Relational Classification Using Automatically Extracted Relations by Record Linkage

Christine Preisach, Steffen Rendle, and Lars Schmidt-Thieme

Machine Learning Lab, University of Hildesheim, Germany,
{preisach,srendle,schmidt-thieme}@ismll.uni-hildesheim.de

Abstract. Relational classifiers often outperform traditional classifiers which assume that objects are independent. For applying relational classifiers relations are required. Since data often is noisy and unstructured, these relations are not given explicitly, but need to be extracted.
In this paper we show a framework for relational classification that first automatically extracts relations from such a noisy database and then applies relational classifiers. For extracting relations we use techniques from the field of record linkage that learn the characteristics for a relation from pairwise features. With the proposed framework relational classifiers can be applied without requiring manual annotation.

1 Introduction

Relational classifiers consider the category or the attributes of related objects as predictor variables instead of only using local variables of an instance. Relational classifiers proved to result in high quality predictions for different tasks [4, 11, 12, 15]. Often they outperform state-of-the-art non-relational classifiers like Support Vector Machines. Recently, relational methods using *collective inference* [10] received attention. *Collective inference* procedures are iterative algorithms, that exploit the relational autocorrelation of variables of connected entities. One of the key problems of any relational classifier is that it can only be applied if relations are available. But in many tasks relations are not given explicitly and in order to apply relational classifiers they have to be manually annotated. In this paper we show how relations can be automatically extracted so that relational classifiers can be used without the costs of annotation by human experts.

A typical example for the application of relational classifiers is to predict the topic of research papers. Here, the relations between two papers could be *SameAuthor*, *SameVenue* or *SamePublisher*. Papers with unknown topic can be classified using these relations. The problem is, that the relations often are not explicitly given and it is not straightforward to extract them. Instead a database like in table 1 is given. As one can see, the true relations like *SameVenue* are not given. For relational classification these relations have to be extracted, e.g. $SameVenue(p_2, p_3)$ and $SameAuthor(p_1, p_2)$. A database like depicted in table 1 could have been built by crawling the web or crawling citations in papers, etc. The differences between identical concepts can emerge from different citation styles or by inherent ambiguities like a conference can either be called

'NIPS 2003' or '17th conference on neural information processing systems' – for a domain expert both statements are correct and in fact the same.

ID	Author	Paper title	Venue
p_1	Freund, Y.	Boosting a weak learning algorithm by majority.	3rd annual workshop on computational learning theory
p_2	Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby	Prediction, and query by committee	advances in neural information processing systems
p_3	S Rosset, E Segal	Boosting density estimation	NIPS 2003

Table 1. Example data from the domain of paper topic classification.

Besides the domain of research papers there are plenty of other domains where relations have to be extracted before a relational classifier can be applied. A further example is to categorize items of online-shops – for example categorize by product types like 'digital camera', 'laptop' or 'washing machine'. Here again the databases typically do not state the relations like *SameBrand* or *SameCountryOfProduction* explicitly.

This paper is structured as follows. First we show the general framework for multi-relational classification from raw noisy data. The framework consists of a relation extraction step and a multi-relational classification step. Next, we describe the extraction steps in detail. Afterwards, a classifier is presented that ensembles relational classifiers learned from several weighted binary relations. In the evaluation we show that the proposed ensemble of simple relational classifiers outperforms non-relational classifiers like an SVM on the task of paper classification. Secondly we show that the proposed learned linkage models are successful in finding the true relations.

The main contributions of this paper are: (1) we present a framework that allows to apply relational learning on datasets without annotated relations by combining relation extraction techniques and multi-relational methods. (2) To accomplish this, we show how learning similarity functions from the field of record linkage can be used to extract probabilistic relations.

2 Related work

2.1 Relational classification

One of the earliest approaches considering relations among objects was by Chakrabarti et al. [4], they proposed a probabilistic model for classification of web pages using the content of the classified page, the class labels of linked pages as well as the content of these linked pages.

