

Learning to Match

Jun Xu, Zhengdong Lu, Tianqi Chen, Hang Li

1. Introduction

The main tasks in many applications can be formalized as matching between heterogeneous objects, including search, recommendation, question answering, paraphrasing, and image retrieval. For example, search can be viewed as a problem of matching between a query and a document, and image retrieval can be viewed as a problem of matching between a text query and an image.

A variety of machine learning techniques have been developed for various matching tasks. We refer to them as ‘learning to match’, which can be formalized as the following learning problem. Suppose that there are two spaces X and Y . Objects x and y belong to the two spaces X and Y , respectively. There is a set of numbers R . Number r belongs to the set R , representing a matching degree. A class of ‘matching functions’

$F = \{f(x, y)\}$ is defined, while the value of the function is a real number. Training

data $\{(x_1, y_1, r_1), (x_2, y_2, r_2), \dots, (x_N, y_N, r_N)\}$ is given, where each instance consists of

objects x and y as well as their matching degree r . The data is assumed to be

generated according to the distributions $(x, y) \sim P(X, Y), r \sim P(R | X, Y)$. The goal of

the learning task is to select a matching function $f(x, y)$ from the class F based on the

observation of the training data. The matching function can be used in classification, regression, or ranking of two matching objects. The learning task, then, becomes the following optimization problem.

$$\arg \min_{f \in F} \sum_{i=1}^N L(r_i, f(x_i, y_i)) + \Omega(f)$$

where $L(\cdot, \cdot)$ denotes a loss function and Ω denotes regularization.

Learning to match is unique in that it is about learning of a two-input function $f(x, y)$,

in contrast to learning of a one-input function in conventional classification and regression. There usually exist relations between the two inputs x and y , and we can and should leverage the relations to enhance the accuracy of learning. In fact, the inputs x and y can be instances (IDs), feature vectors, and structured objects, and thus the task can be carried out at instance level, feature level, and structure level, as shown in Figure 1.

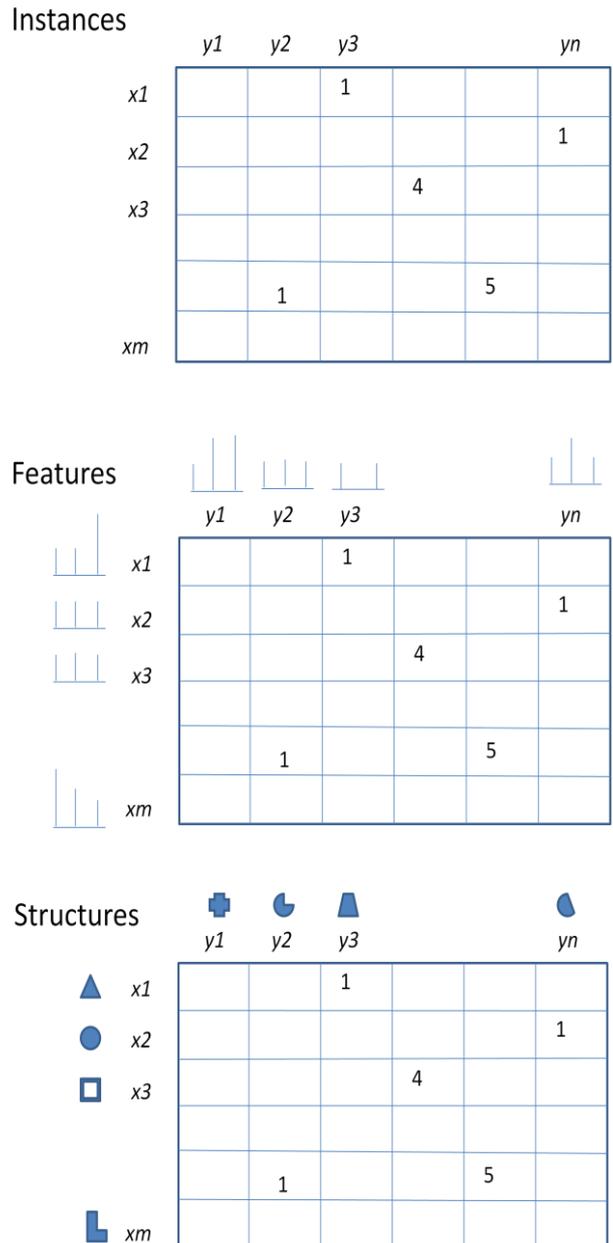


Figure 1. Matching between heterogeneous objects can be performed at instance, feature, and structure levels.

