# Relational Constraints for Metric Learning on Relational Data

Jiajun Pan, Hoel Le Capitaine, Philippe Leray

# Relational Constraints for Metric Learning on Relational Data

**Jiajun Pan, Hoel Le Capitaine & Philippe Leray**
LS2N, UMR CNRS 6004
University of Nantes
44300 Nantes Cedex, France
{jiajun.pan, hoel.lecapitaine, philippe.leray}@ls2n.fr



**Abstract**

Most of metric learning approaches are dedicated to be applied on data described by feature vectors, with some notable exceptions such as times series, trees or graphs. The objective of this paper is to propose a metric learning algorithm that specifically considers relational data. The proposed approach can take benefit from both the topological structure of the data and supervised labels. For selecting relative constraints representing the relational information, we introduce a link-strength function that measures the strength of relationship links between entities by the side-information of their common parents. We show the performance of the proposed method with two different classical metric learning algorithms, which are ITML (Information Theoretic Metric Learning) and LSML (Least Squares Metric Learning), and test on several real-world datasets. Experimental results show that using relational information improves the quality of the learned metric.

## Introduction

Sample similarity measurement lies at the heart of many classification and clustering methods in pattern recognition and machine learning. For instance, in classification, the k-Nearest Neighbor classifier uses a metric to identify the nearest neighbors; in clustering algorithms, k-means rely on distance measurements between data points; in information retrieval, documents are often ranked according to their relevance to a given query based on similarity scores. The performance of these algorithms rely on the quality of the metric. The conventionally used Euclidean distance cannot give a convenient dissimilarity in many cases, due to the distribution of the data (see (Tenenbaum, De Silva, and Langford 2000)). Thus, it calls a great need for appropriate ways to measure the distance or similarity between observations in learning algorithms.

Metric learning has now been used for more than a decade to deal with this problem, and can be seen a feature/representation learning allowing the use of Euclidean distances later on. The vast majority of metric learning approaches are dedicated to be applied on data described by feature vectors, where the objective is generally to learn a matrix $M$ that is used for the Mahalanobis distance

$$D^2_M(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T M (\mathbf{x} - \mathbf{y}),$$

where $\mathbf{x}$ and $\mathbf{y}$ are $d$-dimensional vectors and $M$ is a positive semi-definite matrix. Using such a metric is equivalent to perform a linear projection of the data by the matrix decomposition $M = L^T L$, where $L \in \mathbb{R}^{k \times d}$, and then use the conventional Euclidean distance in this new feature space.

The interested reader can refer to reviews in (Kulis and others 2013) and (Bellet, Habrard, and Sebban 2015).

In this model, $M$ contains the learned parameters and is learned mostly from supervised information. Most of the approaches make use of label constraints, which means that the constraints are generated by the target labels or other supervised information. Such a distance is perfectly adapted for flat or iid data, but obviously fails to take into account complex and/or (semi-)structured, non-iid data without considering the structured information.

There are some notable exceptions such as times series in (Garreau et al. 2014) (through dynamic time warping methods) and trees or graphs in (Bellet et al. 2016) (by using an edit distance) or networks, proposed in (Shaw, Huang, and Jebara 2011). Relational databases are increasingly used in almost all applications. A lot of real world datasets present aspects of multi-relation between the observations. For instance, social service networks, Wikipedia network, molecular biology classification and so forth. These databases are organized based on a relational model of data which contains entity tables and association tables between entity tables. Using this data in machine learning is now under consideration for years (Getoor 2007), but to the best of our knowledge, no attention on metric learning has been paid for such data. Naturally, one can use traditional metric learning algorithms for individual entities, but at the price of losing rich information coming from the relational structure of the data. Taking good use of associations between entities can help to improve metric performance.

The goal of this paper is to propose the use of both relational information and supervised information in constraints selection for metric learning. Such a definition allows to build rich models, which can eventually be used for domain adaptation, transfer learning, feature learning and data visualization with both flat and multi-relational data. In particular, we propose a solution that is able to incorporate rela-

tional information within metric learning, and then illustrate its benefit compared to traditional flat approaches.

Note that the proposed approach starts from (hyper)graph data, where as approaches as in (Dhillon, Talukdar, and Crammer 2012). Instead of using label constraints, the authors used graph based constraints which enforces the distance between the unlinked node and target node to be bigger than the distance between the k-farthest linked node and target node. We propose a more general model denoted as link-strength constraints which is generated by a link-strength function measuring the similarity of nodes by the side information of the relationships between them. The proposed link-strength function gives the possibility of using similarity learning to encode the relational information into the constraints for metric learning algorithm.

