Transductive Relational Classification in the Co-training Paradigm

Michelangelo Ceci[1], Annalisa Appice[1], Herna L. Viktor[2], Donato Malerba[1], Eric Paquet[2,3], and Hongyu Guo[3]

[1] Dipartimento di Informatica, Università degli Studi di Bari "A. Moro", Italy
{ceci,appice,malerba}@di.uniba.it
[2] School of Electrical Engineering and Computer Science, University of Ottawa
{hlviktor,epaquet}@site.uottawa.ca
[3] Institute for Information Technology, National Research Council of Canada
{eric.paquet,hongyu.guo}@nrc-cnrc.gc.ca

Abstract. Consider a multi-relational database, to be used for classification, that contains a large number of unlabeled data. It follows that the cost of labeling such data is prohibitive. Transductive learning, which learns from labeled as well as from unlabeled data already known at learning time, is highly suited to address this scenario. In this paper, we construct multi-views from a relational database, by considering different subsets of the tables as contained in a multi-relational database. These views are used to boost the classification of examples in a co-training schema. The automatically generated views allow us to overcome the independence problem that negatively affect the performance of co-training methods. Our experimental evaluation empirically shows that co-training is beneficial in the transductive learning setting when mining multi-relational data and that our approach works well with only a small amount of labeled data.

Keywords: Transductive learning, Co-training, Multi-relational classification.

1 Introduction

Increasingly, sets of structured data including relational databases, spatial data and biological data, amongst others, are available for mining. In these settings, using supervised learning methods often becomes impractical, due to the high costs associated with labeling the data. To this end, during recent years, there has been a growing interest in learning algorithms capable of utilizing both labeled and unlabeled data for prediction tasks. In the literature, two main settings have been proposed to exploit the information contained in both labeled and unlabeled data, namely the semi-supervised setting and the transductive paradigm. The former is based on the principle of inductive learning, where the learned function is used to make predictions, both on currently known and future observations. Transductive learning, on the other hand, makes predictions for the

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 11–25, 2012.

set of unlabeled data that is already known at learning time, which simplifies the learning task. Another strength of transductive learning is that it allows the examples in the training (and working) set to be mutually dependent.

Research further suggests that such structured (or semi-structured) data repositories often consist of subsets that provide us with different, complementary viewpoints of the dataset. That is, the data may be analyzed by exploring multiple complementary *views*, which each brings additional information regarding the problem domain. For example, during social network analysis, a contact may be viewed by considering: links to the person, textual content of the webpage, meta-data, multimedia content, and so on. In a relational database, these units of analysis may naturally be modeled as a set of tables $T_1, \ldots, T_n$, such that each table T_i describes a specific type of object involved in the units of analysis, while foreign key constraints explicitly model the relationships between objects. Multi-view learning, which concerns the selection of multiple independent views to be used for classification, is based on this observation.

When the number of unlabeled data is high, it follows that multi-view learning may be successfully combined with a transductive learning approach. To this end, we introduce the CoTReC (Co-training for Transductive Relational Classification) method, which is inspired by the above-mentioned principle of multi-view learning. Our method differs from the standard setting, since these views are mutually independent and are constructed from separate subsets of the dataset. In our approach, as a first step, we create a number of uncorrelated training and working views. Next, during co-training, the transductive learner proceeds to build a model and to predict the class value of the working set views as accurately as possible. Our approach requires only a small amount of labeled data, in order to construct accurate models.

The use of co-training is motivated by the need of improving the accuracy of weak transductive classifiers. This weakness is due to the low number of labeled examples that transductive algorithms are required to work with. In fact, with co-training, we augment the training set of a classifier learned from a database view by using predictions of unlabeled examples from the classifiers learned on the other views. This is performed with an iterative learning process [2].

This paper is organized as follows. Section 2 introduces the related work. This is followed, in Section 3 with a discussion of the CoTReC approach. Section 4 presents experiments, while Section 5 concludes this paper.

2 Related Work

This section introduces related research concerning transductive learning, multi-view learning and co-training.

2.1 Transductive Learning

Transductive learning has been investigated in classical data mining for SVMs ([8] [11]), for k-NN classifiers ([12]) and even for general classifiers ([16]).

However, more recently, transductive learning has been also investigated in multi-relational data mining, which aims to discover useful patterns across multiple relations (tables) in a relational database [9]. Krogel and Scheffer ([15]) investigated a transformation (known as propositionalization) of a relational description of gene interaction data into a classical double-entry table, and then study transduction with the well-known transductive support vector machines. Taskar et al. ([25]) built a generative probabilistic model that captures interactions between examples, some of which have class labels, while others do not. However, given sufficient data, a discriminative model generally provides significant improvements in classification accuracy over generative models. Malerba et al. [19] proposed a relational classification algorithm that works in a transductive setting and employs a probabilistic classification approach. The transductive learning strategy iterates on a k-NN based re-classification of labeled and unlabeled examples, in order to identify borderline examples, and uses the relational probabilistic classifier Mr-SBC to bootstrap the transductive algorithm.