We think that it is necessary and important to conduct research on learning to match. To generalize the techniques for matching developed in different applications, we can make better understanding on the problems, develop more powerful machine learning methodologies, and apply them to all the applications.

At Noah's Ark Lab, we have invented learning to match techniques for document retrieval, recommendation, natural language processing, and image retrieval, as described below. Actually, the general view on learning to match has helped us a lot in the development of the technologies.

2. Learning to Match for Document Retrieval

Document retrieval or search is a problem mainly based on matching. Current search systems adopt the term-based approach, in which documents are retrieved based on the query terms, and the relevance scores of the documents with respect to the query are mainly calculated on the basis of the matching degrees between the query and documents at term level. In such a framework, query document mismatch often arises, as the most critical problem that stands on the way of a successful search. It occurs, when the searcher and the author use different terms to represent the same concept, e.g., NY and New York. The phenomenon is prevalent due to the flexibility and diversity of human languages. A fundamental solution to the problem would be to perform ‘semantic matching’, in which the query and a document can match if they are semantically similar. We can leverage learning to match techniques to train models to carry out semantic matching [Li & Xu, 2014].

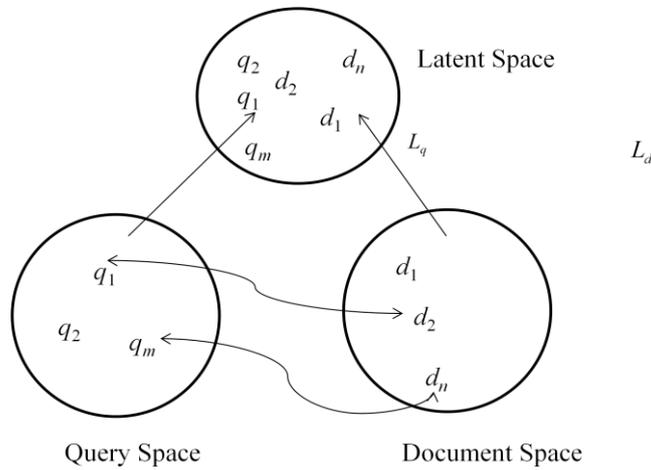


Figure 2. Regularized Mapping to Latent Space

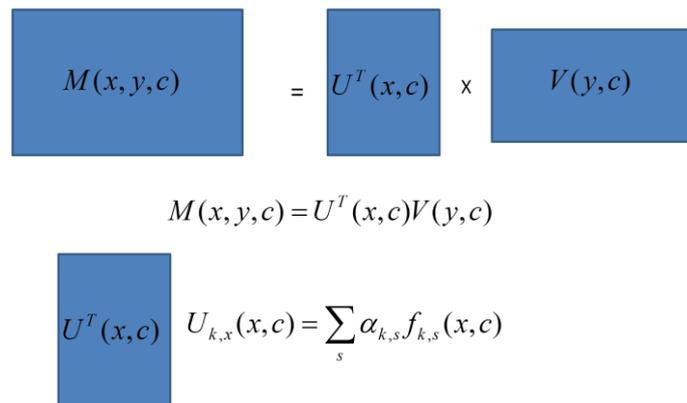
In [Wu et al., 2013a; Wu et al. 2013b], a new method for learning to match called regularized mapping to latent space (RMLS) is proposed. RMLS learns two linear mapping functions for respectively mapping two objects (query and document) into a latent space, and taking the dot product of their images in the latent space as matching degree between the two objects (cf. Figure 2). RMLS is an extension of the method of Partial Least Square (PLS). To enhance the scalability, RMLS replaces the orthogonal constraint in PLS with L_1 and L_2 regularization. In this way, learning can be performed in parallel and thus is scalable and efficient. RMLS is applied to web search where a large click-through dataset including associated queries and documents is utilized as training data. The result shows that both PLS and RMLS can significantly outperform the baseline methods, while RMLS can substantially speed up the learning process.