More recently, researches have been done for instance on relational probability trees [15], which is a complex learning algorithm taking into account

relations among entities and using probability trees. Furthermore on relational dependency networks [16], a type of probabilistic relational model and on regression models using link distributions [11], as well as on simple models like probabilistic relational neighbor (PRN) classifier [12] or other simple methods in [2]. Macskassy and Provost [12] too, considered multiple relations, they represented the considered relations in a single graph (merging the relations and summing the weights of the common links) and performed their algorithms only once on this representation.

Fürnkranz [9] used ensemble classification in the area of hyperlink classification. Ensemble classification methods have been incorporated in order to combine the results of the predictions for each hyperlink of a target page. He showed, that this technique performed as well as a classifier considering only the content of the target page or even better. In contrast to his approach, we do not combine the prediction for each instance of a relation (link) but only for each relation type.

2.2 Record linkage

The problem of record linkage can be formulated as a prediction of an equivalence relation. Almost all models for record linkage rely on predicting the equivalence of a pair of objects. To estimate the likelihood of two objects being identical, heuristic similarity or distance measures [6] are used. Today often an adaptive method is used where multiple heuristic similarity measures over multiple attributes are combined to a single learned similarity measure [5]. Here, probabilistic classifiers can be used. Nowadays overall consistency – i.e. the transitivity axiom – is guaranteed not just by taking transitive closure, but by more sophisticated methods such as clustering [5]. For learning the models one uses a training set that is assumed to be similar to the test set for which object identities should be predicted. If the labeled subset is drawn from the problem itself, constrained clustering can help to improve the prediction quality [19].

Record linkage is related to coreference resolution [7] in natural language processing. The main differences between the methods in record linkage and coreference resolution are in the feature extraction step.

3 Relation extraction and relational classification framework

The relation extraction and relational classification framework aims at first, extracting multiple relations from unstructured data, e.g. noisy database entries and second, using them for classification. Our framework contains two main components, the relation extraction and the multi-relational classification component (see figure 1). In the following we briefly describe the components and formally specify their inputs and outputs.

Relation extraction component The relation extraction component extracts several relations from a noisy dataset like shown in table 1, e.g. *SameAuthor*,

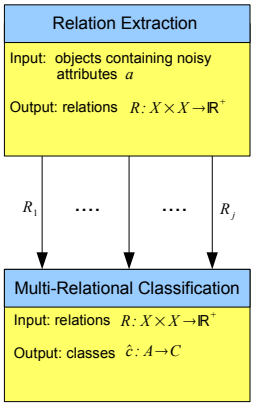

Fig. 1. Relation extraction and multi-relational classification framework

SameVenue or *SamePublisher* of scientific papers. Therefore, we use techniques from the field of record linkage (see section 4).

The input of the relation extraction component is a set of objects which are described by an id $x \in X$ and a map $a : X \to V$, from objects x to values V, the type of V can be arbitrary (e.g. strings, numbers).

After extraction we receive several relations $R : X \times X \to \mathbb{R}^+$ between objects $x \in X$. A relation R can be seen as a weighted, undirected graph $G(X, E, w)$, where X denotes the set of vertices and $E \subseteq X \times X$ a set of edges. The edges are labeled with weights $w : X \times X \to \mathbb{R}^+$ where $w(x_1, x_2) := R(x_1, x_2)$, in our case these weights will be probabilities. E.g. in the domain of scientific publications, the vertices are papers, and the edges may denote the *SameAuthor* relation, furthermore the weights encode the probability that the authors of two papers, or some of the authors, are the same.

Multi-relational classification component The multi-relational classification component uses the extracted relations for classification of objects. Each relation is used separately, i.e. the relational classification methods to be introduced in section 5.1 are applied on each relation. In order to use all the extracted relations one has to combine the results of the relational classifiers. We applied different ensemble methods in order to do that (see section 5.2).