2.2 Multi-view Learning

Multi-view learning concerns the learning from multiple independent sets of features, i.e. views, of the data. This framework has been applied to real-world applications such as information extraction [2], face recognition [18] and locating users in a WiFi environment [23], amongst others. In essence, multi-view learning utilizes experiential inputs to explore diverse representations so that an increasingly variety of problems may be recognized, formulated and solved. In this way, the strengths of each view are amplified and the weaknesses are alleviated. This allows for cooperation, knowledge sharing and knowledge fusion.

Formally, a multi-view problem with n views may be seen as n uncorrelated or weakly dependent features sets [9]. Here, the correlation estimates the departure of two variables (views) from independence. If we consider two views as being correlated, it implies that knowing the first view helps in predicting the second. On the other hand, if knowing the first brings no (or little) knowledge about the second, the correlation coefficient should be near to zero.

Intuitively, multi-view learning is well suited for relational data mining. Within the relational database context, it follows that such sets of disjoint features are typically contained in multiple relations. That is, a relational database schema, as designed by a domain expert, usually groups attributes into relations (or entities in an extended entity-relationship (EER) diagram) with very close semantic meaning. Interested readers are referred to [9] for a detailed discussion of a multi-view learning methodology within the relational database setting.

2.3 Co-training

The iterative co-training framework was originally introduced in [2]. The idea is to split the set of predictor attributes $\mathbf{X}$ into two disjoint subsets $\mathbf{X1}$ and $\mathbf{X2}$. Each labeled example (x, y) is viewed as $(x1, x2, y)$ where $x1$ contains the values of the attributes in $\mathbf{X1}$ and $x2$ the values of the attributes in $\mathbf{X2}$. Then, two

classifiers $f_1(\cdot)$ and $f_2(\cdot)$ (where $f_1 : \mathbf{X1} \rightarrow Y$ and $f_2 : \mathbf{X2} \rightarrow Y$) are learned by bootstrapping one another with labels for the unlabeled data. Intuitively, the initial classifiers f_1 and f_2 are learned over a small sample and the iterative bootstrap takes unlabeled examples $(x1, x2)$ for which f_1 is confident, but f_2 is not (or vice-versa) and using the reliable classification to label such examples for the learning algorithm on f_2 (or f_1), while improving the other classifier.

Two assumptions are made in co-training: *i)* (*Compatibility* assumption) The example distribution is compatible with the classifier $f : \mathbf{X} \rightarrow Y$. That is, either attribute set suffices to learn the target $f(\cdot)$ such that for each example x: $f_1(x) = f_2(x) = f(x)$. *ii)* (*Conditionally independence* assumption) The predictor attributes in **X1** and the predictor attributes in **X2** are conditionally independent given the class label. Formally, $P(x1|f(x), x2) = P(x1|f(x))$ and $P(x2|f(x), x1) = P(x2|f(x))$. While assumption *ii)* justifies an iterative approach, assumption *i)* can be significantly weakened while still permitting the use of unlabelled data to iteratively boost weak classifiers. Accordingly, several studies [21] [14] have shown that co-training may improve classifier performance even when the assumptions are violated to some extent.

The iterative co-training approach has been successfully applied to several learning problems, including named entity classification [5], text classification [22] and image classification [17]. Most of these works extract the multiple views by trying to meet empirically the co-training assumptions (or weakening of them) on the training domain. However, unlike the work presented in this paper, these techniques do not address the problem of extracting multiple views from data spanned in multiple tables of a relational database by trying to meet the compatibility and conditionally independence assumptions.

3 The CoTReC Method

Let D be a set of units of analysis (examples) whose description is spread in multiple tables of a relational database. Examples can be labeled according to an unknown target function whose range is a finite set $Y = \{C_1, \ldots, C_L\}$. The transductive classification problem is formalized as follows: *Given* a training set $TS \subset D$, and a working set $WS = D - TS$ with unknown target values. The problem is to *predict* the class value of each example in the working set WS which is as accurate as possible. The learner receives full information (including labels) for the examples in TS and partial information (without labels) for the examples in WS and is required to predict the class only of the examples in WS.

The use of the co-training framework to solve this problem is illustrated in Algorithm 1. The algorithm takes as input n training and working views[1] and returns the set of labeled working examples. In the while loop, the algorithm iterates by bootstrapping the data labeling on the different views. In particular, it classifies examples in each view and identifies the examples for which the classification is reliable. These examples are added to the other training views (see

[1] Training and Working views are defined according to the same schema, but are used on the training and working database, respectively.

