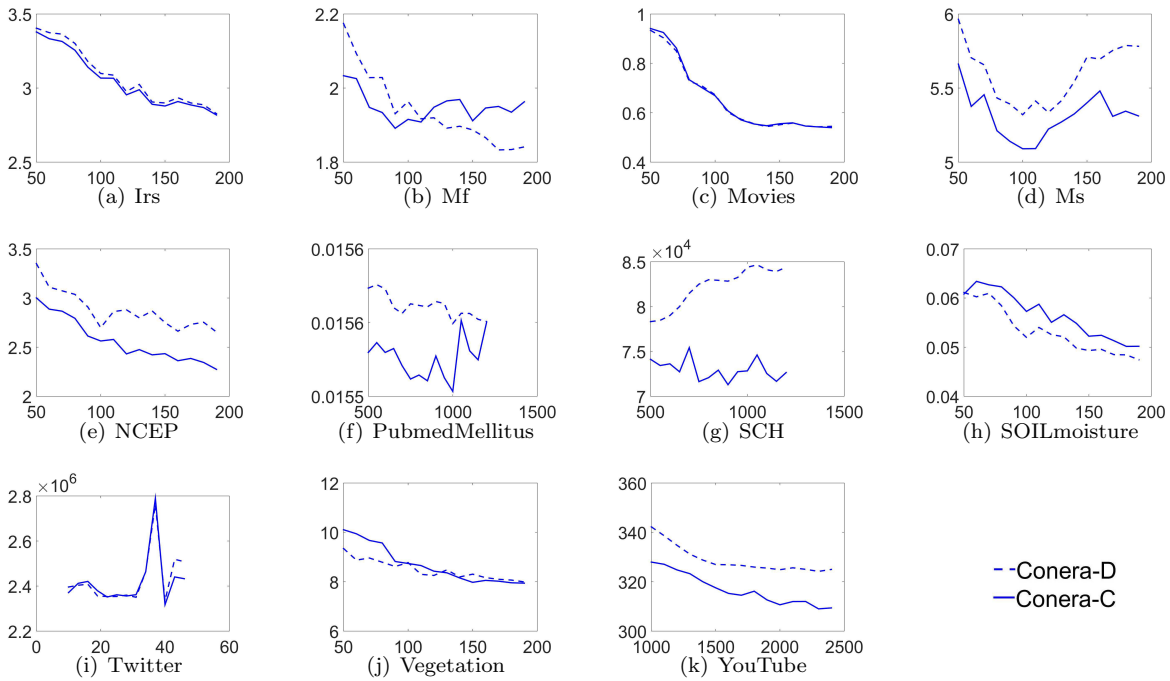


Figure 5: CoNeRa error curves (RMSE (axis Y) computed on the testing data along the labeled set size at consecutive iterations (axis X)): regression model learned in labeled set spanned on the descriptive variable space (CoNeRa-D) vs regression model learned in the labeled set spanned on the descriptive+collective variable space (CoNeRa-C) .

Constraint clustering vs Graph clustering. The actual contribution of the selected constraint clustering solution is analyzed. In CoNeRa, clustering information is used both in the initialization phase and in the iterative phase. In the initialization phase, the initial labeled set is constructed by sampling the examples to be labeled cluster by cluster. In the iterative phase, the correlation-aware disagreement of an example is computed on the cluster-based representation of its neighborhood and weighted according to the size of the labeled part of the cluster it belongs to. This idea of exploiting the knowledge of the cluster structure of the network in the active learning strategy is introduced at first in [7], where clustering is performed by taking advantage of the available network structure and by using a graph clustering algorithm to find the clusters. On the contrary, this study proposes to adopt a constraint clustering algorithm that identifies clusters of connected, similar nodes. In particular, the cluster structure of the data network (see details in Section 4.2) is determined by accounting for both the network information (i.e. the network linkedness) and the node similarity (i.e. the similarity between the nodes computed in the descriptive space). In the described experiments, the error curve of CoNeRa run with the constraint clustering algorithm (CoNeRa-CC), is compared to the error curve of CoNeRa run with the graph clustering algorithm (CoNeRa-GC). The graph clustering algorithm described in [37] is considered. The error curves, shown in Figures 6(a)-6(k), reveal that the constraint clustering algorithm generally stands out as the winning clustering algorithm for our

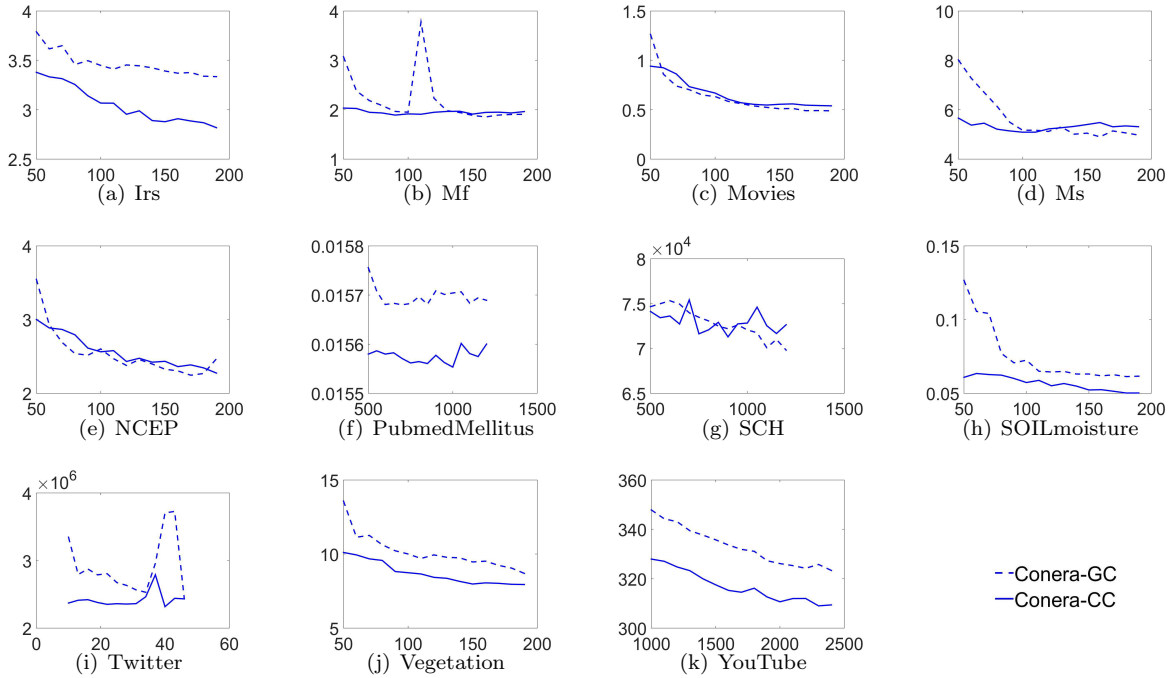


Figure 6: CoNeRa error curves (RMSE (axis Y) computed on the testing data along the labeled set size at consecutive iterations (axis X)): graph clustering (CoNeRa-GC) vs constraint clustering (CoNeRa-CC).

active learning strategy. In all datasets, the initial labeled set, constructed by considering the constraint-cluster structure of the network data, allows us to construct a regression model that is more accurate than the regression model constructed from the labeled set, initialized by considering the graph-cluster structure of the network. On the other hand, in almost all datasets, the consideration of a constraint-cluster structure contributes to a gain in accuracy in the iterative phase, compared to the graph-cluster structure. In particular, graph clustering constantly outperforms constraint clustering in the Movies dataset only.