This paper is organized as follows. We recall basic definitions related to metric learning, as well as related works, in section . In section , we present our approach, which is validated by experimental comparisons on real-world datasets in section . Final comments and perspectives are drawn in section .

## Related Works

### Metric Learning and Relational Learning

Since its seminal paper in 2003 (Xing et al. 2003), there have been many propositions dealing with learning metrics. In (Kulis and others 2013), one can find a number of different metric learning algorithms. Most of the propositions of metric learning rely on a new metric $D'(\mathbf{x}, \mathbf{y})$ as $D(f(\mathbf{x}), f(\mathbf{y}))$ with a mapping function $f$. The metric learning then simply consists in learning the projection $f$ by using constraints on the (dis)similarity on observations. With constraints generated from given information, metric learning approaches generally consider the generic loss function written as

$$L(M) = \sum_{(i,j,k)\in\mathcal{C}} \ell_M(i,j,k) + \lambda r(M),$$

where $\ell_M(i,j,k)$ is the encoded loss from pairs of entity nodes (i,j) in set $\mathcal{S} = \{(i,j)\}$ of similar observations and pairs of entity nodes (i,k) in set $\mathcal{D} = \{(i,k)\}$ of the dissimilar observations. $r(M)$ is a regularization term on the matrix $M$ (e.g Frobenius norm, trace-norm). The loss $\mathcal{C}\ell_M(i,j,k)$ is then generally written as the hinge loss functions $max(0, D_M(\mathbf{x}_i, \mathbf{x}_j) - u)$ and $max(0, l - D_M(\mathbf{x}_i, \mathbf{x}_k))$ with threshold parameters $u$ and $l$ or $max(0, m + D_M(\mathbf{x}_i, \mathbf{x}_j) - D_M(\mathbf{x}_i, \mathbf{x}_k))$ with margin parameter $m$.

Traditionally, in metric learning, one uses label constraints in order to select similar and dissimilar constraint sets by the target label or other supervised information. In this case, the pair $(i,j)$ contains the nodes in same class, while the labels of nodes $(i,k)$ are different.

Relational learning deals with learning models for which data consists in a generally complex relational structure. As a difference with flat datasets, the main learning tasks pay more attention on supervised information from the relations and knowledge from the topology of the relational graph. Collective classification is classification of related entities that may share identical classes (Sen et al. 2008) from the relationship information.

Note that, in (Dumančić and Blockeel 2017), the authors propose an analysis of the meaning of the latent space learned by a deep learning algorithm on relational datasets, mainly because of the black box problem of deep approaches. The same analysis can be conducted for the meaning of metric learning, which can also be treated as mapping original feature space to a latent space. This paper mainly focuses on the usefulness of latent space and the redundancy of the latent features. They show the good performance of using unsupervised relational information for a classifier in a latent space. We specifically focus on using relational information for collective classification.

For collective classification, there are several approaches to learn metric with relational information. In (Kramer, Lavrač, and Flach 2001), they use graph relational information with propositionalization, transfer the relational representation of a learning problem into a propositional representation. Another approach uses metric learning on graphs for domain adaptation (Dhillon, Talukdar, and Crammer 2012). To this aim, they propose an iterative learning algorithm on the graph. From the resource domain, the nodes with labels, they learn a new metric and apply it on related target domain, the nodes without labels. Then, the graph is updated depending on the learned distance, and constraints with low entropy instances are selected for next iteration.

Those approaches do not consider the different modalities of the observations (features and relations). In many cases, the structure of the data does not allow to directly measure distances as if the observations were belonging to an Euclidean space. In particular, complex and structured data needs to be processed in a different way than usual tabular data. Such complex data includes times series, videos, graphs, relational data, without exhaustivity.

Some metric learning algorithms consider relational data as heterogeneous networks for each different relationships. For instance, in (Zhai, Peng, and Xiao 2013), they propose a heterogeneous metric learning algorithm, which integrates the structure of different graphs into a joint graph regularization. They use two mapping function for the feature space of the object entities and subject entities in one relation, and then introduce a joint graph regularization for iterative optimize the loss function. In (Dong, Chawla, and Swami 2017), they start from the same principle but use meta-path-based random walks to incorporate the heterogeneous network structures into skip-gram vectors for dealing with the relational graph. Those algorithms use joint regularization for different entities in heterogeneous networks, with good performance on considering the structure information in relational dataset. However, without considering the side information in the relational links. It processes the relational variables the same way as the entities and subject to their algorithm, but it ignores the differences between entity tables and association tables. Our proposed method includes the value of different variables on the relationship in the datasets and distinguish them with entities.