Algorithm 1, lines 16-18) for subsequent iterations. The associated label is identified according to the Bayesian Optimal Classifier [20] that permits to combine the effect of (possibly multiple) reliable classifications of the same example (see Algorithm 1, lines 13-14). Formally, it is defined as:

$$BOC(e, Y'', h_1, \ldots, h_n, a_1, \ldots, a_n) = \underset{c \in Y''}{\operatorname{argmax}} \sum_{j=1,\ldots,n} P(c|h_j) \times a_j \quad (1)$$

where $Y'' \subseteq Y$ is a subset of the set of labels, $P(c|h_j)$ is equal to one if the classifier h_j associates the example e with the label c, 0 otherwise and a_j estimates the probability $P(h_j|A_j)$ and is computed as the accuracy of the classifier h_j on the training view A_j (see Algorithm 1, line 9). The classifier h_j may utilize one of many available propositional classification algorithms.

It is noteworthy that decisions on the classifications are propagated coherently through the training views (and added to L). Once a stopping criterion is satisfied, the Bayesian Optimal Classifier is used in order to classify the remaining unlabeled examples according to the classifiers learned on training views obtained after the last iteration (see Algorithm 1, lines 25-38).

Three stopping criteria are considered in the algorithm, that is, all the examples have been classified; no example is added to the training views during the last iteration; a maximum number of iterations (MAX_ITERS) is reached.

Two issues remain to be discussed: *i)* how views are constructed and *ii)* how the reliable classifications are identified. These issues are discussed in the next.

3.1 Constructing Multi-views from Relational Data

In a relational setting, there is a database $\Re$, which consists of a target table T_{target} and a set of background relations $\{T_i\}_1^n$. The target relation T_{target} has a target variable Y. That is, each tuple in this table (target tuple) is associated with a class label which belongs to Y. Typically, the relational classification task is to find a function $F(x)$ which maps each target tuple x to the category set Y:

$$Y = F(x, T_{1\cdots n}, T_{target}), x \in T_{target} \quad (2)$$

This relational classification task may be mapped to a multi-view learning problem, as follows. Join paths link the target relation with the other relations in the relational database, through following foreign key links. That is, foreign key attributes link to primary keys of other tables: this link specifies a *join* between two tables, and a foreign key attribute of table T_2 referencing table T_1 is denoted as $T_2.T_{1\_key}$. Each one of these paths provides a complementary, potentially information-rich view of the target concept to be learned.

The next section discusses the multi-view construction process which corresponds to the first four steps of the MRC multi-view learner reported in [9].

Multi-View Construction. Algorithm 2 highlights the steps we following when constructing the multiple views. As shown in Algorithm 2, the method initially propagates and aggregates information to form multiple views from a

Algorithm 1. CoTReC

Require: $A_1, \ldots, A_n$ training views, $W_1, \ldots, W_n$ working views.
Ensure: L working examples associated with labels

$L \leftarrow \emptyset$;
while no stopping criterion is satisfied **do**
  for all $i = 1, \ldots, n$ **do**
    Train the classifier h_i on the training view A_i;
    Classify all the examples in W_i according to the classifier h_i;
    Compute the ranked of the class predicted by h_i for each example in W_i;
    Sort the examples in W_i by reliability in decreasing order;
    $T_i \leftarrow$ the reliability threshold for W_i;
    $a_i \leftarrow$ accuracy of the classifier h_i on the training view A_i;
  end for
  for all $i = 1, \ldots, n$ **do**
    for all $e \in W_i$ with reliability greater than T_i **do**
      $Y' \leftarrow \{c \in Y |\ \exists j \in [1, \ldots, n],\ h_j(e) = c$ with reliability greater than $T_j\}$;
      $label \leftarrow BOC(e, Y', h_1, \ldots, h_n, a_1, \ldots, a_n)$
      for all $j = 1, \ldots, n$ **do**
        if $i \neq j$ **then**
          $A_j = A_j \cup \{(e, label)\}$;
        end if
        $W_j = W_j - \{e\}$;
      end for
      $L = L \cup \{(e, label)\}$;
    end for
  end for
end while
for all $i = 1, \ldots, n$ **do**
  Train the classifier h_i on the training set A_i;
  Classify all the examples in W_i according to the classifier h_i;
  $a_i \leftarrow$ accuracy of the classifier h_i on the training view A_i;
end for
for all $i = 1, \ldots, n$ **do**
  for all $e \in W_i$ **do**
    $label \leftarrow BOC(e, Y, h_1, \ldots, h_n, a_1, \ldots, a_n)$
    for all $j = 1, \ldots, n$ **do**
      $W_j = W_j - \{e\}$;
    end for
    $L = L \cup \{(e, label)\}$;
  end for
end for
RETURN L

relational database. Subsequently, the view validation step identifies a subset of uncorrelated views. This set of views may subsequently be used in a transductive learning setting, to co-train classifiers.

1) Information Propagation Stage: The Information Propagation Stage constructs training data sets for use by a number of view learners. The original

Algorithm 2. Multi-view Construction Algorithm

Require: A database $\Re = \{T_{target}, T_1, T_2, \cdots, T_n\}$, a view learner $\mathcal{L}$, a meta learner $\mathcal{M}$, and a view validation data $\mathcal{T}_v$ from $\Re$.