Cluster-based disagreement. The role of the cluster structure in the disagreement measurement is explored. The complete version of CoNeRa, where the disagreement of a node (see details in Section 4.2) is computed on the cluster-based projection of its neighborhood (CoNeRa-CS), is compared to the variant that computes the disagreement of an example in its entire neighborhood, but without accounting for the clustering information (CoNeRa-NCS). The error curves, shown in Figures 7(a)-7(k), highlight the fact that using clustering information provides, in general, gains over using just disagreement information. The evidence of this gain is clear in various datasets, i.e. Irs, Movies, Ms, NCEP, SCH, SOILmoisture, Vegetation and YouTube.

5.4.3. Sensitivity study

The sensitivity of the active learning strategy is investigated along size l of the initial labeled set and size t of the sample pool. For this analysis, SCH, SOILmoisture and YouTube are considered. The error curves,

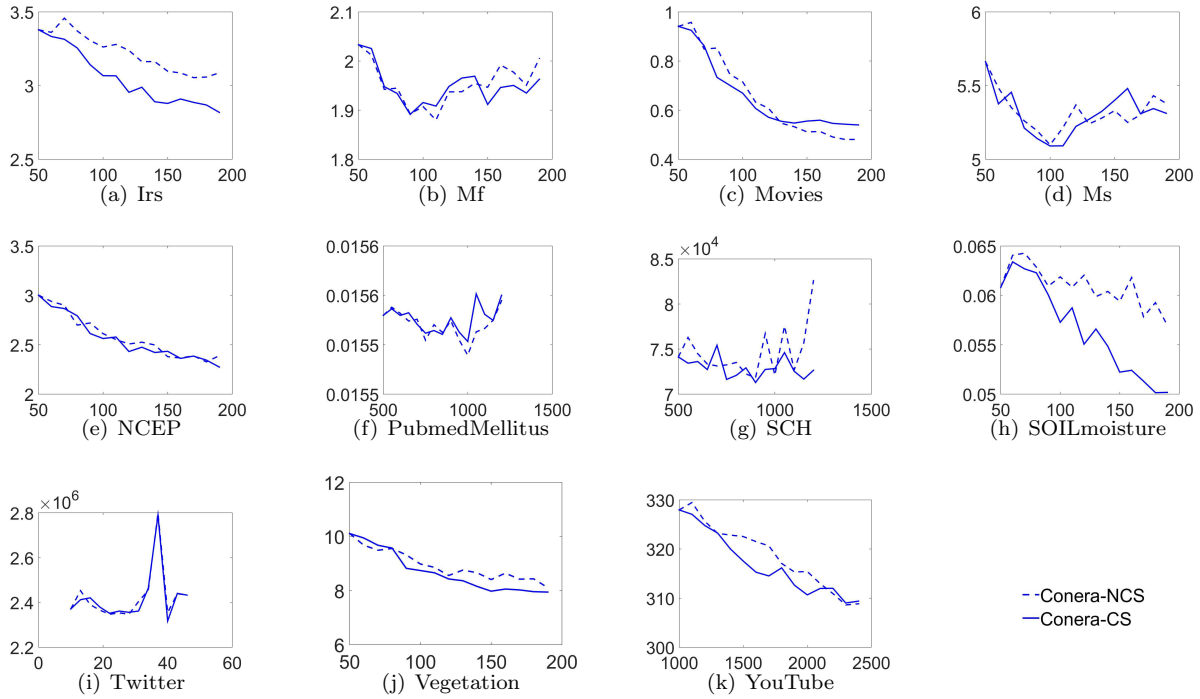


Figure 7: CoNeRa error curves (RMSE (axis Y) computed on the testing data along the labeled set size at consecutive iterations (axis X)): sample pool selection with no-cluster based disagreement measure (CoNeRa-NCS) vs sample pool selection with cluster-based disagreement measure (CoNeRa-CS) .

by varying l, are shown in Figures 8(a)-8(c), while the error curves, by varying t, are shown in Figures 8(d)-8(f). In addition, the impact of the choice of l and t on the average learning times (in milliseconds) spent completing the active learning process is evaluated.The learning times are collected in Table 6. These results show that the lower l and/or t, the higher the number of performed iterative steps and, hence, the higher the learning time spent using the entire budget and then completing of the active learning process. On the other hand, the shape of the error curve is clearly influenced by the choice of l and t. In any case, all the regression models, which are learned when the entire budget B is used, converge generally to an approximately fixed point of testing accuracy. The only exceptions are observed in the SCH dataset, where the accuracy of the final regression model is visibly greater when either the size of the initial labeled set (l=700 in Figure 8(a)) is enlarged or the size of the sample pool (t = 25 in Figure 8(d)) is diminished. This is attributed to the presence of various target outliers in SCH (see Figure 4(a)). The presence of outliers may add complexity to the task. Having outliers in a labeled set, especially in the initial iterations (i.e. when the regressors are learned with few labeled data), may lead to computing models that overfit outlier characteristics. On the other hand, outliers diminish the effectiveness of the correlation analysis during the collective inference. Therefore, starting with a larger labeled set may be a valid means to balance the presence of outliers and contribute to learning accurate initial models that will positively influence the subsequent learning process.