In this paper, we consider the case of relational data, where several tabular datasets (entities) are linked together

through associations. The basic principle of our approach is to use relational links between entities when setting the constraints of the metric learning algorithm. Consequently, this approach can be used in any constraint-selection based metric learning algorithm.

### Metric Learning with Relational Constraints

Recently, some metric learning approaches have been focusing on graph data. In this context, the structural information consists in the presence or absence of links between nodes of the graph. Constructing the set of similar nodes and dissimilar nodes then just uses the adjacency matrix of the graph. For example, the simplest relative link constraints is $D_M(\mathbf{x}_i, \mathbf{x}_j) \leq D_M(\mathbf{x}_i, \mathbf{x}_k) + m, (i,j) \in \mathcal{S}, (i,k) \in \mathcal{D}$ with the adjacent matrix $A_{ij} = 1, A_{ik} = 0, \forall (i,j) \in \mathcal{S}, \forall (i,k) \in \mathcal{D}$, which only check the relative distance between the connected and disconnected relational links. In the sequel, this method is termed as relative link constraint.

In (Shaw, Huang, and Jebara 2011), they formulate the metric in a preserving embedding structure and learn from linear constraints with the graph topology. They proposed nearest neighbor graphs and maximum weight subgraphs, which are two ways for generating the supervised constraints with the relational information in graphs. The nearest neighbour constraints is denoted as $D_M^2(\mathbf{x}_i, \mathbf{x}_j) > (1 - A_{ij}) \max_l (A_{il} D_M^2(\mathbf{x}_i, \mathbf{x}_l)), \forall i, j$ which aims to constraint the disconnected node to be more far from the target node than its farthest connected node (neighbour).

However, this approach does not take into account that there are a lot of information in the valued link with numerical or categorical variables. Furthermore, target labels of nodes are not used in this model. We propose to define new relative link constraints from to the link-strength constraints, with a link-strength function to measure the importance of the relational links between entities, as well as using the supervised information obtained from the labels.

## On selecting constraints with link-strength function

In this paper, our objective is to propose a new approach of metric learning considering enforcing constraints with both the relational links and label information. The basic statement behind our proposal is that we consider that two connected individuals are more similar than two unconnected individuals. We then evaluate the amount of link similarity between individuals by considering their common parents in the graph. If, furthermore, labels are available for individuals, this can be incorporated into the learning algorithm. Consequently, our approach can be both supervised or unsupervised, depending on the availability of the information in the data.

We extend the relative link constraints by separating $\mathcal{S}$ and $\mathcal{D}$ with a dedicated link-strength function. The proposed constraints enhance the classical metric learning algorithms using relative constraints, such as ITML (Davis et al. 2007) and LSML (Liu et al. 2012).

A link-strength function $LS(\mathbf{x}_i, \mathbf{x}_j | r)$ is a function with the input is the relational information between the entities nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ and the output is a real value to measure the "strength" or the similarity of the two nodes. The symbol $r$ represents the relationship information of the entities nodes $\mathbf{x} \in R^{n \times d}$, where $n$ is the number of instances and $d$ the number of node attributes.

There are many ways to encode the relational information for this link-strength function, and we choose side-information of the common parents between the input nodes.

A relational schema $R = R_e \cup R_r$ contains a set of relational information where $R_e$ denotes the set of groups and types between entities in same tables and $R_r$ denotes the set of reference links between different tables. For a relation subset $r_k \subseteq R_r$ including all references between two entity tables, for example like the reference Ratings between User and Movie as shown in Figure 1, we consider it as a many-to-many relationship. Let $P_{ij}$ be the set of common parents of $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\ell_{ij} = |P_{ij}|$, the number of common parents.

In Figure 2, we give a subsample of a bipartite relational graph, along with an example of common parents.

Naturally, the similar node would get the similar references from the same parents node. Consequently, we consider that the link-strength depends on the side-information of the common parents which are the values of the references.

Additionally, the references can be quantified by $\alpha$ numerical variables $v$ and $\beta$ categorical variables $v^*$, see Figure 3.