One shortcoming with RMLS is that it is hard to train a model for tail queries and tail documents for which there is not enough click through data. In [Wang et al., 2015], a

new method is proposed to tackle the challenge by using not only click-through data but also semantic knowledge in learning of the RMLS model. The semantic knowledge can be categories of queries and documents as well as synonyms of words. The objective function of the method is defined such that semantic knowledge is incorporated through regularization. Two methods for conducting the optimization using coordinate decent and gradient decent are also proposed. Experimental results on two datasets demonstrate that the new model can make effective use of semantic knowledge, and can significantly enhance the accuracy of RMLS, particularly for tail queries.

3. Learning to Match for Recommendation

Recommender system is another area in which matching plays an important role. In the collaborative filtering setting, the essential problem is to recommend product items to users, which is equivalent to matching product items to users. The problem can be formalized as matrix factorization, when only IDs of users and items are provided. It can also be extended to feature-based matrix factorization when features of users and items are also available.



$$M(x, y, c) = U^T(x, c) \times V(y, c)$$

$$M(x, y, c) = U^T(x, c)V(y, c)$$

$$U^T(x, c) U_{k,x}(x, c) = \sum_s \alpha_{k,s} f_{k,s}(x, c)$$

Feature based matrix factorization, where x and y denote objects, and c denote context. Each element of matrix is a linear combination of feature functions.

Figure 3. Functional Matrix Factorization

In feature-based matrix factorization, the feature vectors in the feature space are mapped into a latent space. One key question here is how to encode useful features to enhance the performance of matching. Ideally one would like to automatically construct ‘good’ feature functions in the learning process. In [Chen et al., 2013], we propose formalizing the problem as general functional matrix factorization, whose model includes conventional matrix factorization models as special cases. Specifically, in our method feature functions are automatically created through search in a functional space during the process of learning. We propose a gradient boosting based algorithm to efficiently solve the optimization problem, with the general framework

outlined in Figure 3. The experimental results demonstrate that the proposed method significantly outperforms the baseline methods in recommendation.

Another important issue is how to scale up learning to match to extremely large scale recommendation problems. We propose a parallel algorithm to tackle the challenge in feature based matrix factorization [Shang et al., 2014]. In general, it is very hard to parallelize the learning of feature-based matrix factorization, because it requires simultaneous access to all the features. In our work, we solve the problem by leveraging the properties of learning to match. The algorithm parallelizes and accelerates the coordinate descent algorithm by (1) iteratively relaxing the objective function to facilitate parallel updates of parameters, and (2) storing intermediate results to avoid repeated calculations with respect to the features. Experimental results demonstrate that the proposed method is very effective on a wide range of matching problems, with efficiency significantly improved.

4. Learning to Match for Natural Language Processing

Many problems in natural language processing depend on matching of two linguistic objects, for example, two sentences. In paraphrasing, one needs to decide whether two sentences are synonymous expressions to each other; in question answering, one needs to figure out whether one sentence is an answer to the other sentence; in dialogue one needs to judge whether the two sentences form one round of conversation. That is to say, all of them amount to matching of two sentences, while the matching relations vary according to tasks. Note that the two sentences have structures and thus we need to consider how to perform matching of sentences at structure level.

Recently we have developed several models for matching between sentences in natural language, using deep learning [Lu & Li, 2013; Hu et al., 2014; Wang et al., 2015]. The deep matching models are employed in paraphrasing, question answering, and single turn dialogue.