The input of the multi-relational classification component are the aforementioned relations R represented as graphs $G(X, E)$. For a traditional classification task of objects into possible classes C, a set $X_{\text{tr}} \subseteq X$ of objects together with their classes $c : X_{\text{tr}} \to C$ is given for training and another set $X_{\text{tst}} \subseteq X$ of objects together with their classes $c : X_{\text{tst}} \to C$ is given for evaluation (with $X_{\text{tr}} \cap X_{\text{tst}} = \emptyset$). The task then is to learn an attribute-based classifier model

$$\hat{c} : A \to C \tag{1}$$

s.t. $\hat{c}(a(x))$ equals $c(x)$ for as many $x \in X_{\text{tst}}$ as possible, e.g., the misclassification rate

$$\text{err}_{X_{\text{tst}}}(\hat{c} \circ a, c) := \frac{|\{x \in X_{\text{tst}} \mid \hat{c}(a(x)) \neq c(x)\}|}{|X_{\text{tst}}|}$$

is minimal. Instead of just using attributes attached to an object, classifiers now can make use of additional information contained in the extracted relations, e.g. the attributes or classes of related objects. The neighborhood with known classes of the object x can be formalized as follows (a dot denotes a missing value)

$$N_x := \{x' \in X \mid (x, x') \in R, c(x') \neq .\}$$

To make use of relational information for classification, it must be propositionalized, i.e. converted to suitable attributes (see section 5.1). So the output of the multi-relational classification component is described by the mapping in (1), i.e. the assignment of an object to a class (represented by propositonalized relations hence, attributes A).

4 Extracting binary relations

The task of record linkage also known as duplicate detection or object identification is to detect which entities in a fuzzy database refer to the same underlying entity. This corresponds to the task of extracting an equivalence relation for relational classification. A record linkage model typically extracts one equivalence relation. In the domain of research paper typically the *SamePaper* relation is extracted [14, 5, 19]. But the same methods could be applied to learn a different target relation like *SameVenue* or *SameAuthor*. Thus, we propose to transfer methods from record linkage to our task of relation extraction. We create one record linkage model for each relation that should be extracted.

4.1 Generic Record Linkage Model

Record linkage models typically consist of three components (see figure 2). Pairwise feature extraction creates a real valued feature vector of two objects $f : X^2 \to \mathbb{R}^n$. This is done by comparing attributes of a pair of objects by several heuristic similarity functions like *TFIDF cosine similarity*, *Levenshtein* or *Jaccard distance*. A comparison of similarity functions is given in [6].

The probabilistic pairwise decision model predicts the probability that two objects are equivalent $P(x \equiv y)$. This step is also called *learning a similarity measure* [3]. The predictions are based on the pairwise features. A probabilistic classifier can be used for this step [5, 19].

At last the collective decision model utilizes the likelihood of pairwise decisions to create a consistent prediction, i.e. an equivalence relation, for the whole dataset. For this task clustering methods are used [5] including constrained clustering [19].

For scaling to large datasets, often a candidate pair generator $b : \mathcal{P}(X) \to \mathcal{P}(X^2)$ with $b(Y) \subseteq Y^2$, called *blocker*, is added. An overview of different blockers is given by Baxter et al. [1]. For problem instances with both, a large number of equivalence classes and large equivalence classes, more sophisticated scaling methods like parallelizing and object reduction are proposed [20].

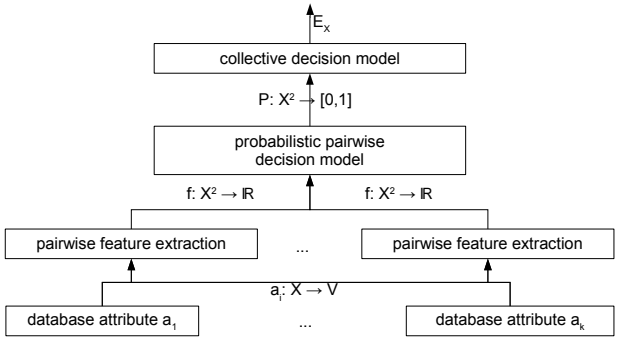

Fig. 2. Generic record linkage model

4.2 Record linkage models for relation extraction

For the task of extracting relations $R_1, \dots R_j$ from a fuzzy database with attributes $a_1, \dots, a_k$, we propose to use one record linkage model M_i for each target relation R_i. The learned similarity function P_i of each model M_i will be adapted to R_i using some training data for this specific relation. In the following we will discuss the extraction of one relation R – for j relations the proposed method has to be performed j times.

