LEARNING A METRIC FOR 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}@univ-nantes.fr

ABSTRACT

The vast majority of metric learning approaches are dedicated to be applied on data described by feature vectors, with some notable exceptions such as times series and trees or graphs. The objective of this paper is to propose metric learning algorithms that consider (multi)-relational data. The proposed approach takes benefit from both the topological structure of the data and supervised labels.

1 INTRODUCTION AND RELATED WORKS

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 et al. (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 A that is used for the Mahalanobis distance $d^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T A(\mathbf{x} - \mathbf{y})$, see reviews in Kulis et al. (2013) and Bellet et al. (2015).

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 et al. (2011). Naturally, a *good* metric should respect the intrinsic structure of the data. Relational databases are increasingly used in almost all applications. 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 algorithm 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 relational information 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 relational information within metric learning, and then illustrate its benefit compared to traditional approaches. Note that the proposed approach starts from (hyper)graph data, whereas approaches as in Dhillon et al. (2012) start from usual tabular data to generate graphs for domain adaptation.

2 RELATIONAL CONSTRAINT SELECTION

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. The basic principle of the approach is to use relational

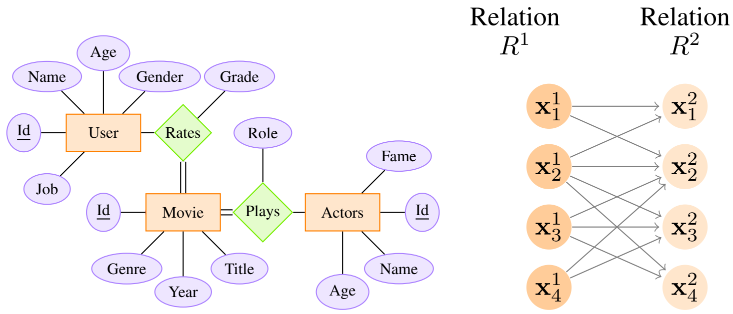

Figure 1: Left : A relational schema for a movie domain. Right : Bipartite relational graph for a *many-to-many* relationship table. The common parents of $\{\mathbf{x}_2^2, \mathbf{x}_3^2\}$ is the set of entities $C = \{\mathbf{x}_2^1, \mathbf{x}_3^1, \mathbf{x}_4^1\}$.

links between entities when setting the constraints of the metric learning algorithm. Mapping the relationship table into the similarity/dissimilarity constraints is a convenient way to use the side-information of the relationships between entities. A relational (or database) schema $R = R_e \cup R_a$ defines a set of relation schema where R_e denotes the set of entity types, and R_a denotes the set of associations between them. Every relation schema $R^k \in R_e$ is described by a set of attributes $A(R^k)$. In addition, every association type $R_a^k \in R_a$ defines a set of references $R(R_a^k)$. Each reference $r \in R(R_a^k)$ has a domain $dom(r) = R_a^k$ and a referenced entity type in R_e. Additionally, the references can be quantified by M numerical variables r_k and M' categorical variables s_k. The Figure 1 presents an example of a movie domain, with 3 entity types (movies, actors, and users) related to each other through 2 association types: Rating, between users and the rated movies, with the corresponding grade; Plays between actors and movies, with the corresponding role an actor has played in the movie. Each entity type in this example has 4 attributes, while each association has 1 attribute and 2 references. There are different type of constraints in metric learning. Roughly speaking, two objects should be considered as similar by the metric if they belong to the same class, and naturally dissimilar if they do not belong to the same class. In practice, we define a set $\mathcal{S}$ of index pairs (i, j) corresponding to similar objects $(\mathbf{x}_i, \mathbf{x}_j)$, and a set $\mathcal{D}$ of index pairs (i, k) corresponding to dissimilar objects $(\mathbf{x}_i, \mathbf{x}_k)$. $\mathcal{S}$ and $\mathcal{D}$ are built using the class labels, as mentioned above. Now, we consider a many-to-many relationship. Let $\mathcal{C}_{ij}$ be the set of common parents of $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\ell_{ij} = |\mathcal{C}_{ij}|$. We define the link strength LS between two entities $\mathbf{x}_i$ and $\mathbf{x}_j$, belonging to the same relation R^k, as follows