Given a relation $r_k$, we propose to defined the link-strength function as:

$$LS(\mathbf{x}_i, \mathbf{x}_j | r_k) = LS(\mathbf{x}_i, \mathbf{x}_j | P_{ij}) = \sum_{h=1}^{\ell_{ij}} (\gamma \cdot w(h,i,j) + (1-\gamma) \cdot z(h,i,j)) \quad (1)$$

where

$$w(h,i,j) = \sum_{m=1}^{\alpha} \exp\left(-\left|v_m(p_h, \mathbf{x}_i) - v_m(p_h, \mathbf{x}_j)\right|\right),$$

and

$$z(h,i,j) = \sum_{m=1}^{\beta} \left(v_m^*(p_h, \mathbf{x}_i) \circ v_m^*(p_h, \mathbf{x}_j)\right),$$

in which $x \circ y = 1$ iff $x = y$, and 0 otherwise, and $p_h$ is the $h$-th parent node in the set $P_{ij}$. Note that numerical association attributes $v$ are normalized in the unit interval prior to link strength computation. Note also that we restrict to unit-length slot chains, i.e. the length of the sequence of foreign key references is equal to 1. Then, we select the strongest links as similarity constraints, and the weakest links as dissimilarity constraints. The corresponding algorithm is given in Algorithm 1. Remark that if two entities $\mathbf{x}_i$ and $\mathbf{x}_j$ do not have common parents, their link strength is zero, and therefore considered as dissimilar.

In this paper, we mainly focus testing the proposed algorithm on one $R_r$ relation dataset. However, this link-strength function could be easily extended to multi-relational datasets by summing the link-strength for each relations. It could also

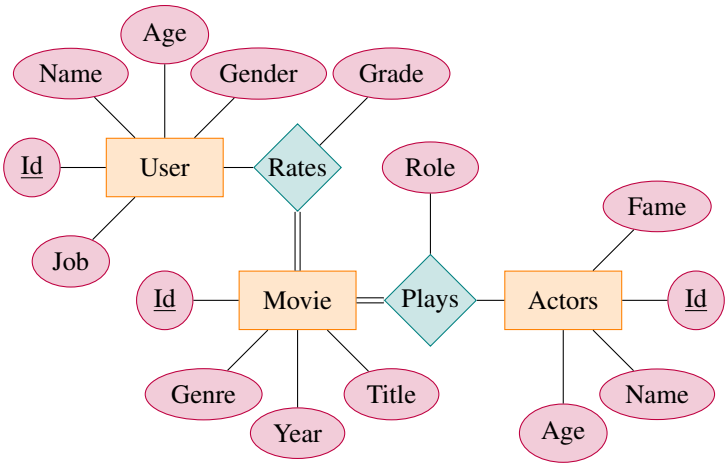


Figure 1: Typical example of an Entity-Association relational model

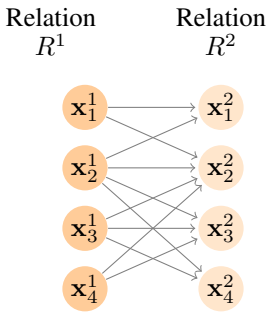


Figure 2: Bipartite relational graph for a *many-to-many* relationship table. The common parents of $\{\mathbf{x}_2^2, \mathbf{x}_3^2\}$ is the set of entities $P_{\mathbf{x}_2^2,\mathbf{x}_3^2} = \{\mathbf{x}_2^1, \mathbf{x}_3^1, \mathbf{x}_4^1\}$

**Algorithm 1** Relational constraints learning

**Require:** $N_{max}$ : number of desired constraints

$p \leftarrow 1$ ; $\mathcal{S} \leftarrow \emptyset$ ; $\mathcal{D} \leftarrow \emptyset$
**while** $p \leq N_{max}$ **do**
  $X_p \leftarrow (\mathbf{x}_i, \mathbf{x}_j)$ random pair generation
  compute link strength $LS_p$ of $X_p$ using Equation ()
  $p \leftarrow p + 1$
**end while**
**while** $|LS| > 0$ **do**
  $\mathcal{S} \leftarrow \mathcal{S} \cup X_{\text{argmax}\{LS\}}$
  $\mathcal{D} \leftarrow \mathcal{D} \cup X_{\text{argmin}\{LS\}}$
  $LS \leftarrow LS \setminus \{\mathcal{S} \cup \mathcal{D}\}$
**end while**
**return** $\{\mathcal{S}, \mathcal{D}\}$

be extended from the reference relation $R_r$ to $R_e$, that consider the group structures as every edge between nodes is the side-information of common parents but as binary value. In that case, the link-strength function would consider the additional term

$$\sum_k^{\ell_{ij}^k} v_m P_k(i,j),$$

where $P_k(i,j)$ is the parent adjacency matrix of the relation $k$ in the group structure defined as

$$P_k(i,j) = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ have common parents in relation } r_k \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

and $\ell_{ij}^k$ is the number of common parents of $\mathbf{x}_i$ and $\mathbf{x}_j$ in the relation $r_k$.