Relation extraction with a record linkage model works as follows:

1. Choose several heuristic similarity functions $f_1, \dots, f_l$ that compare two objects over an attribute. There can be multiple heuristics functions over multiple attributes. E.g. compare the *title* attribute with TFIDF cosine similarity f_1 and Levenshtein distance f_2, and compare *venue* with Overlap-Index f_3 and TFIDF cosine similarity f_4.
2. The pairwise similarities $f_1, \dots, f_l$ are used as features for a probabilistic classifier, e.g. logistic regression or probabilistic SVM. This gives an estimate $\hat{C} : \mathbb{R}^l \to [0,1]$. The classifier tries to find the underlying concepts of R in a training set $X_{tr} \subseteq X^2$ where R is known $R : X_{tr} \to [0,1]$. A training case for $(x,y) \in X_{tr}$ is a $l+1$ dimensional vector consisting of the feature vector of the heuristic similarity functions and the known target: $(f_1(x,y), \dots, f_l(x,y), R(x,y))$.
3. For prediction of an unlabeled pair $(x,y) \notin X_{tr}$, the estimate is taken: $\hat{C}(f_1(x,y), \dots, f_l(x,y)) =: \hat{R}(x,y)$
4. If the target relation is known to be an equivalence relationship, clustering of $\hat{R}$ can be performed to translate it in a binary relation – i.e. the target is in $\{0,1\}$ – and to ensure transitivity of $\hat{R}$.

5 Multi-relational classifiers

The aim of multi-relational classification is, instead of only using local attributes of an object for classification, to take into account multiple relations existing

between objects. This can be done by first, using relational classification on each relation (represented as graph) and then, applying ensemble methods on the results.

First, we will describe relational classification methods that use a single relation, followed by a brief description of how to combine multiple relations.

5.1 Relational classification using a single relation

The methods we present use the classes of related objects as predictor variables, since we assume that related objects belong to similar classes. All the methods we introduce, use collective inference.These methods are iterative procedures, which classify related instances simultaneously. Collective inference exploits relational autocorrelation, the correlation among the values of a variable of an object to the same variable of a related object. Collective inference can be formally expressed as follows:

$$P(c|x)^t = M(N_x^{(t-1)}) \tag{2}$$

We iteratively calculate the class-membership probability of an object x with a relational classification method M in the inner loop. We initialize the categories of the test instances with the prior probability. The procedure stops when a certain number of iterations is reached or the algorithm converges. The relational classification algorithm takes the neighborhood of the object x as input.

We apply different relational classification methods in the inner loop of the collective inference procedure. The *probabilistic relational classifier (PRN)* introduced by Macskassy and Provost [12] is one of them, it calculates the class-membership probability of an object $x \in X$ and a class $c \in C$ as the weighted arithmetic mean of the class-membership probabilities of the neighbors x', whereas N_x is the set of neighbors of x:

$$P(c|x) = \frac{1}{Z} \cdot \sum_{x' \in N_x} w(x, x') \cdot P(c|x') \tag{3}$$

where Z is a normalizing constant and $w(x, x')$ is the weight of the edge between the objects x and x'. We have introduced extensions of this algorithm in [17, 18], an interesting extension is to take into account not only direct neighbors but also indirectly linked objects by longer paths. This densifies the usually very sparse graphs. We call the algorithm that considers neighbors on a path of length two *PRN2Hop*.

Furthermore we want to present a *relational classification* methods, which learn a model on the training instances based on an aggregation of classes of the neighborhood of an object.

We have analyzed several aggregation functions, which map the set-valued attributes (classes or probability distributions of the neighbors of a particular object) to an aggregated value in order to use standard classification methods.

First, we use *weighted average* as aggregation function, which calculates the attribute value by building the average of the weighted probabilities of the neighbors of an instance (which is equivalent to *PRN* in equation (3)). After calculat-

ing the attributes for the training instances, we learn a model on these attributes with a machine learning method.

