ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Learning missing edges via kernels in partially-known graphs

Senka Krivic, Sandor Szedmak, Hanchen Xiong, Justus Piater *

University of Innsbruck - Institute of Computer Science
Technikerstr.21a A-6020 - Innsbruck, Austria

Abstract. This paper deals with the problem of learning unknown edges with attributes in a partially-given multigraph. The method is an extension of Maximum Margin Multi-Valued Regression (M^3VM) to the case where those edges are characterized by different attributes. It is applied on a large-scale problem where an agent tries to learn unknown object-object relations by exploiting known such relations. The method can handle not only binary relations but also complex, structured relations such as text, images, collections of labels, categories, etc., which can be represented by kernels. We compare the performance with a specialized, state-of-the-art matrix completion method.

1 Introduction

Learning object-object relations is a difficult problem with sparse, noisy, corrupted and incomplete information. One way of representing how objects can relate to each other is modeling them by a graph where relations are represented by edges where the vertices play the role of objects. Learning relations is formulated as the problem of predicting edges. Network representations are already used in many scientific fields, e.g. in biology, information technology, or in social sciences [1].

Discovering relations between objects in a large database requires large numbers of experiments, and it is a computationally expensive procedure. Another difficulty is that the data sources can be incomplete, biased or noisy. The approach to learn sparse, incomplete relations used in this paper is an extended version of Maximum Margin Multi-Valued Regression (M^3VM) [2, 3] which is a kernel-based learning framework. This method is at the core of a recommender system proposed by Ghazanfar et al. [2] and has shown state-of-the-art performance in various real-world scenarios, including sparse, skewed and imbalanced datasets. Szedmak et al. [3] applied a version of the same method in learning affordances where the effect of an action on a pair of objects was predicted. Those effects are represented by multi-class labels. Here we extend M^3VM to the class of learning problems where item-item relations might be given by a set of different categorical labels. In a graph representation items can be interpreted as vertices, and multiple relations by multiple edges between the vertices.

*The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 610532, Squirrel and no. 270273, Xperience.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

The learning scenario is a toy clean-up task in a room of kids, where an agent needs to plan how to transform a messy child's room into a tidy one by moving objects to their storage locations and creating order. This scenario raises a number of challenges such as learning spatial relations between objects in the room. An agent can integrate this knowledge into the planning process and use it to refine the world model. Large numbers of objects and their potential interactions in this scenario make this task a large-scale problem. Estimating the missing relations based on those already known, and discovering the underlying structure in a graph where vertices represent objects in the room, can accelerate planning procedures. Learning missing edges in a graph based on those observed earlier can be interpreted as a generalization of semi-supervised learning. M^{3}VM seeks similar edges among the vertices and predicts the missing edges. In this way properties of the existing subset of knowledge are propagated over the whole graph, similarly to models based on random graphs [4, 5].

2 Problem Description

The task of reordering a room involves placing objects at target positions where several cases can occur. For example, *a teddy bear can be in* or *on a box* but also *on a bed* (see the small example in Fig. 1). Relations can also be bidirectional, e.g. *box A can be on box B* and *box B can be on box A*. Thus all possible relations have to be observed between the objects. In this paper four relations were considered: *in, on, below* and *next to*. Hence, the number of attributes describing edges is four, and each can take a value from the set $\{1, -1, 0\}$ denoting *direct, reverse* and *no connection* respectively. We assume that relations are partially known in the agent's database, and the problem is to predict the missing relations. The agent can exploit this knowledge, and based on predictions and on prior knowledge complex relations can be formed, e.g. the teddy bear is *in* box A and box A is *below* the bed, and it is *next to* box B. The nature of this problem implies that missing values do not follow independent and identical distributions, which makes this application interesting.