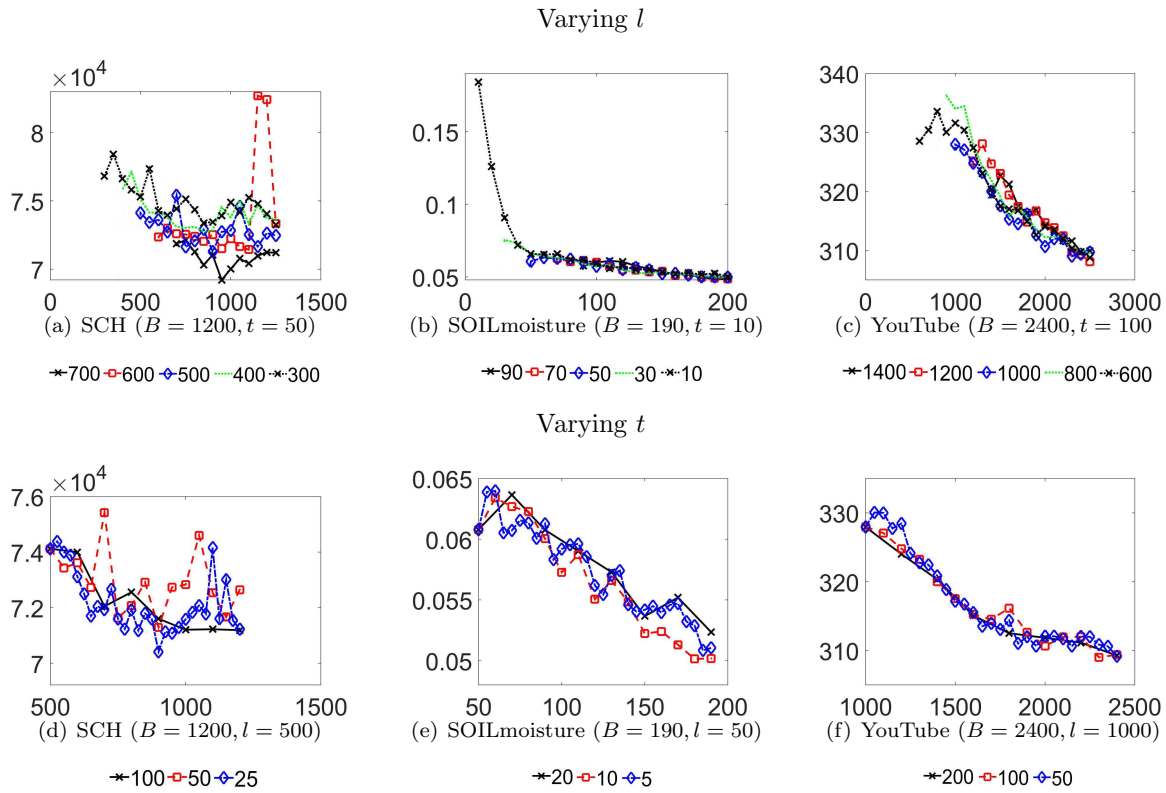


Figure 8: CoNeRa error curves (RMSE (axis Y) computed on the testing data along the labeled set size at consecutive iterations (axis X)) by varying l or t.

At the same time, reducing the sample pool size avoids the addition of too many outliers, especially during the initial iterations, when their contribution will have a stronger effect on the training phase.

Table 6: **Learning times**. Results (in millisecs) are collected by varying l or t.

	l					
SCH ($B = 1200, t = 50$)	l	700	600	500	400	300
	time	187782	213725	230924	259178	289581
SOILmoisture ($B = 190, t = 10$)	l	90	70	50	30	10
	time	9858	10555	11623	12106	13055
youTubeScribers ($B = 2400, t = 100$)	l	1400	1200	1000	800	600
	time	25676	26307	27239	26883	27223
	t					
SCH ($B = 1200, l = 500$)	t	100	50	25		
	time	145877	230924	411660		
SOILmoisture ($B = 190, l = 50$)	t	20	10	5		
	time	7783	11623	19165		
youTubeScribers ($B = 2400, l = 1000$)	t	200	100	50		
	time	218580	27239	42320		

5.4.4. Cluster structure study

This investigation evaluates the clustering impact on the initialization sampler, as well as the clustering stability along the seed selection order (i.e. order according to seeds are evaluated during the cluster construction) and the distance computed (i.e distance computed to evaluate the cluster homogeneity).

Clustering along sampling. The initial labeled set corresponds to the set of l training nodes that are initially sampled from $\mathcal{U}$ and labeled by the oracle before starting the iterative active and collective learning process. For each training dataset, the initial labeled set is fed with nodes that are randomly selected from the training dataset by resorting to the random sampling procedure without replacement. In Random, DAL and PAL, sampling is fully random across the entire training dataset (i.e. each node is randomly extracted from the entire training dataset without replacement), while in ALFNET-R and CoNeRa a random sample is selected from the cluster structure of the training dataset (i.e. nodes are randomly sampled per cluster without replacement, the number of nodes sampled per cluster is proportional to the cluster size; details are in Section 4.2). In all cases, the use of a random sampler implicates that, by repeating the random sampling on several trials, different nodes may be selected to feed the initial labeled set. In the entire empirical study, to account for the impact of the random sampling procedure on the initialization of the labeled set, five trials of labeled sets are always randomly generated for each training set of the 5-fold CV (see details in Section 5.2). However, a feedback on the impact of the random sampling procedure can be achieved by a deeper analysis on the descriptive regression model that is constructed, during the initialization phase, after the selection of the training nodes done by the sampler and labeled by the oracle. With this intention, a comparison is carried out between the average and standard deviation of the (testing) RMSEs of the initial descriptive regression model constructed with the fully random sampler, the graph cluster-based random sampler and the contiguity constraint cluster-based random sampler. Results, collected in Table 7, confirm that the contiguity constraint cluster structure of the training data set provides useful information to drive the selection of a "good" initial sample of nodes. The contiguity constraint cluster-based sampling (being repeated on various trials) guarantees the learning of the most accurate initial descriptive regression model in general. This also confirms the best initialization of the labels for the collective inference. Finally, the standard deviation analysis clarifies that the contiguity constraint cluster-based sampler is commonly more robust than its competitors (random sampler and graph cluster-based sampler) when evaluated along the random aspect of the initialization phase. The initial descriptive regression models, computed on the various repetitions of the experiment, commonly achieve the lowest variability (standard deviation) of the RMSE.

Clustering along seed selection. The contiguity constraint clustering algorithm starts the cluster construction from a seed node that is randomly selected across the set of nodes, still un-clustered, in the training dataset. The execution of the clustering algorithm is repeated on five trials (by selecting randomly, at each trial, cluster seeds), in order to evaluate the robustness of the cluster structure to the seed selection. The derived cluster structures are compared to the cluster structure (baseline) used in the remaining part of this empirical study. For this purpose, the Rand index [42] is computed as a measure of agreement between two cluster

Table 7: **Sampling procedure accuracy analysis**. For each data set, the training set can be initialized by resorting to either a fully random sampler, a graph cluster-based sampler or a constraint cluster-based sampler. The accuracy of the descriptive regression model, learned during the initialization phase, is measured. The average ± standard deviation of the testing RMSE is collected (columns 2-4). The lower average and standard deviation are signed in bold.