The second aggregation function we apply [18] is the *Indirect RVS Score*, it is based on the relational vector space model [2] and corresponds to the Cosine Similarity measure.

Moreover we use a Naive Bayes Classifier based on the approach of Chakrabarti et al. [4] and Macskassy and Provost [13], with a different *collective inference* algorithm. We call this algorithm *Weighted Naive Bayes*. In contrast to the classical Naive Bayes classifier, this approach considers the weights of the edges between an object and its neighbors. Here the attributes of an object are the classes of its neighbors.

The class-membership probability for an object x and class c is computed according to Bayes rule as follows:

$$P(c|x) = \frac{P(N_x|c) \cdot P(c)}{P(N_x)} \tag{4}$$

While $P(N_x|c)$ can be computed as:

$$P(N_x|c) = \frac{1}{Z} \cdot \prod_{x' \in N_x} P(c(x')|c)^{w(x,x')} \tag{5}$$

5.2 Combining multiple relations

Since usually several relations can be found implicitly in the data, they should be extracted by the relation extraction component and used for classification. We showed in the former subsection how to classify objects according to a relation. The relational classification has to be applied on each single relation. Now we need to combine these results. We use *ensemble classification* methods for this task. We have decided to use *voting*, a simple *ensemble classification* technique and *stacking*, a more complex learning method. *Ensemble classification* may lead to significant improvement on classification accuracy. This is because uncorrelated errors coming from individual classifiers are removed by the combination of different classifiers [8]. *Ensemble classification* reduces variance and bias, moreover, the combination of several classifiers may learn a more expressive concept compared to a single classifier.

Using unweighed voting means, we build the arithmetic mean over the probability distributions of each relational classifier K_l. The second *ensemble classification* method we have used is *stacking* [21]. This method uses a meta-classifier to learn the probability distributions of the individual relational classifiers and predicts the probability distribution of the combination of these classifiers. We perform the individual classifiers K_l using k-fold cross validation first, these classifiers are called *level-0* classifiers. Then, we need to set up new objects, so called *level-1* objects, which are built of the class-membership probabilities achieved by the individual classifiers and the original label c:

$$x_{new} = (P(c_1|x)_1, \ldots, P(c_n|x)_1, \ldots, P(c_1|x)_L, \ldots, P(c_n|x)_L, c) \tag{6}$$

Then, a meta-classifier (we have used Logistic Regression) is learned on the new *level-1* training instances and the model is applied to the *level-1* test instances using k-fold cross validation.

The result of combining multiple relations is a mapping $\hat{c} : A \rightarrow C$ (see (1)), i.e. the attributes which describe an object are mapped to a class.

6 Evaluation

In this paper we provide experimental results on data from the domain of scientific publications for each of the two tasks, i.e. multi-relational classification an extracting relations. The first evaluation shows that if relations are given, relational classifiers work well and can outperform a SVM using no relations but only local attributes. Secondly we show how a relation can successfully be extracted from a noisy scientific publication dataset.

6.1 Multi-Relational Classifiers

We evaluated our multi-relational classifiers on the *CompuScience*[1] dataset, it is from the domain of scientific papers within Computer Science. We have used 147 571 papers with 117 936 unique authors, 9 914 reviewers and 2 833 journals. These relations where already extracted and present in a structured format. For each publication, a title and an abstract is available, but citations are not. Thus, we will use the relations *SameAuthor*, *SameReviewer* and *SameJournal*. A characteristic of this dataset is, that each publication can be assigned to more than one category out of the 77 categories (topics of publications) that are available, we are performing multi-label classification.

We have performed an experiment, using three-fold cross validation and have used F-Measure as an evaluation measure. In this experiment the aim was to find out how the multi-relational classifiers perform compared to local classifiers (e.g. SVM) using only local features.