Deep Match CNN [Hu et al., 2014] is a neural network model for matching a pair of sentences in a specific task. There are two architectures: Arc-I and Arc-II. In the former architecture, given two sentences, each of them is fed into a one-dimensional convolutional neural network, yielding a representation of the sentence, and then the two representations are given to a multi layer network, generating a matching score between the two sentences (Figure 4). The model is trained with a large number of sentence pairs in the task. In the latter architecture, given two sentences, both of them are input into a two-dimensional convolutional neural network, outputting a matching score. The model captures both the representations of the two sentences and the interactions between the two sentences in the matching problem. The major advantage of Deep Match CNN lies in its flexibility and robustness. Given matching data for any

task in any language, it can automatically learn a model, without the need of exploiting linguistic knowledge.

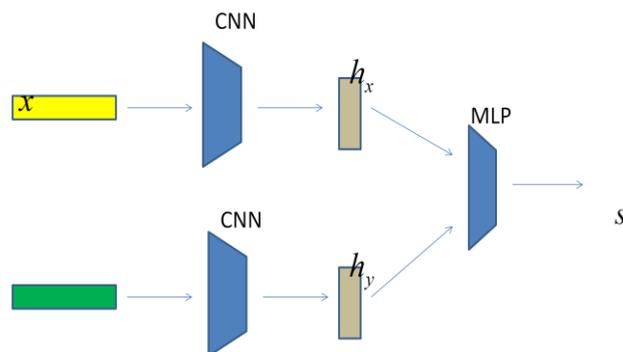


Figure 4. Deep Match CNN Model (Arc-I)

Another matching model is Deep Match Tree, whose main characteristic is to utilize linguistic knowledge (Figure 5). Given two sentences, a dependency parse tree is created for each sentence using a dependency parser. All the subtree pairs between the two parse trees are collected and fed into a multi-layer neural network. A final matching score is output from the network. Each neuron of the first layer corresponds to one subtree matching pattern. All the subtree matching patterns are mined from a large data corpus in advance. The corresponding neuron will become one if a subtree pair agrees with a subtree matching pattern. Intuitively, the more matched patterns there are, the higher the final score will be, i.e., the two sentences will be matched.

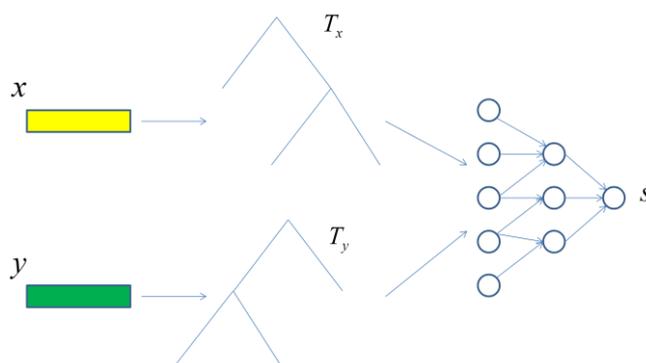


Figure 5. Deep Match Tree Model

5. Learning to Match for Image Retrieval

Image and text are heterogeneous data. Image retrieval is a problem as follows. Given a query in natural language, we retrieve from a collection of images the most relevant ones and return them in a ranking list. The key problem is, again, matching between heterogeneous data and the challenge is that the data is very different in nature.

We have developed a model called multimodal convolutional neural network (m-CNNs) for matching image and text. As shown in Figure 6, m-CNN employs convolutional architectures to model image and text matching. More specifically, it

exploits one CNN for encoding the image and one CNN for encoding the text, as well as the matching relations between image and text. Experimental results demonstrate that m-CNN can effectively capture the information necessary for image and text matching, and significantly outperform the state-of-the-art methods in bidirectional image and text retrieval.