Data	Fully random	Graph cluster-based random	Constraint cluster-based random
Irs	3.7530 ± 0.2772	3.8825 ± 0.3760	**3.4043± 0.2346**
Mf	3.0016 ± 1.4126	2.9104 ± 1.4903	**2.1745 ± 0.9204**
Movies	1.3419 ± 0.2268	1.2076 ± 0.3149	**0.9339 ±0.2196**
Ms	7.6072 ± 3.3628	7.4161 ± 3.2780	**5.9646 ± 1.5979**
NCEP	3.5880 ± **0.3856**	3.6925 ± 0.6868	**3.3510** ± 0.4414
PubmedMellitus	0.0156 ± 0.0012	0.0157 ± 0.0012	**0.0156 ± 0.0012**
SCH	79190.3488 ± 4164.3062	**78028.2597** ± 2501.9610	78322.7468 ± **2269.3593**
SOILmoisture	0.1150 ± 0.0364	0.1126 ± 0.0686	**0.0611 ± 0.0101**
Twitter	4981991.8417 ± 7037146.7156	3334259.3878 ± 2682523.1932	**2395733.9126 ± 2332402.5417**
Vegetation	11.8321 ± 6.2437	12.8515 ± 3.6365	**9.347**4 ± **1.5830**
YouTube	**326.3828 ± 18.7350**	340.8697 ± 22.4620	342.2367 ± 25.0099

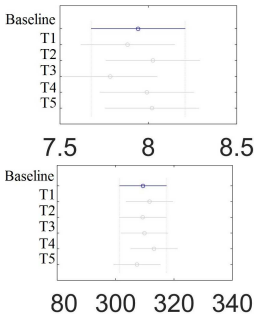
structures of the same dataset. It is measured by considering all pairs of nodes and counting pairs that are assigned to the same or different clusters in the trial and baseline cluster structure. It assumes values between 0 and 1. It has a value close to 0.0 for random labeling independently of the number of clusters and samples and exactly 1.0 value when the cluster structures are identical (up to a permutation). Using the Rand Index to compare the cluster structures (see column 2, Table 8), it can be observed that a moderate to high degree of similarity appears between cluster structure outcomes. The Rand index is commonly greater than 0.80, except for Irs (0.6601) and SCH (0.69143), where it is still greater than .65. This analysis shows that the clustering algorithm can be considered, generally, robust to the seed selection order.

To support this conclusion, CoNeRa is run along the various cluster structures generated when changing the seed order selection. For each cluster configuration, the regression accuracy (RMSE) of CoNeRa is measured over the several trials of the considered dataset, as they are defined by the methodology presented in Section 5.2. The testing accuracy of the collective regression model, learned when the iterative process stops (and therefore labels of the full budget are acquired) is analyzed. The one-way analysis of variance (ANOVA) [8] is performed, in order to determine whether there are any significant differences between the means of the RMSEs collected along the compared cluster configurations on the tested trials. The tested hypothesis shows that they are all the same against the general alternative that they are not. By considering that the common significance level of this statistical analysis is 0.01, the ANOVA analysis (see column 3, Table 8) shows that there is no statistical significant difference due to the impact of the cluster structures generated with different seeds on the final accuracy of the entire learning process. To display the results of the ANOVA analysis, the interactive graph of the estimates with the comparison intervals of the multiple comparison test are derived for datasets Vegetation and YouTube (see columns 4-5, Table 8) .[11] In the visualization, each group mean is represented by a symbol, and the interval is represented by a line

[11] Similar visual results are achieved for the remaining datasets. They are omitted due to space limitation.

Table 8: **Seed selection ANOVA analysis**. For each training set, the cluster structure construction is repeated on five new trials by changing randomly the seed selection order during the cluster construction. The average of the Rand Index is computed between the cluster structure constructed in each trial of this specific experiment and the baseline cluster structure used in the remaining part of this empirical study (column 2). Th p-value of the ANOVA analysis compares RMSEs of the collective regression models learned for each cluster configuration generated by varying the seed selection order (column 3). The interactive graph represents the estimates and comparison intervals of the multiple RMSE comparison test associated with the ANOVA analysis (using Matlab function multcompare()) visualized for Vegetation and YouTube (columns 4-5). For each interactive graph, axis Y represents: Baseline (i.e. the cluster configuration considered as the baseline in the experimental study), as well as T1, T2, T3, T4 and T5 (i.e. the cluster configurations generated by running clustering on five trials). Axis X represents the comparison intervals around the configurations on axis Y.

Data	RandIndex	ANOVA p-value	multicompare
Irs	0.66018	0.0536	
Mf	0.82673	0.9977	
Movies	0.84011	0.1853	
Ms	0,81672	0.9918	Vegetation
NCEP	0.88099	0.0985	
PubmedMellitus	0.99927	0.9997	
SCH	0.69143	0.6289	
SOILmoisture	0.85561	0.6876	
Twitter	0.99294	1.0000	YouTube
Vegetation	0.83045	0.7706	
YouTube	0.93313	0.9330	



extending out from the symbol. Two group means are significantly different if their intervals are disjoint; they are not significantly different if their intervals overlap. This analysis confirms the robustness of the entire learning process to the seed selection order considered during the cluster structure construction showing that collective regression accuracy does not change greatly, despite slight differences possibly observed either in the cluster boundaries or in the labeled nodes initially sampled across these boundaries.

Clustering along distance. The contiguity constraint clustering algorithm bases the computation of the homogeneity property (see Formula 2) on the Euclidean distance between the vectors of the descriptive values associated to two (candidate) clustered nodes. To evaluate the quality of the cluster structure along the distance, the cluster configuration computed by employing the Euclidean distance can be compared to the configuration computed employing alternative distances, e.g. Chebyshev distance or Mahalanobis distance. The Chebyshev distance is a metric defined on a vector space, where the distance between two vectors is the greatest of their differences along any coordinate dimension. The Mahalanobis distance is a measure of the distance between a node n and a distribution $\mathcal{N}$. It is a multi-dimensional generalization of the idea of measuring how many standard deviations are away from the average of $\mathcal{N}$. This distance is zero if n is at the average of $\mathcal{N}$, and grows as n moves away from the average: along each principal component axis, it measures the number of standard deviations from n to the average of $\mathcal{N}$. The mathematical formulation and the time complexity of the compared distances are reported in Table 9. To evaluate the stability of the cluster configuration along the distance, the Rand index is computed between the cluster structure determined

Table 9: **Distance complexity analysis**. Let $n(x_1, x_2, \ldots, x_m)$ be a vector of descriptive values associated with a node $n \in \mathcal{N}$, $n_i, n_j \in \mathcal{N}$ be two nodes, $\mathbf{C}$ is the covariance matrix of the node set. The formulation of the Euclidean distance, Chebyshev distance and Mahalanobis distance (column 2) and their complexity in the O notation (column 3).

Distance	Formula	O complexity
Euclidean	$d(n_i, n_j) = \sqrt{\frac{\sum_{k=1,\ldots,m} (n_i(x_k) - n_j(x_k))^2}{m}}$	$O(m)$
Chebyshev	$d(n_i, n_j) = \max_{k=1,\ldots,m} \lvert n_i(x_k) - n_j(x_k) \rvert$	$O(m)$
Mahalanobis	$d(n_i, n_j) = \sqrt{(n_i(x_1, \ldots, x_m) - n_j(x_1, \ldots, x_m))^T \mathbf{C}^{-1} (n_i(x_1, \ldots, x_m) - n_j(x_1, \ldots, x_m))}$	$O(N^2 m)$

Table 10: **Rand index analysis**. For each dataset, the Rand Index is computed between the cluster structure constructed by considering the baseline Euclidean distance - E and the cluster structure constructed by considering either the Chebyshev distance - C or the Mahalanobis distance - M.