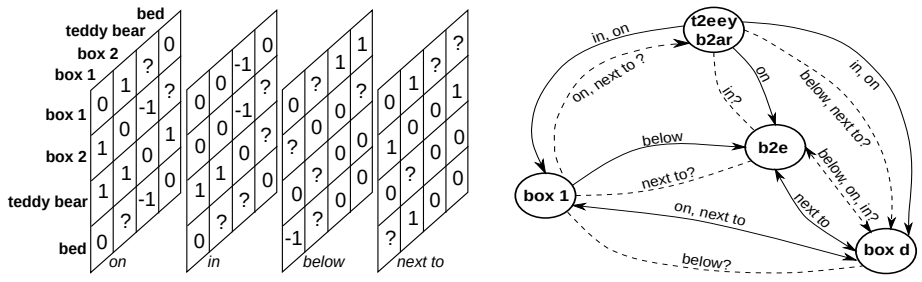

Fig. 1: Small example: (left) Dataset representation; (right) Graph representation

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

3 Description of the relational learner

Learning a relation can be realized via learning a function $f : \mathcal{B} \times \mathcal{U} \to \mathcal{Z}$, where $\mathcal{B}$ and $\mathcal{U}$ are two sets whose elements are connected by that relation, and the elements of the set $\mathcal{Z}$ express the value of the similarity measure between the elements of pairs from the Cartesian product $\mathcal{B} \times \mathcal{U}$.

The function f is indirectly and partially given by a subset $\mathcal{D}$ of all pairs of $\mathcal{B} \times \mathcal{U}$, and the task is to reproduce the function f from the available data of $\mathcal{D}$. Since the elements of sets $\mathcal{B}$, $\mathcal{U}$ and $\mathcal{Z}$ might be given by symbolic objects, e.g. strings or labels, we need to represent them in spaces where their similarities can be measured. To this end we assume that for each of those three sets there is a corresponding function ϕ_B, ϕ_U and ϕ_Z, respectively, which maps those it into a Hilbert space $\mathcal{H}_B$, $\mathcal{H}_U$ and $\mathcal{H}_Z$, respectively. Thus, the inner product between pairs of the elements of $\mathcal{B}$, $\mathcal{U}$ and $\mathcal{Z}$ can be computed within those sets. Those functions are also known as feature representations of their domain.

The set $\mathcal{Z}$ might be equal to $\{0, 1\}$ and describe a simple relation, a multi-valued mapping, between the elements of $\mathcal{B}$ and $\mathcal{U}$. But it could consist of real numbers, expressing e.g. the joint probability of element pairs. It can contain labels of categories [3] or correspond to a range of ranks [2].

The learning problem is given by a set of sample items consisting of three elements (b, u, z_{bu}) where $(b, u) \in \mathcal{D}$. The task is to reconstruct the function f on the full domain $\mathcal{B} \times \mathcal{U}$. In our problem $\mathcal{B} = \mathcal{U}$ and contains the objects to be connected, and $\mathcal{Z}$ contains values taken from $\{-1, 0, +1\}$ for each of the object relations *below*, *in*, *next to*, and *on*.

Since the function f is unknown we need to create a model to describe it. That model has to deal with the complexity, the inhomogeneity of the relationship arising from a mixture of the non-identical and non-independent distributions used to generate the data. For this purpose we apply a piece-wise linear function composed of several linear learners on feature spaces $\mathcal{H}_B$, $\mathcal{H}_U$ and $\mathcal{H}_Z$. To each of the elements of $\mathcal{B}$ we assign a learner $F_b(b, u, z_{bu})$ such that

$$F_b(b, u, z_{bu}|\mathbf{W}_b) = \langle \phi_Z(z_{bu}), \mathbf{W}_b \phi_U(u) \rangle_{H_Z} , \tag{1}$$

where $\mathbf{W}_b$ is a linear operator which maps $\mathcal{H}_U$ to $\mathcal{H}_Z$. The inner product here has greater value if the correlation between the vectors $\mathbf{W}\phi_U(u)$ and $\phi_Z(z_{bu})$ describing the relationship is higher.

To write down the entire optimization problem realizing the learning task we need some additional notation. Let the projections of $\mathcal{D}$ into $\mathcal{U}$ and $\mathcal{B}$ be given by $\mathcal{D}_b = \{u | u \in \mathcal{U}, (b, u) \in \mathcal{D}\}$ and $\mathcal{D}_u = \{b | b \in \mathcal{B}, (b, u) \in \mathcal{D}\}$.