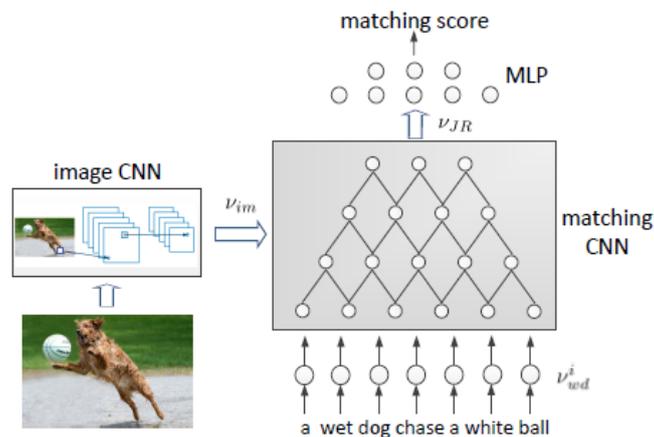


Figure 6. Multimodal Convolutional Neural Network

6. Open Problems

There are still many open questions to address in the research on learning to match. We list some of them here.

- **Unified model:** Different models have been proposed for different tasks. One may ask whether there is a general model which can be employed in all matching problems, for example, a powerful deep neural network.
- **Incorporation of knowledge:** How to incorporate human knowledge into the matching model in order to improve the accuracy of matching in different tasks is still an open problem. It is not clear how to do it in an effective and a theoretically sound way, beyond regularization.
- **Relations between objects:** Sometimes relations between the objects exist in the form of network. How to make effective use of the information in the matching model is still an important research topic, although some research has been done.
- **Online learning:** In some cases, one may want to train the model in an online fashion, which requires new learning algorithms.
- **Theory:** Existing methods on learning to match are still lack of theoretical support. For example, it is not clear whether some of the methods have statistical consistency, i.e., the model learned converges to the 'true' model.
- **Scalability and efficiency:** In practice, the scale of some of the matching problems is extremely large. How to make the learning process more scalable and efficient needs more investigation.

References

- Tianqi Chen, Hang Li, Qiang Yang, Yong Yu, General Functional Matrix Factorization Using Gradient Boosting, In Proceedings of 30th International Conference on Machine Learning (ICML), JMLR W&CP 28(1):436-444, 2013.
- Baotian Hu, Zhengdong Lu, Hang Li, Qingcai Chen, Convolutional Neural Network Architectures for Matching Natural Language Sentences, in Proceedings of Advances in Neural Information Processing Systems 27 (NIPS), 2042-2050, 2014.
- Hang Li, Jun Xu, Semantic Matching in Search, Foundations and Trends in Information Retrieval, Now Publishers, 2014.
- Zhengdong Lu, Hang Li, A Deep Architecture for Matching Short Texts, In Proceedings of Neural Information Processing Systems 26 (NIPS), 1367-1375, 2013.
- Lin Ma, Zhengdong Lu, Lifeng Shang, Hang Li, Multimodal Convolutional Neural Networks for Matching Image and Sentence. Proceedings of 2015 IEEE International Conference on Computer Vision (ICCV), 2015. to appear.
- Jingbo Shang, Tianqi Chen, Hang Li, Zhengdong Lu, Yong Yu, A Parallel and Efficient Algorithm for Learning to Match, in Proceedings of IEEE International Conference on Data Mining (ICDM), 971-976, 2014.
- Shuxin Wang, Xin Jiang, Jun Xu, Hang Li, Bin Wang, Incorporating Semantic Knowledge into Latent Matching Model in Search, to appear, 2015.
- Mingxuan Wang, Zhengdong Lu, Hang Li, Qun Liu, Syntax-based Deep Matching of Short Texts, in Proceedings of International Conference on Artificial Intelligence (IJCAI'15), 2015.
- Wei Wu, Hang Li, Jun Xu, Learning Query and Document Similarities from Click-through Bipartite Graph with Metadata, In Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM), 687-696, 2013.
- Wei Wu, Zhengdong Lu, Hang Li, Learning Bilinear Model for Matching Queries and Documents, Journal of Machine Learning Research (JMLR), 14: 2519-2548, 2013.