Dataset	E vs C	E vs M	Dataset	E vs C	E vs M	Dataset	E vs C	E vs M
Irs	0.67729	0.67057	Mf	0.84045	0.85075	Movies	0.85786	0.85377
Ms	0.84495	0.84864	NCEP	0.82490	0.82002	PubmedMellitus	0.99972	0.99959
SCH	0.51647	0.46140	SOILmoisture	0.80896	0.80723	Twitter	0.99556	0.99556
Vegetation	0.81397	0.81210	YouTube	0.89898	0.73477			

with the Euclidean distance (baseline) and the cluster structure determined with the Chebyshev distance or the Mahalanobis distance. The results, reported in Table 10, generally show high the degree of similarity between cluster structure outcomes, although computing the Mahalanobis distance is asymptotically more complex than computing the Euclidean distance (see column 3, Table 9).

Again the actual impact of the differences possibly observed in the cluster boundaries is measured in terms of the regression accuracy of the collective regression model learned by CoNeRa when the iterative process stops. The average and standard deviation of the (testing) RMSEs are reported in Table 11. These results show that the Euclidean configuration outperforms competitors in five-out-of-eleven datasets, the Chebyshev configuration outperforms competitors in two-out-of-eleven datasets, while the Mahalanobis configurations outperforms competitors in four-out-of-eleven datasets. The one-way analysis of variance (ANOVA) is, finally, performed, in order to determine whether there are any significant differences between the averages of the RMSE along the compared distances on the tested trials. The significance level of this statistical analysis is 0.01 (see column 4, Table 11). The ANOVA analysis reveals that there are only four-out-of-eleven datasets, where the test hypothesis is rejected as the errors of the compared configurations are statistically different. In these datasets, a multiple comparison test is performed, in order to determine whether any of those averages are significantly different from each other [8]. The results, plotted in Figures 9(a)-9(d), highlight that the Euclidean configuration is statistically better than the Chebyshev and Mahalanobis configurations in both Movies and SOILmoisture, while it is statistically better than the Mahalanobis configuration (and equally to the Chebyshev configuration) in Vegetation. The only case where the Chebyshev configuration is statistically better than the Euclidean one is in NCEP. This empirical analysis combined with the considerations on the complexity of the distance computation confirms the efficacy of the idea of performing the constraint clustering algorithm in combination with the Euclidean distance.

Table 11: **Distance ANOVA analysis**. For each data set, the testing RMSE (average ± standard deviation) of CoNeRa (final collective regression model) is collected. The constraint cluster structures are computed with the Euclidean distance, the Chebyshev distance or the Mahalanobis distance (columns 2-4). The lowest error is in bold. The p-value of the ANOVA analysis (column 4). The p-value less than .01 (in bold) means that there is one configuration that statistically outperforms the compared configurations.

Dataset	Euclidean	Chebyshev	Mahalanobis	ANOVA p-value
Irs	**2.81793** ± 0.17195	2.87298 ± 0.17193	2.85332 ± 0.16570	0.52664
Mf	1.96288 ± 1.01006	1.95065 ± 0.97069	**1.91082**± 1.01582	0.98234
Movies	**0.53963** ± 0.17491	0.72619 ± 0.21951	0.74987 ± 0.23927	**0.00164**
Ms	5.31166 ± 1.42782	4.98325 ± 1.52704	**4.98081** ± 1.19437	0.6395
NCEP	2.27356 ± 0.16962	**2.04945** ± 0.20279	2.15830 ± 0.15588	**0.00020**
PubmedMellitus	0.01560 ± 0.00115	**0.01553** ± 0.00110	0.01553 ± 0.00115	0.9705
SCH	72640.976± 4531.820	70501.938 ± 2423.068	**69538.287** ± 2842.734	0.00734
SOILmoisture	**0.05017** ± 0.00346	0.06351 ± 0.00459	0.07721 ± 0.00798	**3.598e-25**
Twitter	2432550.609 ± 2058910.482	2432550.609 ± 2058910.482	**2432550.609** ± 2058910.482	1.000
Vegetation	**7.94238** ± 0.61360	8.30071 ± 0.53465	15.28869 ± 2.41583	**1.1.882e-29**
YoutTube	**309.38877** ± 18.18947	316.73096 ± 20.73862	317.29956 ± 17.97720	0.28075

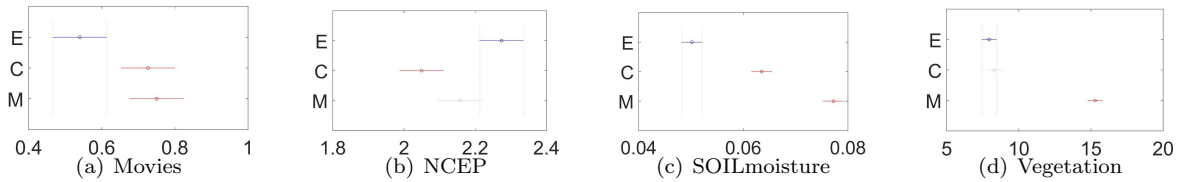


Figure 9: ANOVA analysis visualization (p-value=0.01): Euclidean - E, Chebyshev - C, Mahalanobis - M . The Matlab function multcompare() is used, in order to visualize the multiple comparison test. Axis Y represents the distance (E, C, M), while axis X represents the comparison intervals around the configurations on axis Y.

5.5. Temporal analysis

Although CoNeRa is designed to process network data collected at a specific point of time, this experiment is performed, in order to preliminary investigate the impact of the temporal dimension on the viability of the collective model learned by CoNeRa. The considered dataset contains the atmospheric forcing data collected by the Geophysical Fuild Dynamics Laboratory for NOAA [14]. Data were recorded monthly from November, 1997 to November, 2000 by 162 sensors, which roughly cover the area of the U.S. state of Alaska. The target variable is the cloud area fraction, while the descriptive variables are: the atmosphere cloud ice content, the snowfall flux, the surface air pressure and the surface downward eastward stress. The sensors represent the nodes and the variables associated with the nodes take the original values of the geophysical measurements recorded during the period of observation. As for spatial networks described in Section 5.1.2, the coordinates of the sensors are used to define the edge spatial structure of the network. In particular, every node is actually linked to the nodes falling in the influence sphere with radius equal to the 10 percent of the maximum Euclidean distance between the sensors in the network. The Global Moran's I and its expected value $E(I)$, calculated on the target data, every month between November, 1997 and November

2000, are reported in Figure 10(b). Values $I \geq E(I)$ confirm that a positive in-network phenomenon of correlation of the target can be observed at each specific time point. This means that close target values are generally similar when they are collected at the same time point, although their visualization, reported in Figure 10(a), highlights that the target variable changes over time. In any case, the observed temporal data change exhibits a seasonal pattern as the fluctuations roughly repeat on a yearlong period. This suggests that a regression model, learned at a time point, can be efficaciously employed to forecast upcoming periodic testing data coherently with the observed seasonal pattern. The validity of this assumption is here verified by investigating the forecasting performance of collective regression models.