To achieve that the learners will cooperate with each other we define a hinge loss function for all $b \in \mathcal{B}$ to measure the prediction error, namely

$$L_b(b, u, z_{bu}) = \begin{cases} 0 \text{ if } F_b(b, u, z_{bu}|\mathbf{W}_b) \geq 1, \\ \max_u(1 - F_b(b, u, z_{bu}|\mathbf{W}_b)) \text{ otherwise,} \end{cases} \qquad \forall u \in \mathcal{D}_b. \tag{2}$$

Now for a fixed $u \in \mathcal{U}$ all learners $\{F_b | b \in \mathcal{D}_u\}$ acting on a sample instance (b, u, z_{bu}) we prescribe that the corresponding loss functions have to be bounded

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

by only one slack variable ξ_u in the optimization problem, and the value of that slack is minimized. As a consequence we have that ξ_u can have the minimum value if all learners sharing it yield smaller error than ξ_u. Thus their loss is uniformly minimized on the common instances, and hence they cannot vary independently.

The optimization problem expressing the ideas introduced above is stated as

$$\begin{array}{ll} \min & \frac{1}{2}\|\mathbf{W}\|_b^2 + C\sum_{u\in\mathcal{U}}\xi_u \\ \text{with respect to} & \mathbf{W}_b \in (\mathcal{H}_Z \otimes \mathcal{H}_U)^*,\ \boldsymbol{\xi} \in \mathbb{R}^{card(\mathcal{U})}, \\ \text{subject to} & \langle \phi_Z(z_{bu}), \mathbf{W}_b \phi_U(u)\rangle \geq 1-\xi_u,\ b\in\mathcal{B},\ u\in\mathcal{D}_b, \\ & \xi_u \geq 0,\ u\in\mathcal{U}, \end{array} \tag{3}$$

where * denotes the space of the linear operators on the given space. Note that the roles of $\mathcal{B}$ and $\mathcal{U}$ can be swapped. After solving the joint optimization problem we have for all $b \in \mathcal{B}$

$$\mathbf{W}_b = \sum_{u\in\mathcal{D}_b} \alpha_{bu}(\phi_Z(z_{bu}) \otimes \phi_U(u)), \tag{4}$$

where (α_{bu}) are the optimal Lagrangian multipliers. The prediction to a given pair $(\hat{b}, \hat{u})$ can be derived by

$$\begin{aligned} z^*_{\hat{b}u} &= \max_{\hat{b}\hat{u}\in\mathcal{D}} \phi_Z(z_{\hat{b}\hat{u}})\mathbf{W}_{\hat{b}}\phi_U(\hat{u}) = \max_{\hat{b}\hat{u}\in\mathcal{D}} \phi_Z(z_{\hat{b}\hat{u}}) \sum_{u\in\mathcal{D}_{\hat{b}}} \alpha_{\hat{b}u}(\phi_Z(z_{\hat{b}u}) \otimes \phi_U(u))\phi_U(\hat{u}) \\ &= \max_{(\hat{b}\hat{u})\in\mathcal{D}} \sum_{u\in\mathcal{D}_{\hat{b}}} \alpha_{\hat{b}u} \underbrace{\langle \phi_Z(z_{\hat{b}\hat{u}}), \phi_Z(z_{\hat{b}u})\rangle}_{K(z_{\hat{b}\hat{u}}, z_{\hat{b}u})} \underbrace{\langle \phi_U(u), \phi_U(\hat{u})\rangle}_{K(u,\hat{u})}, \end{aligned}$$

where $K(z_{\hat{b}\hat{u}}, z_{\hat{b}u})$ and $K(u, \hat{u})$ are kernel matrices built on the inner product between the corresponding elements.

4 Experimental Results

The presented method was tested on a dataset created from the Princeton Shape Benchmark database [6]. Prior edges in the graph were made based on labels created by hand for the 761 items from the database.[1] Four types of relations were considered as described in Section 2. This implies a possible number of connections in the network of 2316484, which makes this a large-scale problem. Moreover, this task requires the handling of a large proportion of missing data (see Table 1).

The problem of predicting edges is equivalent to the completion of the adjacency matrix of the graph. The method is compared with the Augmented Lagrange Multiplier Method (ALM technique) [7, 8] designed to recover corrupted low-rank matrices. Table 1 shows that our general-purpose approach is competitive with this state-of-art, specialized matrix completion method.

[1] The dataset is available on request.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

In the learning procedure, first the base kernels are computed from the corresponding features. The kernels are defined by the following rules: $K(z_{bu}, z_{bu'})$ is equal to the number of shared attributes between the multiple edges bu and bu', and $K(u, u') = \sum_b K(z_{bu}, z_{bu'})$ where the sum includes all terms that have at least one edge in common between bu and bu'. The penalty factor C of (3) is set to 1, and the kernel $K(z_{bu}, z_{bu'})$ is Gaussian with parameter 0.5 for all datasets.