With the link-strength function, we select the relative constraints set $\mathcal{C} = \{(i,j,k) : LS(\mathbf{x}_i, \mathbf{x}_j) \geq LS(\mathbf{x}_i, \mathbf{x}_k)\}$ and use the constraints on two different classical metric learning algorithms, ITML and LSML.

## Experiments

### Datasets and Tasks

We conduct experiments to compare the performance of the constraints generated by link-strength function and the constraints generated by the label information. To compare fairly, we set the amount of constraints generated by different ways are same and the formula are both in the relative distance constraint. Basically, any relational data for which classification is needed can be tackled by our proposition.

We consider several real-world relational datasets which contains feature information for mapping with the learned metric, the target label information, and the relational information (existing links or valued links). We learn the metric of one entity table for predicting target label, so for the same dataset, we can learn different metrics for different tasks. Here are the descriptions of the chosen datasets and tasks:

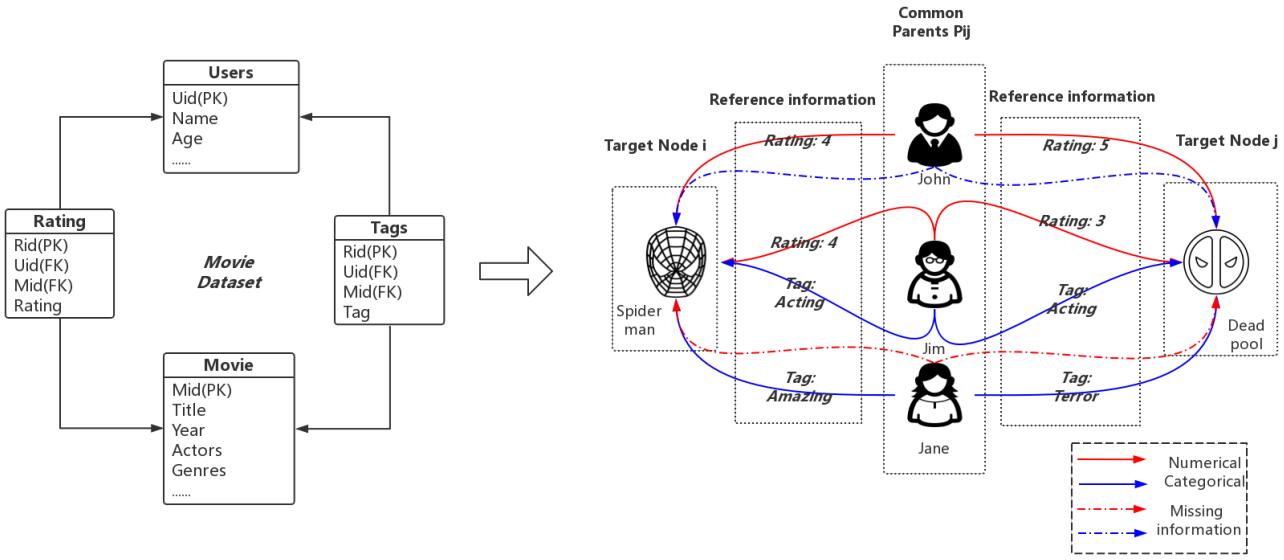


Figure 3: The side information of references from common parents $P_{ij}$ to node $\mathbf{x}_i$ and node $\mathbf{x}_j$, in the movie dataset.

- Movie: MovieLens dataset (Harper and Konstan 2016) is a classical relational dataset which is widely used in many related papers. It consists of a relational table which has 100,000 ratings (1-5) from 943 users on 1682 movies; a movie entity table with feature information about the movies; and a user entity table with id, age, gender, occupation, and other feature information on users. Each user has rated at least 20 movies so the relational supervised information is quite dense. We define two tasks on this dataset:
  - Movie-item: We select the movies table as the entity table to learn the metric on. We choose the most popular genre as the target label and use the release date and other genres as the attributes.
  - Movie-year: We select the users table as the entity table to learn the metric on. The age of users is discretized into 5 bins as the target label and the other feature information are the attributes.
- BookCX: We also consider the book-crossing database (Ziegler et al. 2005). We select a randomly sampled subset BookCX from the data. This subset contains 2,400 users giving 5,000 ratings (1-10) on 10,000 books. For this dataset, we use the bag-of-words model to encode the text information from the titles, the authors and the publishers into binary attributes.
  - BookCX-year: We consider the public year segmented into 5 bins as the target label and the bag-of-word of text information as the attributes.
  - BookCX-word: We apply PCA (Principal Component Analysis) on the bag-of-word of text information and limit the number of dimensions to 12. Then we randomly choose one of the processed dimension and segment it into 5 bins as the target label. The other features are considered as attributes.
- Citelike: There are two versions of citelike dataset, Citelike-t and Citelike-a, both used in the paper (Wang, Chen, and Li 2013). They were collected from CiteULike and Google Scholar. CiteULike allows users to create their own collections of articles. There are abstracts, titles, and tags for each article. They manually select hundreds of seed tags and collect all the articles with at least one of these tags. They also crawl the citations between the articles from Google Scholar. Note that the final number of tags associated with all the collected articles is far more than the number of seed tags. To reduce the computation complexity, we apply PCA on the large and sparse tag feature space and limit the number of dimensions to 12. We randomly choose one of the processed features and segment it into 5 bins as the target label. Note that the sampling of Citelike-t and Citelike-a are independent and the density of the links is different.
- Mondial: This dataset (May 1999) contains the relational version of the geographical Web data sources which is composed of CIA World Factbook, a predecessor of Global Statistics, additional textual sources for coordinates, the International Atlas and some geographical data of the Karlsruhe TERRA database. We chose part of the entity table City to learn and use the table Countries as the parent table. The population is segmented into 5 bins as the target label