1: Propagate and aggregate information in $\Re$, forming candidate view set $\{V_d^1, \cdots, V_d^n\}$;
2: Remove candidate views with view learners' accuracy less than 50% from $\{V_d^1, \cdots, V_d^m\}$, forming view set $V^{'}$;
3: Generate a validation data set $\mathcal{T}_v^{'}$, using $\mathcal{T}_v$ and $V^{'}$;
4: Select a view feature set $\mathcal{A}^{'}$ from $\mathcal{T}_v^{'}$;
5: Let validated view set $V = \emptyset$;
6: **for each** view V^i ($V^i \in V^{'}$) which has at least one attribute in $\mathcal{A}^{'}$ **do**
7: V.add(V^i);
8: **end for**
9: RETURN V.

relational database is used as input and the process starts from the target table. Our approach was inspired by the so-called *tuple id* propagation, as introduced in CrossMine [26], which is a technique for performing virtual joins between tables. The target and background relations are linked together through their foreign keys, which are identified by means of the *tuple identifiers*. In SQL, this is done by an equi-join of the tables over the values of the key identifiers.

2) Aggregation Stage: The Aggregation Stage summarizes the information embedded in the tuples associated with one to many relationships (i.e., one target tuple is associated with multiple entities in a background relation) by converting them into one row. This procedure is applied to each of the data sets constructed during the Information Propagation Stage. To achieve this objective, aggregation functions are applied. By applying the basic aggregation functions in SQL, new features are created to summarize information stored in the multiple tuples. For nominal attributes, we employ the *count* function. For numeric values, the *count, sum, average, maximum, minimum and standard deviation* are calculated.

3) Multiple Views Construction Stage: In the third phase of the algorithm, the Multiple Views Construction Stage constructs various hypotheses on the target concept, based on the multiple training data sets given by the Aggregation Stage. To this end, view learners such as decision trees or support vector machines (SVMs) are used in order to learn the target concept from each view of the database separately. In this stage, a number of view learners are trained.

4) View Validation Stage: All view learners constructed in the Multiple Views Construction Stage are then evaluated in the View Validation Stage. This processing is needed to ensure that they are, indeed, able to learn the target concept. In addition, strongly uncorrelated view learners are preferred. The aim of this stage is to maintain only views that are able to accurately predict the class.

As a first step, we discard all views that perform worse than random guessing. We eliminate all views with an error rate lower than 50%. Research suggests that disjoint views are preferred by multi-view learning, due to the inherent philosophy of diverse viewpoints [6,5,9]. Following this line of thought, we employ

a *Correlation-based View Validation (CVV)* algorithm that uses the heuristic goodness, from test theory, as utilized by the CFS subset feature selection approach of Hall [10]. The heuristic 'goodness' of a subset of features [10] is:

$$\mathcal{C} = K\overline{R_{cf}}/\sqrt{K + K(K-1)\overline{R_{ff}}} \tag{3}$$

where $\mathcal{C}$ is the heuristic 'goodness' of a selected feature subset. K is the number of features in the selected subset. $\overline{R_{cf}}$ calculates the average feature-to-class correlation, and $\overline{R_{ff}}$ stands for the average feature-to-feature dependence.

We adopt the *Symmetrical Uncertainty (𝒰)* [9] to calculate $\overline{R_{cf}}$ and $\overline{R_{ff}}$. This measure is a modified version of the *information gain* measure which compensates for information gain's bias toward attributes with more values.

The *Symmetrical uncertainty* is defined as follows:
Given features X and Y,

$$\mathcal{U} = 2.0 \times \left[\frac{InfoGain}{H(Y) + H(X)}\right] \quad \text{where } H(Y) = -\sum_{y \in Y} p(y)\log(p(y)).$$

In summary, during view validation, each view is called upon to give a prediction of the target class, against a validation data set. Different subsets of the views are then ranked, according to their C values and the subset with the highest rankings is kept. These views are then used as the input to CoTReC.

3.2 Reliability Measures

In order to identify reliable classifications, it is necessary to use measures that quantify how reliable the classification is. Indeed, many classification algorithms return confidence measures (see, for example naïve Bayes, k-NN and SVM classifiers). However, such measures are algorithm-dependent and cannot be used in a general framework. Several algorithm-independent reliability measures have been proposed in the literature. In [4], twelve reliability measures are compared by concluding that no measure, independently considered, can be robust enough in measuring the reliability of any class in any domain. On the basis of this consideration, authors proposed to learn a decision tree which aggregates basic measures by obtaining a robust piecewise reliability measure. Delany et al. [7], on the other hand, propose three reliability measures that are domain-independent, monotonic in terms of the reliability and do not require a training phase.

In this paper, we employ the results of both these studies. In particular, from [4], we use the idea of combining several reliability measures, while from [7] we identify the algorithm and domain-independent measures to be combined. Recall that we aim to evaluate the reliability of the label predicted for an example e, by considering labels associated to examples in the k-nearest neighborhood.