The collective regression models are those learned by CoNeRa when the collective active learning process stops. The algorithm is run with $l = 10$, $t = 3$ and $B = 34$ by resorting the experimental methodology already described in Section 5.2. Specifically, the 5-fold CV of the network is performed (nodes are randomly divided in five folds; for each trial, one fold defines the testing sets, the hold-out folds define the training sets). The learning process is repeated from scratch as new data are collected. As reported in Section 5.2, for each training set, the learning process is repeated five times (by changing the initialization of $\mathcal{L}$). Final collective regression models learned at time t are employed to predict the held-out testing targets at the same time point t, as well as to forecast testing targets at 24-ahead time points $t+1, t+2, \ldots, t+24$. The forecasting error curve is constructed. Each point of the curve represents the average RMSE computed on the testing sets, for each time point $t, t+1, t+2, \ldots, t+24$, by using the collective regression models learned at time t. This experiment was repeated with t ranging between November, 1997 and October, 1998.

The average forecasting error curve, computed by averaging the forecasting error curves constructed with t ranging between November, 1997 and October, 1998, is reported in Figure 10(c). Interestingly the overall error exhibits the periodic pattern initially observed in the target data. In addition, the forecasting efficacy of the collective regression model decreases over time. However, the performance can be improved accounting for a known periodic pattern and employing a model to predict upcoming testing data only when they behave similarly to the training data. This consideration is confirmed by exploring the individual forecasting error curves constructed with $t = November, 1997$ (Figure 10(d)), $t = March, 1998$ (Figure 10(e)) and $t = July, 1998$ (Figure 10(f)), respectively. In this detailed analysis, a baseline is considered. It is the error curve, where each point represents the average performance of the regression models trained at the specific time point employed to predict only held-out testing data collected at the same time point. This comparative analysis shows that the accuracy performance is improved when the collective active learning process is repeated from scratch as new data are collected online. However, this analysis also highlights that old models can sometimes perform appropriately in the future when training and testing data behave similarly (i.e. forecasting error curve and baseline error curve are repeatedly closer every twelve time points). Although this preliminary empirical investigation cannot be considered conclusive, it paves the way for future developments in the area of spatio-temporal data mining. In particular, it requires new investigations, in order to extend the presented collective active learning strategy by integrating change detection mechanisms, incremental learning solutions, as well as regression models accounting for trend and seasonality of data.

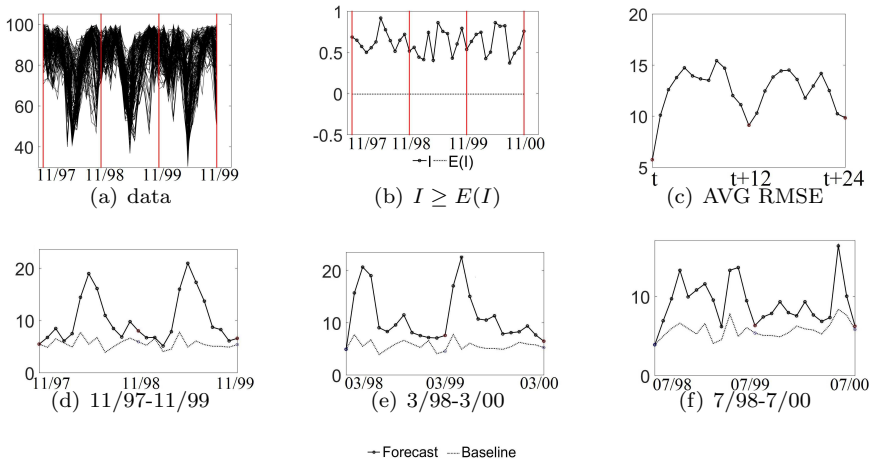


Figure 10: Figure 10(a): The time series of target variable "cloud area fraction" (axis Y) monthly measured by each sensor in the network from November, 1997 to November 2000 (axis X). Figure 10(b): The Global Moran's I and its expected value $E(I)$ (axis Y) of "cloud area fraction" computed by varying time (axis X) monthly between November, 1997 and November 2000. Figure 10(c): The average forecasting error curve (axis Y) computed by averaging the error curves of the regression models trained at time t and used to predict testing data at $t, t+1, \ldots, t+24$ (axis X). The average is computed on the error curves generated with t ranging monthly between November, 1997 and October, 1998. Figures 10(d)-10(f): The forecasting error curve (continuous line - Forecast) compared to the baseline error curve (dotted line - Baseline). Both curves are generated starting from $t = November, 1997$ (Figure 10(d)), $t = March, 1998$ (Figure 10(e) and $t = July, 1998$ (Figure 10(f)), respectively.

5.6. Active Network Regression Literature

Active learning targeting on network regression problems has received significantly less attention than classification. In any case, some recent results achieved with active regression in the network scenario can be considered. Cai et al. [10] evaluate EMCM on forest fires dataset (`http://archive.ics.uci.edu/ml/`). This is a spatial network regression problem as data are collected with spatial coordinates. The target (burned area of forest fires) ranges between 0 and 1090.84 with a few outliers. Removing three top ranked values, the target ranges between 0 and 212.88. The target Global Moran's I and its expected value $E(I)$, calculated without outliers, are 0.0051 and -0.0019, respectively. As the target exhibits positive correlation, the base condition to take advantage of collective inference can be considered satisfied. The experimental setting described in [10] is here repeated. The dataset is split into a training (80%) and a testing set (20%), l is set equal to 10% of the dataset size, t is set equal to 3% of the dataset size and B is set equal to 40% of the dataset size. The split between the training and the testing set is repeated on 10 trials. The RMSE is computed on each testing set with the regressors learned, as B is fully used. The RMSE is averaged on the 10 trials. DAL, PAL, ALFNET-R and CoNeRa are run in this setting. Average errors are: 38.75 (CoNeRa), 39.59 (DAL), 40.50 (PAL) and 40.22 (ALFNET-R), while the errors of EMCM with linear regression (see Fig. 2.c of [10]) are greater than 50.0 and the errors of EMCM with non-linear gradient boosting decision trees (GBDT) for regression (see Fig. 4.c of [10]) are greater than 40.0. Therefore, CoNeRa