## Result and Analysis

For all the used datasets, the balance parameter between association attributes in the link-strength function is set to $\gamma = \frac{\alpha}{\alpha+\beta}$ in order to adapt to different situations of the

datasets, where $\alpha$ is the number of numerical variables and $\beta$ is the number of categorical variables as mentioned before.

The evaluation of the proposition is done by comparing the effect of learned metric with k-nearest-neighbour classification. For the set of k-nn classification, we use $k$ equal to 5 and score the performance with accuracy rate via randomly shuffled 3-fold cross validation. Note that we tried different values for $k$ (in particular 3, 5, 7 and 9), and the results were consistent with the results reported here for $k = 5$ on most datasets.

For each experiment, the number of constraints varies from 100 to 500, and we give the average value of each sets as the final result. All the experiments were run on a 3.1Ghz Intel Core i5 processor, with 16 Go 1867 MHz DDR3, and the code will be published for research reproducibility. We also give results obtained without learning a metric, i.e. using an Euclidean distance for k-nn algorithm (`Euc`). Results are given in Tables 1 and 2.

In Tables 1 and 2, `Lab` indicates the result obtained using only the constraints generated from label, `Rel` shows the result obtained by the constraints generated from the relative link constraints, i.e. using the adjacency matrix $A$ of the graph. `Pro` gives the performance of our proposition based on link-strength constraints and `Both` shows the best result with both label constraints and the link-strength constraints while the proportion of them are appropriated. The Table 3 and 4 show the results with different set of proportion of label constraints and the link-strength constraints. Proportion equal to 1 corresponds to the situation of using only labels, and a proportion of 0 corresponds to the fact of using only link-strength based constraints. As can be seen in the Table, results tend to be better when using mostly link-strength constraints.

As can be seen, except on Movie-item task and BookCX-year task, comparing with constraints generated only from labels, the link-strength constraints lead to a great improvement of accuracy. On most datasets, the link-strength constraints shows better performance than the relative link constraints, except the BookCX-word task with ITML. For both labels and relational information, it provides a better accuracy than the constraints obtained from labels and similar to the constraints generated with link-strength function. Considering the different number of references in these datasets, for example 100, 000 references for 943 entities for Movie-user and 5,000 references for 10,000 entities for BookCX-year, we speculate that density or sparsity of the references leads to the deviation of results.

## Conclusion

In this paper, we propose a simple, yet effective, way of learning a metric dedicated to (multi-)relational data. This work on relational metric learning clearly shows the benefit, in terms of accuracy, of considering relational information between entities instead of the sole consideration of labels. As a first perspective, we plan to consider other way of computing link strength, that may be inspired from graph analysis techniques, e.g. connection strength metric, length of the shortest path, value of the maximum network flow between nodes. In particular, we want to consider slot chains (i.e. sequences of foreign key references) which are longer than 1. We also plan to define a dedicated relational metric that could be learned directly, instead of setting relational constraints on standard metric learning algorithms.

## References


[Bellet et al. 2016] Bellet, A.; Bernabeu, J. F.; Habrard, A.; and Sebban, M. 2016. Learning discriminative tree edit similarities for linear classification application to melody recognition. *Neurocomputing* 214:155–161.

[Bellet, Habrard, and Sebban 2015] Bellet, A.; Habrard, A.; and Sebban, M. 2015. Metric learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 9(1):1–151.