Dataset		ALM		M^3VM	
relations	known	binary	triplets	binary	triplets
below	13.3	98.79(10.6)	99.84(6.6)	**99.96**(4.1)	**99.92**(3.9)
in	15.2	**99.83**(16.1)	95.83(15.2)	98.40(4.5)	**98.21**(5.1)
next to	44.0	91.37(24.4)	91.34(23.4)	**97.64**(7.4)	**97.52**(7.9)
on	15.1	**99.89**(14.8)	**99.99**(0.3)	99.35(4.6)	99.49(4.3)
all four	22.2	95.34(30.9)	95.31(29.5)	**97.7**(32.8)	**98.5**(34.8)

Table 1: Number of known relations in %. Accuracies in % (training time in seconds) of the predictions on the four test sets. The *binary* case represents the existence of edges in undirected graphs (-1, 1), and *triplets* in directed graphs (-1, 0, $+1$).

A 5-fold cross-validation procedure is applied on the known data while prediction is done on the whole dataset. The parameter corresponding to each kernel is found by cross validation restricted to the training data. Error is measured by root-mean-square error (RMSE), and the training time in seconds. For the comparison of computational power needed for both methods, additional testing was done on large-scale datasets (see Table 2). The datasets[2] used in these experiments consist of annotated images given by binary labels expressing that the image has a certain property, e.g. it contains a building or a tree. We consider two images as similar if they share common labels. The number of those labels measures the similarity between each pair of the images.

"Corel5k" (3.97M connections)		"Espgame" (83.04M connections)	
ALM	M^3VM	ALM	M^3VM
0.3921 (270.2)	**0.3338 (25.7)**	0.4676 (4973.6)	**0.4603 (429.5)**

Table 2: Accuracies in % (training time in second) of the predictions on additional datasets for comparing computational power. The number of connections corresponds to those pairs of images which share at least one label, and it is given in millions (M).

The Singular Vale Thresholding (SVT) [9] and the TenAls algorithm [10] were also tested on the presented problem. However, these methods diverge on the dataset used since they make the strong assumption of Gaussianity about the distribution of missing values, unlike M^3VM.

[2]The datasets can be found at http://lear.inrialpes.fr/people/guillaumin/data.php.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

5 Conclusion

In this paper we have shown how M^3VM can be applied to learn unknown multiple edges in graphs. The case study is a room reordering task, where we predicted possible missing relations between the objects. The presented method M^3VM is capable of predicting edges with multiple attributes via kernels. Predicted relations between items can guide the agent in new exploration. The same principle can be used also for making and/or consolidating assumptions on item features in a knowledge database which can be used in different scenarios, e.g., the presented scenario of reordering the room. The accuracy of the testing results gives confidence to the agent in learning relations and can be used in exploration tasks. As presented, this method can be applied to problems concerning categorical values as well as tensors. Thus it can easily be extended to problems of learning edges of protein interactions, or to recommending connections within social networks based only on already-existing links.

References

[1] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. *Found. Trends Mach. Learn.*, 2(2):129–233, February 2010.

[2] M. A. Ghazanfar, A. Prügel-Bennett, and S. Szedmak. Kernel-mapping recommender system algorithms. *Inf. Sci.*, 208:81–104, 2012.

[3] S. Szedmak, E. Ugur, and J. Piater. Knowledge propagation and relation learning for predicting action effects. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 623–629, Sept 2014.

[4] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[5] B. Bollobás. Random graphs. In *Modern Graph Theory*, volume 184 of *Graduate Texts in Mathematics*, pages 215–252. Springer New York, 1998.

[6] P. Min P.Shilane, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. In *Shape Modeling International*, June 2004.

[7] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *UIUC Technical Report UILU-ENG-09-2215 arXiv:1009.5055.*, 2009.

[8] Z. Lin, R.Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low-rank representation. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 612–620. Curran Associates, Inc., 2011.

[9] E. J. Candes and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[10] P. Jain and S. Oh. Provable tensor factorization with missing data. In Z. Ghahramani, M. Welling, C. Cortes, N.d. Lawrence, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1431–1439. Curran Associates, Inc., 2014.