As basic reliability measures, we select the following three measures:

$$AvgNUNI(e,k) = \sum_{i=1}^{k} IndexOfNUN_i(e) \tag{4}$$

$$SR(e,k) = \frac{\sum_{i=1}^{k} sim(e, NLN_i(e))}{\sum_{i=1}^{k} sim(e, NUN_i(e))} \tag{5}$$

$$SRK(e,k) = \frac{\sum_{i=1}^{k} sim(t, NN_i(e)) \cdot 1(t, NN_i(e))}{\sum_{i=1}^{k} sim(t, NN_i(e)) \cdot (1 - 1(t, NN_i(e)))} \tag{6}$$

where:

- $NN_i(e)$: the i-th nearest neighbor of e,
- $NLN_i(e)$: the i-th nearest like (same label) neighbor of e,
- $NUN_i(e)$: the i-th nearest unlike (different label) neighbor of e,
- $IndexOfNUN_i(e)$ is the index of $NN_i(e)$. The index is the ordinal ranking of the example in the list of nearest neighbors,
- $sim(a,b)$ is the similarity between examples a and b,
- $1(a,b) = 1$ if $label(a) = label(b)$; 0 otherwise .

These measures are used as follows. $AvgNUNI(e,k)$ measures how close e is to the first k nearest unlike neighbors (NUNs); $SR(e,k)$ measures the ratio of the similarity between e and its first k nearest like neighbors (NLNs) and the similarity between e and its first k NUNs; $SRK(e,k)$ is similar to $SR(\cdot,\cdot)$ except that, it considers only the first k examples, independent from the class label.

The similarity function $sim(a,b)$ has range in $[0,1]$ and is computed in a standard way. In particular, we use Manhattan distance for continuous predictor attributes and the Hamming distance for discrete ones. More formally:

$$sim(a,b) = 1 - \frac{m^{(C)} d^{(C)}(a,b) + m^{(D)} d^{(D)}(a,b)}{m^{(C)} + m^{(D)}} \tag{7}$$

where $m^{(C)}$ ($m^{(D)}$) is the number of continuous (discrete) predictor attributes in the view and $d^{(C)}(a,b)$ ($d^{(D)}(a,b)$) is the Euclidean or the Manhattan (Hamming) distance computed on continuous (discrete) attributes. Each continuous attribute is normalized through a linear scaling into $[0,1]$. This guarantees that $sim(a,b)$ belongs to $[0,1]$ and attributes uniformly contribute to $sim(a,b)$.

The aggregated reliability measure $ARM(e,k)$ is computed by averaging normalized values of $AvgNUNI(e,k)$, $SR(e,k)$ and $SRK(e,k)$, as suggested in [4]. Again, the normalization is performed on the interval $[0,1]$:

$$ARM(e,k) = \frac{1}{3}(AvgNUNI'(e,k) + SR'(e,k) + SRK'(e,k)) \tag{8}$$

where $AvgNUNI'(e,k)$, $SR'(e,k)$ and $SRK'(e,k)$ are the normalized versions of $AvgNUNI(e,k)$, $SR(e,k)$ and $SRK(e,k)$, respectively. With a single reliability value we can rank confidences in decreasing order and then select the top examples for which $ARM(e,k)$ is greater than the reliability threshold T_i. This threshold is empirically identified at each iteration, for each view.

3.3 Computing Reliability Thresholds

In the CoTReC algorithm, we use the $ARM(e,k)$ computed for each training example to compute reliability thresholds. We identify the reliability threshold that allows us to maximize the AUC (Area under the ROC curve) of a threshold-based classifier G_θ that solves the following two classes classification problem: $(+)$ the label of a training example is correctly predicted according to h_i; $(-)$ the label of a training example is incorrectly predicted according to h_i-

Algorithmically, for each candidate threshold θ, the AUC of the classifier G_θ that classifies as $(+)$ examples in A_i for which $ACM(e,k) \geq \theta$ and classifies as $(-)$ examples in A_i for which $ACM(e,k) < \theta$ is computed. The set of candidate thresholds Θ_i is computed by ranking examples in A_i according to $ACM(e,k)$ and by considering the middle values of reliabilities of two consecutive examples.

Therefore, each threshold T_i is computed as:

$$T_i = \underset{\theta \in \Theta}{\operatorname{argmax}} \; AUC(G_\theta, A_i) \tag{9}$$

The advantages of this solution are: *i)* thresholds are local to each view for each iteration and *ii)* thresholds are computed on the current training set.

3.4 Implementation Considerations

With a naïve implementation, at each iteration, for each view, it is necessary to compute the similarities between the training examples and all the examples. These similarities are used both for computing reliability thresholds and deciding which working examples can be considered as reliably classified. The cost of computing similarities, in the worst case, is $O(l_A \times (l_A + l_W) \times MAX\_ITERS)$, where l_A is the total number of training examples and l_W is the total number of working examples. In our implementation, we do not recompute the same similarities from one iteration to the next. Moreover, we organize examples according to an *r-tree* structure in order to compute similarities between close examples in logarithmic time. In this way, at the first iteration, we compute similarities with a time complexity $O(log(l_A) \times log(l_A + l_W))$. From the second iteration, we only compute similarities between examples kept in the working set and examples lastly moved in the training set.