We combined first, all relations using *stacking* and then combined the result with the outcome of a text classifier using *stacking* as well. As a local classifier, we have chosen a Support Vector Machine using the words of the abstract and title of a paper as features of the SVM. All our methods presented in 5 are prefixed with "E", since all of them are using ensemble methods. Figure 3 shows the F-Measure of some algorithms, which we have described in section 5 (ELog Reg WeightedAvg denotes Logistic Regression using *weighted average* as an aggregation function). One can see, that the ensembles of *relational classifiers* and the text classifier always achieves higher F-Measure values than the text classifier alone (using only local features). The result of *EPRN* fused with the results of the text classifier are actually significantly[2] better than the results of the SVM.

Our approach performs better because of the relational collective inference methods which are exploiting the relational autocorrelation and are propagating

[1] This corpora is extracted from a database produced by FIZ (Fachinformationszentrum) Karlsruhe for the information service io-port.net.

[2] Significance level is 0.05. The null hypothesis is, that the result of an algorithm is not higher than the other results. We used t-test.

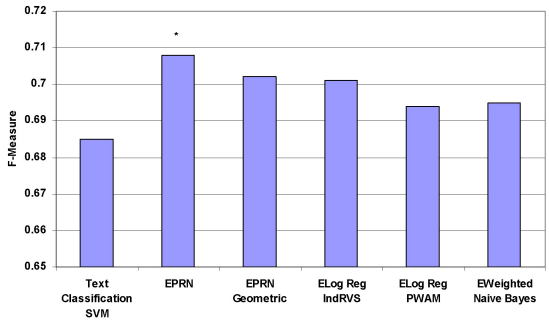

Fig. 3. F-Measure of multi-relational classifiers compared to SVM on *CompuScience*

the information in the graph. Additionally it gains from the ensemble methods which may reduce variance and bias of the single classifiers.

Experiments on the Cora dataset have also been performed [17], they have shown that using the multi-relational methods described in section 5 outperform complex relational methods RPT, RDN and RBC as well as PRN, the approach of Macskassy and Provost [12], who are not using ensemble methods but merge the relations and sum up the weights for these relations.

6.2 Extracting Relations

In this experiment we show how the *SamePaper* can be extracted from a noisy citation database. This experiment is performed on the *Cora* deduplication dataset[3] [5] containing 1 295 citations. The dataset contains citations to identical paper and is highly noisy in all attribute values. The task for our experiment is to find identical papers, i.e. the equivalence relation *SamePaper*.

The model is setup as follows: For pairwise feature extraction it uses TFIDF cosine similarity, Levenshtein string distance and Jaccard distance between every single attribute, namely *author*, *volume*, *title*, *institution*, *venue*, *address*, *publisher*, *editor*, *note*, *month*, *year* and *pages*. For learning the similarity measures a C-SVM from libSVM[4] is used. A constrained hierarchical clustering algorithm [19] ensures the axioms of an equivalence relation. A canopy blocker [14] reduces the candidate pairs. The training set $X_{tr} := Y^2$ is generated by randomly labeling 25% of the objects $X_{tr} \subset X$ with the true class value. We report the pairwise F1-Measure both over the unlabeled 75% of the dataset $X_{tst} := (X \setminus Y)^2$ and the full data set X. All experiments were repeated 4 times.

Table 2 shows the quality of three linkage methods. As you can see, the proposed model with average linkage successfully finds the true relation *SamePaper*

[3] Please note that this dataset is different from the relational *Cora* dataset where noise has been removed, deduplication is already done and which contains other papers. But the domain and even the application are the same.

[4] http://www.csie.ntu.edu.tw/ cjlin/libsvm

Evaluation set	single linkage	complete linkage	average linkage
X_{tst}	0.90	0.74	**0.92**
X	0.92	0.71	**0.93**

Table 2. Pairwise F1-Measure for finding the relation *SamePaper* in the *Cora* deduplication dataset.

with a F-Measure quality of 0.92 on $(X \setminus Y)^2$ and with a quality of 0.93 on the full dataset X^2.