[Davis et al. 2007] Davis, J. V.; Kulis, B.; Jain, P.; Sra, S.; and Dhillon, I. S. 2007. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, 209–216. ACM.

[Dhillon, Talukdar, and Crammer 2012] Dhillon, P. S.; Talukdar, P.; and Crammer, K. 2012. Metric learning for graph-based domain adaptation. In *International Conference on Computational Linguistics*, 255–264.

[Dong, Chawla, and Swami 2017] Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 135–144. ACM.

[Dumančić and Blockeel 2017] Dumančić, S., and Blockeel, H. 2017. Demystifying relational latent representations. *arXiv preprint arXiv:1705.05785*.

[Garreau et al. 2014] Garreau, D.; Lajugie, R.; Arlot, S.; and Bach, F. 2014. Metric learning for temporal sequence alignment. In *Advances in Neural Information Processing Systems*, 1817–1825.

[Getoor 2007] Getoor, L. 2007. *Introduction to statistical relational learning*. MIT press.

[Harper and Konstan 2016] Harper, F. M., and Konstan, J. A. 2016. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5(4):19.

[Kramer, Lavrač, and Flach 2001] Kramer, S.; Lavrač, N.; and Flach, P. 2001. Propositionalization approaches to relational data mining. In *Relational data mining*. Springer. 262–291.

[Kulis and others 2013] Kulis, B., et al. 2013. Metric learning: A survey. *Foundations and Trends® in Machine Learning* 5(4):287–364.

[Liu et al. 2012] Liu, E. Y.; Guo, Z.; Zhang, X.; Jojic, V.; and Wang, W. 2012. Metric learning from relative comparisons by minimizing squared residual. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, 978–983. IEEE.

[May 1999] May, W. 1999. Information extraction and integration with FLORID: The MONDIAL case study. Technical Report 131, Universität Freiburg, Institut für Informatik. Available from `http://dbis.informatik.uni-goettingen.de/Mondial`.

| ITML | Movie-item | Movie-user | BookCX-year | BookCX-word | Citelike-t | Citelike-a | Mondial |
|---|---|---|---|---|---|---|---|
| Euc | 98.58 ±0.46 | 68.28 ±4.00 | 36.16 ±1.25 | 90.36 ±0.91 | 88.03 ±0.45 | 88.94 ±0.55 | 68.97 ±7.99 |
| Lab | 98.62 ±0.52 | 67.72 ±5.38 | 36.19 ±1.09 | 89.57 ±0.87 | 85.91 ±0.62 | 89.89 ±0.23 | 70.39 ±8.29 |
| Rel | 97.48 ±0.66 | 68.66 ±3.40 | 36.12 ±1.17 | **91.29** ±0.74 | 90.76 ±0.56 | 89.94 ±0.41 | 71.55 ±5.54 |
| Pro | 97.54 ±0.42 | 69.04 ±4.02 | 36.38 ±1.62 | 90.33 ±0.74 | 92.06 ±0.46 | **90.35** ±0.32 | 71.24 ±7.77 |
| Both | **98.67** ±0.50 | **69.48** ±3.08 | **36.97** ±1.69 | 90.43 ±0.72 | **92.65** ±0.40 | **90.35** ±0.32 | **72.00** ±7.16 |

Table 1: The accuracy score of knn with ITML

| LSML | Movie-item | Movie-user | BookCX-year | BookCX-word | Citelike-t | Citelike-a | Mondial |
|---|---|---|---|---|---|---|---|
| Euc | 98.58 ±0.46 | 68.28 ±4.00 | 36.16 ±1.25 | 90.36 ±0.91 | 88.03 ±0.45 | 88.94 ±0.55 | 68.97 ±7.99 |
| Lab | 99.06 ±0.58 | 65.92 ±5.38 | 36.36 ±1.06 | 94.46 ±1.06 | 85.53 ±0.67 | 94.62 ±0.33 | 68.45 ±8.47 |
| Rel | 98.63 ±0.52 | 66.67 ±4.03 | 36.21 ±1.21 | 94.91 ±0.41 | 85.65 ±0.63 | 94.62 ±0.65 | 70.25 ±6.47 |
| Pro | 98.63 ±0.40 | **66.98** ±4.63 | **36.42** ±1.52 | **94.92** ±0.61 | **85.69** ±0.57 | **94.63** ±0.42 | 70.62 ±7.04 |
| Both | **99.12** ±0.52 | **66.98** ±4.63 | **36.42** ±**1.15** | **94.92** ±0.61 | **85.69** ±0.57 | **94.63** ± **0.30** | **71.03** ±7.35 |