4 Experiments

The empirical evaluation of our algorithm was carried out on three real world datasets: PKDD 99 discovery challenge database, Mutagenesis database (which have been both used to test several MRDM algorithms), and North West England (NWE) Census Database. The CoTReC algorithm is evaluated on the basis of the average accuracy on the same 10-fold cross-validation (10-CV) against each database. For each database, the target table is first divided into 10 blocks of nearly equal size and then a subset of tuples related to the tuples of the target

table block is extracted by means of foreign key constraints. In this way, 10 database instances are created. For each trial, CoTReC is trained on a single database instance and tested on the hold-out nine database instances, forming the working set. It should be noted that the accuracies reported in this work are thus not directly comparable to those reported in other research, which uses a standard 10-fold CV method. This is due to our unusual experimental design, imposed by the transductive co-training paradigm. Indeed, unlike the standard CV approach, here one fold at a time is set aside to be used as the training set (and not as the test set). Small training set sizes allow us to validate the transductive approach, but may result in higher error rates as well.

4.1 Datasets

The first database that we consider is the PKDD 99 discovery challenge database [1], as introduced earlier. Recall that this database concerns the task of determining whether a client will default on his loan. The database contains eight tables. The target Loan table consists of 682 records, including a class attribute status which indicates the status of the loan, i.e. A (finished and good), B (finished but bad), C (good but not finished), or D (bad and not finished). The background information for each loan is stored in the relations *Account, Client, Order, Transaction, Credit Card, Disposition* and *Demographic*. All background relations relate to the target table through directed or undirected foreign key chains. The aim of the learning problem is to classify if a loan is at risk or not.

The Mutagenesis database is related to the problem of identifying some mutagenic compounds [24]. We have considered, similarly to most experiments on relational data mining algorithms, the "regression friendly" dataset consisting of 188 molecules. It consists of data obtained with the molecular modeling package QUANTA. For each compound, it contains the atoms, bonds, bonds types, atom types, and partial charges on atoms plus indicators ind1, inda, logp, and lumo.

The NWE Census database is obtained from both census and digital map data provided by the European project SPIN!. These data concern Greater Manchester, one of the five counties of NWE. Greater Manchester is divided into 214 censual wards. Mortality rate as well as some indexes of the deprivation (Jarman Underprivileged Area Score, Townsend Index, Carstairs Index and Department of the Environment Index) are available at ward level. The goal of the classification task is to predict the value of the Jarman index (low or high) deprivation factor by exploiting the other deprivation factors, mortality rate and geographical factors, represented in some linked topographic maps. Spatial analysis is possible thanks to the availability of vectorized boundaries of the 1998 census wards as well as of other Ordnance Survey digital maps of NWE where urban area (115 spatial objects), green area (9), road net (1687), rail net (805) and water net (716) can be found. Topological non-disjoint relationships between wards and objects in all these layers are materialized as relational tables.

Table 1. Average number of views extracted and validated from MV.

DB	PKDD 99	Mutagensis	NWE
No of views	2	2	6

4.2 Experimental Results

In the experiments, we have considered four learning systems as base classifiers. Specifically, C4.5, the RIPPER rule learner, Naïve Bayes (NB), SMO classifier [13] and 1-nearest neighbours (1-NN) which are all implemented in the Weka system. CoTReC is run by varying k (see (8)) among three values expressed as a percentage (1%,2% and 3%) of the number of the examples; MAX_ITERS is set to the number of unlabeled examples stored in the working database. In Table 1, we report the average number of views extracted within CoTreC. Each view provides a propositionalization of the corresponding database.

In Table 2, we report both the 10-CV average accuracies and the average numbers of iterations for PKDD 99 discovery challenge database and Mutagenesis database. We have chosen these databases, since they have been used to benchmark several state-of-the-art muilti-relational data mining methods. Results, as shown in Table 2, indicate that there is no distance which consistently leads to a better classification accuracy. The accuracy of classification generally improves by increasing the k value. Moreover, the classification accuracy remains high (more than 87.5%) on PKDD 99 discovery challenge database, independently of the base classifier. However, the best accuracy is obtained by using NB. This trend is confirmed when analyzing the Mutagenesis database, where we observe an higher variability of the accuracy by varying the basic classifier. For Mutagenesis, we observe that the best accuracies are obtained with the SMO classifier which, however, is less stable than NB when varying k. By considering the average number of iterations, we observe that it is generally low, except for the case of C4.5 . Here, CoTReC tends to move a small number of examples in the training set at each iteration, since the trees do not significantly change from one iteration to the next one. On the basis of these considerations, NB can be reasonably considered as the classifier that better interacts with CoTReC.