yields more accurate predictions than all its competitors, including than EMCM-GBDT that is configured to be robust to the presence of outliers. Komurlu and Bilgic [25] evaluate DGBm on temperature readings in the Intel Berkeley Research Lab sensor network (http://db.csail.mit.edu/labdata/labdata.html). The experimental setting described in [25] is considered. The initial labeled sample collects the readings of 50 sensors, aggregated every 30 minutes, measured on days 2, 3, and 4. A spatio-temporal network is defined by linking readings to readings that are close over space or (past) time. The consequent Global Moran's I is 0.9080 and its expected value $E(I)$ is -0.0019. The unlabeled set collects aggregated data over the first 6 hours of day 5. The active learning strategy is performed, to acquire temperature labels of 10% , 20%, 30%, 40% and 50% of the unlabeled set. The Mean Absolute Error (MAE) is computed on the entire unlabeled set. Collected errors are: 0.32 (CoNeRa), 0.53 (DAL), 0.54 (PAL), 0.34 (ALFNET-R) and 0.31 (DGBm), with 10% of testing labels acquired; 0.26 (CoNeRa), 0.51 (DAL) ,0.48 (PAL), 0.32 (ALFNET-R) and 0.27 (DGBm), with 20% of testing labels acquired; 0.21 (CoNeRa), 0.46 (DAL) ,0.43 (PAL), 0.26 (ALFNET-R) and 0.23 (DGBm), with 30% of testing labels acquired; 0.17 (CoNeRa), 0.41 (DAL) ,0.36 (PAL), 0.21 (ALFNET-R) and 0.18 (DGBm), with 40% of testing labels acquired; 0.12 (CoNeRa), 0.35 (DAL) ,0.21 (PAL), 0.17 (ALFNET-R) and 0.12 (DGBm), with 50% of testing labels acquired. DGBm errors are reported in Table 4 of [25]. This study confirms that CoNeRa performs better than its baseline competitors and its accuracy is comparable to that of DGBm that also exploits correlations. However, DGBm is formulated for univariate spatio-temporal network data, while CoNeRa is defined for multivariate general-network data, although, here, it is forced to consider univariate data only.

6. Conclusion

A novel collective active learning regression algorithm is described. Collective inference is used to handle network information and achieve better predictive accuracy, by labeling examples collectively, rather than treating them as independent examples. Active learning is used to interact with oracles, guiding them to the most informative examples to be labeled, in order to efficiently learn the correct regression model. Collective inference and active learning have been explored in the literature. The novel contribution of this study is that it combines these two strategies in a single learning algorithm, appositely defined for network regression problems. This algorithm, which represents one of the main contributions of this work, proves effective for the network regression problem. A peculiar contribution of the algorithm is the definition of a clustering procedure that accounts for descriptive and network information, in order to derive a model of the data correlation across the network. This cluster knowledge optimizes the selection of the active examples to be labeled by the oracle during the collective active learning strategy. In addition, this study formulates a sample selection criterion that accounts for the network correlation of node labels, in order to select new training examples for the labeling process, as well as the network correlation of node descriptive data, in order to guarantee diversity in label acquisition. The correlation of node labels is expressed through a disagreement score, measuring local correlation of labels. The correlation of node descriptive data is

modeled through a cluster structure, depicting linked nodes of similar descriptive data. The effectiveness of the proposed algorithm is assessed via an empirical study on various data networks characterized by positive correlation of the target.. This study confirms that collective inference can deal with the correlation property and gain in accuracy modeling network data. It contributes to proving that a collective active learning regression algorithm is more accurate than the traditional active learning algorithms, that disregard the network structure of data. The presented algorithm is more accurate than the state-of-the-art collective active learning algorithm that uses collective inference, but adopts a sample selection criterion that neglects the property of target data correlation. In addition, this study proves that collective inference, clustering and network-aware disagreement contribute to improving the accuracy of the algorithm. It also evaluates the accuracy and scalability of the presented collective active learning process along the labeling budget setting, as well as it evaluates the robustness of the constraint cluster structure along the distance measure definition and the seed selection ordering. A further contribution is the investigation of the accuracy of the collective active regression strategy over time. This temporal analysis paves the way for new investigations, in order to integrate change detection mechanisms, incremental learning solutions, as well as regression models accounting for trend and seasonality of data. A final contribution is the comparison with the results of network regression reported in the recent active learning literature. Additional directions for further work are still to be explored. The Markov random field theory can be investigated in the synthesis of collective information. It would be interesting to study ways to determine both the size and shape of neighborhoods. Finally, big data technologies may be studied to apply the presented solution to big imagery data.

7. Acknowledgments

This work fulfills the research objectives of the ATENEO 2014 project "Mining of network data" funded by the University of Bari Aldo Moro. The authors wish to thank Lynn Rudd and Antonietta Bagnardi for their help in reading the manuscript, as well as Caner Komurlu and Mustafa Bilgic for providing the aggregated data of the Intel Berkeley Research Lab sensor network.

[1] S. Abolfathi, A. Yeganeh-Bakhtiary, S. Hamze-Ziabari, S. Borzooei, Wave runup prediction using m5? model tree algorithm, Ocean Engineering 112 (2016) 76 – 81.

[2] M.H. Ahmadi, S.S.G. Aghaj, A. Nazeri, Prediction of power in solar stirling heat engine by using neural network based on hybrid genetic algorithm and particle swarm optimization, Neural Computing and Applications 22 (2013) 1141–1150.

[3] Ahmadi, Mohammad H., Ahmadi, Mohammad Ali, Ashouri, Milad, Razie Astaraei, F., Ghasempour, R., Aloui, Fethi, Prediction of performance of stirling engine using least squares support machine technique, Mechanics and Industry 17 (2016) 506.

[4] A. Appice, A. Ciampi, D. Malerba, Summarizing numeric spatial data streams by trend cluster discovery, Data Mining and Knowledge Discovery 29 (2013) 84–136.

[5] A. Appice, P. Guccione, D. Malerba, A. Ciampi, Dealing with temporal and spatial correlations to classify outliers in geophysical data streams, Information Sciences 285 (2014) 162–180.

[6] A. Appice, D. Malerba, Leveraging the power of local spatial autocorrelation in geophysical interpolative clustering, Data Mining and Knowledge Discovery 28 (2014) 1266–1313.

[7] M. Bilgic, L. Mihalkova, L. Getoor, Active learning for networked data, in: 27th International Conference on Machine Learning, Omnipress, 2010, pp. 79–86.

[8] M. Borgo, A. Soranzo, M. Grassi, MATLAB for Psychologists, Springer, 2012.

[9] R. Burbidge, J.J. Rowland, R.D. King, Active learning for regression based on query by committee, in: 8th International Conference on Intelligent Data Engineering and Automated Learning, Springer, 2007, pp. 209–218.

[10] W. Cai, M. Zhang, Y. Zhang, Batch mode active learning for regression with expected model change, IEEE Trans. Neural Netw. Learning Syst. 28 (2017) 1668–1681.

[11] V. Ceperic, G.G.E. Gielen, A. Baric, Sparse multikernel support vector regression machines trained by active learning, Expert Systems Applications 39 (2012) 11029–11035.

[12] Y. Chen, Spatial autocorrelation approaches to testing residuals from least squares regression, PLOS ONE 11 (2016) 1–19.

[13] S. Dasgupta, D. Hsu, Hierarchical sampling for active learning, in: 25th International Conference on Machine Learning, ACM, 2008, pp. 208–215.

[14] T. Delworth, Preface, Journal of Climate 19 (2006) 641–641.