Table 2: The accuracy score of knn with LSML

| Proportion | Movie-item | Movie-user | BookCX-year | BookCX-word | Citelike-t | Citelike-a | Mondial |
|---|---|---|---|---|---|---|---|
| 1.0 | 98.62 ±0.52 | 67.72 ±5.38 | 36.19 ±1.09 | 89.57 ±0.87 | 85.91 ±0.62 | 89.89 ±0.23 | 70.39 ±8.29 |
| 0.8 | 98.52 ±0.23 | 66.21 ±3.71 | 36.21 ±1.06 | 89.52 ±0.55 | 86.71 ±0.67 | 89.71 ±0.42 | **72.00** ±7.16 |
| 0.6 | 98.42 ±0.54 | 66.87 ±4.09 | 36.66 ±1.67 | 89.41 ±0.82 | **92.65** ±0.40 | 89.61 ±0.31 | 71.03 ±6.22 |
| 0.4 | **98.67** ±0.50 | 67.65 ±4.82 | **36.97** ±1.69 | 88.87 ±0.78 | 90.87 ±0.38 | 89.91 ±0.35 | 70.82 ±7.66 |
| 0.2 | 97.32 ±1.52 | **69.48** ±3.08 | 36.28 ±1.16 | **90.43** ±0.72 | 91.24 ±0.41 | 90.21 ±0.37 | 69.21 ±7.98 |
| 0.0 | 97.54 ±0.42 | 69.04 ±4.02 | 36.38 ±1.62 | 90.33 ±0.74 | 92.06 ±0.46 | **90.35** ±0.32 | 71.24 ±7.77 |

Table 3: The accuracy score of knn with ITML while the proportion of label constraints and the link-strength constraints gradient change from full label constraints to full link-strength constraints.

| Proportion | Movie-item | Movie-user | BookCX-year | BookCX-word | Citelike-t | Citelike-a | Mondial |
|---|---|---|---|---|---|---|---|
| 1.0 | 99.06 ±0.58 | 65.92 ±5.38 | 36.36 ±1.06 | 94.46 ±1.06 | 85.53 ±0.67 | 94.62 ±0.33 | 68.45 ±8.47 |
| 0.8 | 99.04 ±0.54 | 65.67 ±4.32 | 36.22 ±1.14 | 94.89 ±0.60 | 85.46 ±0.61 | 94.59 ±0.38 | **71.03** ±7.35 |
| 0.6 | **99.12** ±0.52 | 65.78 ±3.80 | 36.13 ±1.18 | 94.78 ±0.58 | 85.53 ±0.79 | 94.61 ±0.41 | 70.22 ±8.42 |
| 0.4 | 99.08 ±0.67 | 66.21 ±4.52 | 36.3 ±1.50 | 94.85 ±0.68 | 85.45 ±0.70 | **94.63** ±**0.30** | 69.79 ±6.54 |
| 0.2 | 98.87 ±0.62 | 66.14 ±5.39 | **36.42** ±**1.15** | 94.83 ±0.86 | 85.45 ±0.67 | **94.63** ±0.52 | 70.67 ±7.22 |
| 0.0 | 98.63 ±0.40 | **66.98** ±4.63 | **36.42** ±1.52 | **94.92** ±0.61 | **85.69** ±0.57 | **94.63** ±0.42 | 70.62 ±7.04 |

Table 4: The accuracy score of knn with LSML while the proportion of label constraints and the link-strength constraints gradient change from full label constraints to full link-strength constraints.


[Sen et al. 2008] Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine* 29(3):93.

[Shaw, Huang, and Jebara 2011] Shaw, B.; Huang, B.; and Jebara, T. 2011. Learning a distance metric from a network. In *Advances in Neural Information Processing Systems*, 1899–1907.

[Tenenbaum, De Silva, and Langford 2000] Tenenbaum, J. B.; De Silva, V.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500):2319–2323.

[Wang, Chen, and Li 2013] Wang, H.; Chen, B.; and Li, W.-J. 2013. Collaborative topic regression with social regularization for tag recommendation. In *IJCAI*, 2719–2725.

[Xing et al. 2003] Xing, E. P.; Ng, A. Y.; Jordan, M. I.; and Russell, S. 2003. Distance metric learning with application to clustering with side-information. In *NIPS*, 505–512.

[Zhai, Peng, and Xiao 2013] Zhai, X.; Peng, Y.; and Xiao, J. 2013. Heterogeneous metric learning with joint graph regularization for cross-media retrieval. In *AAAI*.

[Ziegler et al. 2005] Ziegler, C.-N.; McNee, S. M.; Konstan, J. A.; and Lausen, G. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, 22–32. ACM.