7 Conclusion and Future Work

In this paper we have described a framework that allows to apply relational classifiers to datasets without explicitly stated relations. The framework consists of two steps: (1) relation extraction and (2) multi-relational classification. For relation extraction we suggested to use methods from the field of record linkage. For each target relation a linkage model is learned. We proposed to use an ensemble of relational classifiers hence, each relational classifier uses a relation. Our evaluation indicates that each of the individual tasks, i.e. relation extraction and relational classification, can be solved efficiently.

In future work we would like to evaluate the quality of relational classifiers based on multiple automatically extracted relations. We also would like to evaluate the classifier's quality depending on the quality of the extraction.

8 Acknowledgements

This work was funded by the X-Media project (www.x-media-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-026978.

References

1. Baxter, R., Christen, P., Churches, T.: A comparison of fast blocking methods for record linkage. Proceedings of the 2003 ACM SIGKDD Workshop on Data Cleaning, Record Linkage, and Object Consolidation, 25–27 (2003)
2. Bernstein, A., Clearwater, S., Provost, F.: The Relational Vector-space Model and Industry Classification. Proceedings of IJCAI Workshop on Statistical Models from Relational Data, 8–18 (2003)
3. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2003)
4. Chakrabarti, S., Dom, B., Indyk, P.: Enhanced Hypertext Categorization Using Hyperlinks. Proceedings of ACM SIGMOD, 307–318 (1998)
5. Cohen, W.W., Richman, J.: Learning to match and cluster large high-dimensional data sets for data integration. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 475–480 (2002)

6. Cohen, W.W., Ravikumar, P., Fienberg S.E.: A comparison of string distance metrics for name-matching tasks. Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web, 73–78 (2003)
7. Culotta, A., Wick, M., Hall, R., McCallum, A.: First-Order Probabilistic Models for Coreference Resolution. Proceedings of NAACL HLT (2007)
8. Dietterich, T.,G.: Machine Learning Research: Four Current Directions. AI Magazine 18(4), 97-136 (1997)
9. Fürnkranz, J.: Hyperlink Ensembles: A Case Study in Hypertext Classification. Information Fusion 3, (2002)
10. Jensen, D., Neville, J., Gallagher, B.: Why Collective Inference Improves Relational Classification. Proceedings of the 10th ACM SIGKDD, (2004)
11. Lu, Q., Getoor, L.: Link-based Text Classification. Proceedings of IJCAI Workshop on Text Mining and Link Analysis, (2003)
12. Macskassy, A.,S., Provost, F.: A Simple Relational Classifier. Proceedings of the Multi-relational Data Mining Workshop ACM SIGKDD, (2003)
13. Macskassy, A.,S., Provost, F.: Classification in Networked Data: A Toolkit and a Univariate Case Stud. CeDER Working Paper, Stern School of Business, New York University, (2004)
14. McCallum, A.K., Nigam, K., Ungar, L.: Efficient clustering of high-dimensional data sets with application to reference matching. Proceedings of the 6th International Conference On Knowledge Discovery and Data Mining, 169–178 (2000)
15. Neville, J., Jensen, D., Friedland, L., Hay, M.: Learning Relational Probability Trees. Proceedings of SIGKDD, (2003)
16. Neville, J., Jensen, D.: Collective Classification with Relational Dependency Networks. Proceedings of the second Workshop on Multi-relational Data Mining KDD, (2003)
17. Preisach C., Schmidt-Thieme L.: Relational Ensemble Classification. Proceedings of the 6th IEEE International Conference on Data Mining, 499–509 (2006)
18. Preisach C., Schmidt-Thieme L.: Ensembles of Relatinal Classifiers. Knowledge and Information Systems. 249–272 (2008)
19. Rendle, S., Schmidt-Thieme, L.: Object identification with constraints. Proceedings of the 6th IEEE International Conference on Data Mining (2006)
20. Rendle, S., Schmidt-Thieme, L.: Scaling record linkage to non-uniform distributed class sizes. Proceedings of the 12th Pacific-Asia Conference on Knowledge Discovery and Data Mining (2008)
21. Wolpert, D.,H.: Stacked Generalization. Neural Networks 5, Pergamon Press, 214-259 (1992)