To evaluate the predictive performance of the CoTReC algorithm, we compared our method with the only relational transductive classifier currently available in the literature, namely the TRANSC approach [19]. TRANSC iterates on a k-NN based re-classification of labeled and unlabeled examples, in order to identify class borderline examples, and uses the relational probabilistic classifier Mr-SBC [3] to bootstrap the transductive algorithm. Although similar to CoTReC, TRANSC does not use co-training and does not exploit multi-views. The experimental results reported in [19] include the predictive performance of the TRANSC method as well as the relational naïve Bayesian classifier Mr-SBC [3] against the Mutagenesis and NWE databases. We ran CoTReC against these two databases, using the same CV partitioning as reported in [3]. We compared the accuracy obtained by CoTReC with the best results of the TRANSC and Mr-SBC methods against these two databases, as presented in [3]. We depict

Table 2. PKDD 99 discovery challenge database and Mutagenesis database: Average accuracies/iterations obtained with different k values. Column "Distance Measure" refers to the used distance measure for continuous attributes. The average number of iterations is approximated to the unity.

Learner	Distan.	PKDD 99 discovery challenge			Mutagenesis		
	Measu.	k=1%	k=3%	k=5%	k=1%	k=3%	k=5%
C4.5	Euclid.	87.94% /74	87.94% /124	87.94% /137	50.88% /6	77.51% /4	82.84% /6
	Manha.	87.94% /129	87.94% /66	87.94% /111	53.25% /5	73.96% /5	80.47% /4
RIPPER	Euclid.	87.94% /4	87.94% /4	87.94% /1	37.86% /4	79.88% /3	82.24% /2
	Manha.	87.94% /1	87.94% /3	87.94% /4	43.78% /5	78.69% /3	82.24% /3
NB	Euclid.	88.11% /2	88.11% /2	88.11% /2	69.23% /5	84.02% /3	74.55% /5
	Manha.	88.11% /2	88.11% /2	88.11% /2	68.08% /3	87.57% /4	85.79% /6
SMO	Euclid.	87.94% /9	87.94% /5	87.94% /6	60.94% /5	78.69% /3	88.16% /3
	Manha.	87.94% /10	87.94% /15	87.94% /6	39.64% /7	78.69% /3	88.75% /2
1-NN	Euclid.	87.62% /1	87.62% /1	87.62% /1	78.69% /1	78.69% /1	78.69% /1
	Manha.	87.62% /1	87.62% /1	87.62% /1	78.69% /1	78.69% /1	78.69% /1

Table 3. Mutagenesis and NWE databases: average accuracies obtained with CoTReC, TRANSC and Mr-SBC. CoTReC is run with NB learner as basic classifier, k=5% and Manhattan distance.

System/DB	Mutagenesis	NWE
CoTReC	85.79%	83.33%
TRANSC	82.89%	81.96%
Mr-SBC	75.08%	77.29%

the comparison results in Table 3. Results indicate that the CoTReC method outperformed both the TRANSC and the Mr-SBC algorithms for the two tested databases, in terms of the obtained accuracies. For example, when contrasting CoTReC with the Mr-SBC method, our algorithm improved the accuracy against the Mutagenesis and NWE databases by 10.71% and 6.04%, respectively. Compared to the TRANSC strategy, the CoTReC algorithm was able to reduce the predictive error against the Mutagenesis and NWE databases by 2.90% and 1.37%, respectively. This reduction is mainly due to the co-training approach.

In summary, these results show that, with only a small number of labeled data, CoTReC can successfully construct accurate classification models, through benefiting from the use of co-training and transductive learning paradigms.

5 Conclusions

Numerous real-world applications contain complex and heterogeneous data, which are naturally modeled as several tables in a relational database. Often, such data repositories consist of a small number of labeled data together with a set of unlabeled data. In this paper, we have investigated the combination of transductive inference with co-training for the classification task in order to successfully mine such data. Our method exploits multi-views extracted from a relational database in a co-training schema. Multi-views are extracted by following the foreign key chains from relational databases and these views enable us

to deal with the relational structure of the data. Co-training allows us to boost the classification of examples within an iterative learning process.

The proposed classifier (CoTReC) has been compared to the TRANSC transductive relational classifier and the Mr-SBC inductive relational classifier. Results show that CoTReC outperforms both systems. As future work, we intend to extend the empirical investigation to corroborate our intuition that transductive inference has benefits from co-training when applied to multi-views extracted from a relational database. Our work will further include a thorough investigation of the robustness and efficiency of CoTReC against further databases. In addition, experimental evaluation on the influence of the different confident measures on the CoTReC approach will be conducted. We also plan to compare our method with other state-of-the-art co-training and multi-relational learning strategies. It would also be interesting to extend the CoTReC algorithm to other relational data such as spatial data, deep-web data and social network data.