[15] B. Demir, L. Bruzzone, A multiple criteria active learning method for support vector regression, Pattern Recognition 47 (2014) 2558–2567.

[16] F. Douaka, F. Melgania, N. Alajlanc, E. Pasollia, Y. Bazic, N. Benoudjitb, Active learning for spectroscopic data regression, Journal of Chemometrics 26 (2012) 374–383.

[17] R.A. Dubin, Spatial autocorrelation: A primer, Journal of Housing Economics 7 (1998) 304–327.

[18] B. Epperson, Spatial and space-time correlations in ecological models, Ecological modeling 132 (2000) 63–76.

[19] Y. Freund, H.S. Seung, E. Shamir, N. Tishby, Selective sampling using the query by committee algorithm, Machine Learning 28 (1997) 133–168.

[20] Y. Fu, X. Zhu, B. Li, A survey on instance selection for active learning, Knowledge and Information Systems 35 (2013) 249 – 283.

[21] J.D. Gibbons, S. Chakraborti, Nonparametric statistical inference, in: International Encyclopedia of Statistical Science, 2011, pp. 977–979.

[22] J. Hu, J. Qi, Y. Peng, Q. Ren, Predicting electrical evoked potential in optic nerve visual prostheses by using support vector regression and case-based prediction, Information Sciences 290 (2015) 7–21.

[23] E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, Y. Zhu, A. Leetmaa, B. Reynolds, M. Chelliah, W. Ebisuzaki, W. Higgins, J. Janowiak, K.C. Mo, C. Ropelewski, J. Wang, R. Jenne, D. Joseph, The NCEP/NCAR 40-Year Reanalysis Project, Bullettin of the American Meteorological Society 77 (1996) 437–472.

[24] P. Kelley, R. Barry, Sparse spatial autoregressions, Statistics and Probability Letters 33 (1999) 291–297.

[25] C. Komurlu, M. Bilgic, Active inference and dynamic gaussian bayesian networks for battery optimization in wireless sensor networks, in: AI for Smart Grids and Smart Buildings, Papers from the 2016 AAAI Workshop.

[26] A. Kuwadekar, J. Neville, Relational active learning for joint collective classification models, in: 28th International Conference on Machine Learning, Omnipress, 2011, pp. 385–392.

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

[28] L. Li, X. Zhang, J.B. Holt, J. Tian, R. Piltner, Spatiotemporal interpolation methods for air pollution exposure, in: 9th Symposium on Abstraction, Reformulation, and Approximation, AAAI press, 2011.

[29] C. Loglisci, A. Appice, D. Malerba, Collective regression for handling autocorrelation of network data in a transductive setting, Journal of Intelligent Information Systems 46 (2016) 447–472.

[30] C. Loglisci, D. Malerba, Leveraging temporal autocorrelation of historical data for improving accuracy in network regression, Statistical Analysis and Data Mining 10 (2017) 40–53.

[31] S.A. Macskassy, Using graph-based metrics with empirical risk minimization to speed up active learning on networked data, in: 15th International Conference on Knowledge Discovery and Data Mining, ACM, 2009, pp. 597–606.

[32] M. McPherson, L. Smith-Lovin, J. Cook, Birds of a feather: Homophily in social networks, Annual Review of Sociology 27 (2001) 415–444.

[33] G. Melki, A. Cano, V. Kecman, S. Ventura, Multi-target support vector regression via correlation regressor chains, Information Sciences 415 (2017) 53–69.

[34] P.A.P. Moran, Notes on continuous stochastic phenomena, Biometrika 37 (1950) 17–23.

[35] J. Neville, B. Gallagher, T. Eliassi-Rad, T. Wang, Correcting evaluation bias of relational classifiers with network cross validation, Knowledge and Information Systems 30 (2012) 31–55.

[36] J. Neville, Ö. Şimşek, D.D. Jensen, Autocorrelation and relational learning: Challenges and opportunities, in: Workshop on Statistical Relational Learning and Its Connections to Other Fields, 21st International Conference on Machine Learning, 2004, pp. 74–81.

[37] M.E. Newman, Modularity and community structure in networks, Proceedings of the National Academy of Sciences of the United States of America 103 (2006) 8577–8582.

[38] O. Ohashi, L. Torgo, Spatial interpolation using multiple regression, in: 12th International Conference on Data Mining, IEEE Computer Society, 2012, pp. 1044–1049.

[39] E. Pasolli, F. Melgani, N. Alajlan, Y. Bazi, Active learning methods for biophysical parameter estimation, IEEE Transactions on Geoscience and Remote Sensing 50 (2012) 4071–4084.

[40] D. Pfeffermann, C.R. Rao, Handbook of Statistics, Vol.29, A Sample Surveys: Theory, Methods and Infernece, Elsevier B.V., 2009.

[41] S. Pravilovic, M. Bilancia, A. Appice, D. Malerba, Using multiple time series analysis for geosensor data forecasting, Information Sciences 380 (2017) 31–52.

[42] W.M. Rand, Objective criteria for the evaluation of clustering methods, Journal of the American Statistical Association. American Statistical Association 66 (1971) 846–850.

[43] T. Reitmaier, A. Calma, B. Sick, Transductive active learning a new semi-supervised learning approach based on iteratively refined generative models to capture structure in data, Information Sciences 293 (2015) 275 – 298.

[44] T. Reitmaier, B. Sick, Let us know your decision: Pool-based active training of a generative classifier with the selection strategy 4ds, Information Sciences 230 (2013) 106–131.

[45] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, T. Eliassi-Rad, Collective classification in network data, AI Magazine 29:3 (2008) 93–106.

[46] R.A. Simons, Erddap - the environmental research division's data access program, 2011.

[47] Y. Son, J. Lee, Active learning using transductive sparse bayesian regression, Information Sciences 374 (2016) 240 – 254.

[48] D. Stojanova, Estimating Forest Properties from Remotely Sensed Data by using Machine Learning, Master's thesis, Jožef Stefan International Postgraduate School, Ljubljana, Slovenia, 2009.

[49] D. Stojanova, M. Ceci, A. Appice, S. Deroski, Network regression with predictive clustering trees, Data Mining and Knowledge Discovery 25 (2012) 378–413.

[50] D. Stojanova, M. Ceci, A. Appice, D. Malerba, S. Deroski, Dealing with spatial autocorrelation when learning predictive clustering trees, Ecological Informatics 13 (2013) 22–39.

[51] Y. Wang, I. Witten, Induction of model trees for predicting continuous classes, in: 9th European Conference on Machine Learning, Faculty of Informatics and Statistics, University of Economics, Prague, 1997, pp. 128–137.

[52] R. Zafarani, H. Liu, Social computing data repository at ASU, 2009.

[53] D. Zhang, J. Yin, X. Zhu, C. Zhang, Collective classification via discriminative matrix factorization on sparsely labeled networks, in: 25th International on Conference on Information and Knowledge Management, ACM, 2016, pp. 1563–1572.