$$LS(\mathbf{x}_i^k, \mathbf{x}_j^k | \mathcal{C}_{ij}) = \sum_{k=1}^{\ell_{ij}} (\gamma w(k, i, j) + (1 - \gamma) z(k, i, j)) \tag{1}$$

where

$$w(k, i, j) = \sum_{m=1}^{M} |r_m(c_k, \mathbf{x}_i) - r_m(c_k, \mathbf{x}_j)|\,, \text{ and } z(k, i, j) = \sum_{m=1}^{M'} (s_m(c_k, \mathbf{x}_i) \circ s_m(s_k, \mathbf{x}_j))\,,$$

in which $x \circ y = 1$ iff $x = y$, and 0 otherwise. Note that numerical association attributes r_m are normalized in the unit interval prior to link strength computation. 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.

3 Experiments

Basically, any relational data for which classification is needed can be tackled by our proposition. For brevity reasons, we consider only one, the small MovieLens dataset, see Harper & Konstan (2016). It consists of a relational table which has 100,000 ratings (1-5) with timestamps from 700 users on 9000 movies, a movie entity table with some feature information about the movies and a

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 (1)
  $p \leftarrow p + 1$
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 $\{\mathcal{S}, \mathcal{D}\}$

user entity table with some feature information on users. In this paper, we consider the type of the movie as the supervised label. The balance between association attributes is set to $\gamma = 0.5$. The evaluation of the proposition is done by using k-nn classification via five-fold cross validation, with k set to 5. 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 $k = 5$. We give the results obtained in Table 1. The first two columns indicate the number of constraints that have been generated. The first two lines correspond to the results obtained with a metric learning algorithm (here ITML, Davis et al. (2007)) **without** using relational information (i.e. only labels, baseline algorithm). The next two lines correspond to the **sole** use of relational information (i.e. labels are absolutely not used). The three last lines correspond to the use of both labels and relational information. As can be seen, using the same, rather low, number of constraints, leads to a great improvement in terms of accuracy (48.93 compared to 54.10). On the other hand, this improvement is less visible for a larger number of constraints (56.18 compared to 56.22). Finally, considering both labels and relational information gives the overall best results. Running times are also indicated in this table, showing that the computational cost of the proposed method remains approximately the same as usual methods. It is clear from these results that our proposition, consisting in making use of relations between observations (last 5 lines of Table 1) leads to better results than traditional metric learning algorithms.

Labels	Relations	Accuracy	Time (s.)
13680	0	48.93(± 0.12)	12.05
36594	0	56.18(± 0.17)	19.71
0	13680	54.10(± 0.09)	12.56
0	36594	56.22(± 0.14)	22.41
13680	13680	56.21(± 0.13)	12.23
13680	36594	57.91(± 0.18)	30.89
36594	36594	58.96(± 0.24)	62.67

Table 1: Accuracy as a function of different constraint settings on MovieLens dataset.

4 CONCLUSION

This preliminary 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. Naturally, a deeper study, with more metric learning algorithms, and more data sets, has to be conducted.

REFERENCES

Aurélien Bellet, Amaury Habrard, and Marc Sebban. Metric learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 9(1):1–151, 2015.

Aurelien Bellet, José F Bernabeu, Amaury Habrard, and Marc Sebban. Learning discriminative tree edit similarities for linear classificationapplication to melody recognition. *Neurocomputing*, 214: 155–161, 2016.

Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pp. 209–216. ACM, 2007.

Paramveer S Dhillon, Partha Talukdar, and Koby Crammer. Metric learning for graph-based domain adaptation. In *International Conference on Computational Linguistics*, pp. 255–264, 2012.

Damien Garreau, Rémi Lajugie, Sylvain Arlot, and Francis Bach. Metric learning for temporal sequence alignment. In *Advances in Neural Information Processing Systems*, pp. 1817–1825, 2014.

Lise Getoor. *Introduction to statistical relational learning*. MIT press, 2007.

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

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

Blake Shaw, Bert Huang, and Tony Jebara. Learning a distance metric from a network. In *Advances in Neural Information Processing Systems*, pp. 1899–1907, 2011.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.