Acknowledgment. This work is supported in fulfillment of the research objectives of the PRIN 2009 Project "Learning Techniques in Relational Domains and Their Applications" funded by the Italian Ministry of University and Research (MIUR).

References

1. Berka, P.: Guide to the financial data set. In: Siebes, A., Berka, P. (eds.) PKDD 2000 Discovery Challenge (2000)
2. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Workshop on Computational Learning Theory (1998)
3. Ceci, M., Appice, A., Malerba, D.: Mr-SBC: A Multi-relational Naïve Bayes Classifier. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 95–106. Springer, Heidelberg (2003)
4. Cheetham, W., Price, J.: Measures of Solution Accuracy in Case-Based Reasoning Systems. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 106–118. Springer, Heidelberg (2004)
5. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 100–110 (1999)
6. Dasgupta, S., Littman, M.L., McAllester, D.A.: PAC generalization bounds for co-training. In: NIPS, pp. 375–382 (2001)
7. Delany, S.J., Cunningham, P., Doyle, D., Zamolotskikh, A.: Generating Estimates of Classification Confidence for a Case-Based Spam Filter. In: Muñoz-Ávila, H., Ricci, F. (eds.) ICCBR 2005. LNCS (LNAI), vol. 3620, pp. 177–190. Springer, Heidelberg (2005)
8. Gammerman, A., Azoury, K., Vapnik, V.: Learning by transduction. In: Proc. of the 14th Annual Conference on Uncertainty in Artificial Intelligence, UAI 1998, pp. 148–155. Morgan Kaufmann (1998)
9. Guo, H., Viktor, H.L.: Multirelational classification: A multiple view approach. Knowledge and Information Systems: An International Journal 17, 287–312 (2008)

10. Hall, M.: Correlation-based feature selection for machine learning, Ph.D diss., Waikato Uni. (1998)
11. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proc. of the 16th International Conference on Machine Learning, ICML 1999, pp. 200–209. Morgan Kaufmann (1999)
12. Joachims, T.: Transductive learning via spectral graph partitioning. In: Proc. of the 20th International Conference on Machine Learning, ICML 2003 (2003)
13. Keerthi, S.S., Shevade, S.K., Bhattacharyya, C., Murthy, K.R.K.: Improvements to platt's smo algorithm for svm classifier design. Neural Computation 13(3), 637–649 (2001)
14. Kiritchenko, S., Matwin, S.: Email classification with co-training. In: Proceedings of the 2001 Conference of the Centre for Advanced Studies on Collaborative Research, CASCON 2001, p. 8. IBM Press (2001)
15. Krogel, M.-A., Scheffer, T.: Multi-relational learning, text mining, and semi-supervised learning for functional genomics. Machine Learning 57(1-2), 61–81 (2004)
16. Kukar, M., Kononenko, I.: Reliable Classifications with Machine Learning. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) ECML 2002. LNCS (LNAI), vol. 2430, pp. 219–231. Springer, Heidelberg (2002)
17. Levin, A., Viola, P., Freund, Y.: Unsupervised improvement of visual detectors using co-training. In: Proceedings of the Ninth IEEE International Conference on Computer Vision, ICCV 2003, Washington, DC, USA, vol. 2, pp. 626–637. IEEE Computer Society (2003)
18. Li, S.Z., Zhu, L., Zhang, Z., Blake, A., Zhang, H., Shum, H.-Y.: Statistical Learning of Multi-view Face Detection. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 67–81. Springer, Heidelberg (2002)
19. Malerba, D., Ceci, M., Appice, A.: A relational approach to probabilistic classification in a transductive setting. Eng. Appl. Artif. Intell. 22, 109–116 (2009)
20. Mitchell, T.: Machine Learning. McGraw-Hill, New York (1997)
21. Muslea, I., Minton, S., Knoblock, C.A.: Active + semi-supervised learning = robust multi-view learning. In: Proceedings of the Nineteenth International Conference on Machine Learning, ICML 2002, pp. 435–442. Morgan Kaufmann Publishers Inc., San Francisco (2002)
22. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: CIKM, pp. 86–93. ACM (2000)
23. Pan, S., Kwok, J., Yang, Q., Pan, J.: Adaptive localization in a dynamic wifi environment through multi-view learning. In: AAAI 2007, Menlo Park, CA, pp. 1108–1113 (2007)
24. Srinivasan, A., Muggleton, S., King, R.D., Sternberg, M.J.E.: Mutagenesis: Ilp experiments in a non-determinate biological domain. In: Wrobel, S. (ed.) Proc. of the 4th Inductive Logic Programming Workshop, pp. 217–232. GMD-Studien (1994)
25. Taskar, B., Segal, E., Koller, D.: Probabilistic classification and clustering in relational data. In: Nebel, B. (ed.) IJCAI, pp. 870–878. Morgan Kaufmann (2001)
26. Yin, X., Han, J., Yang, J., Yu, P.S.: Crossmine: Efficient classification across multiple database relations. In: ICDE 2004, Boston, pp. 399–410 (2004)