A Unified Learning Paradigm for Large-scale Personalized Information Management

Edward Y. Chang[2], Steven C.H. Hoi[3], Xinjing Wang[1], Wei-Ying Ma[1], Michael R. Lyu[3]

[1]Microsoft Research Asia, 49 Zhichun Road, Beijing, China
[2]Electrical & Computer Engineering, University of California, Santa Barbara
[3]Computer Science & Engineering, Chinese University of Hong Kong, Hong Kong

ABSTRACT

Statistical-learning approaches such as unsupervised learning, supervised learning, active learning, and reinforcement learning have generally been separately studied and applied to solve application problems. In this paper, we provide an overview of our newly proposed unified learning paradigm (ULP), which combines these approaches into one synergistic framework. We outline the architecture and the algorithm of ULP, and explain benefits of employing this unified learning paradigm on personalizing information management.

1. INTRODUCTION

Human beings learn by being taught (supervised learning), by self-study (unsupervised learning), by asking questions (active learning), and by being examined for the ability to generalize (reinforcement learning), among many ways of acquiring knowledge. An integrated process of supervised, unsupervised, active, and reinforcement learning provides a foundation for acquiring the known and discovering the unknown.

It is natural to extend the human learning process to machine learning tasks. In this paper, we propose a unified learning paradigm (ULP), which combines several machine-learning techniques in a synergistic way to maximize the effectiveness of a learning task. Three characteristics distinguish ULP from a traditional hybrid approach such as semi-supervised learning. First, ULP aims to minimize the human effort in collection of quality labeled data. Second, ULP uses the stability of the membership of unlabeled data, together with active learning, to ensure sufficiency of both labeled and unlabeled data, thus guaranteeing the generalization ability of the learned result. Third, ULP uses active learning and reinforcement learning (or some other techniques) to access the convergence of the learning process.

More specifically, ULP is an interactive algorithm consisting of four steps. The first step uses a prior kernel function to generate a kernel matrix. The second step employs unsupervised learning algorithms to measure the stability of selected pairs of unlabeled instances in the kernel matrix. The similarity (or dissimilarity) of a pair of instances is reinforced when the changes of parameters of the clustering algorithms do not affect the instances' cluster memberships. For instance, suppose we employ the spectral clustering algorithm [7] to cluster data. If a pair of instances always belongs to the same cluster, despite various choices of the kernel parameter and the number of clusters, we increase the similarity score of the pair. If a pair is distributed to several different clusters, we can comfortably decrease the similarity scores. When the membership of an unlabeled pair is unstable, ULP in its third step uses active learning to solicit user feedback to confirm the similarity score. The "questions" selected by the active learning component must balance between two sub-goals: maximizing information gain and maximizing generalization. The last step of ULP tests for the convergence conditions. If the algorithm has not converged, ULP returns to the second step using the newly aligned kernel matrix to conduct an unsupervised, membership stability test. When the algorithm converges, ULP outputs a kernel matrix [6] or function [5].

ULP is essential for large-scale information management. First, for a large-scale task, a supervised approach for pattern analysis or knowledge discovery is not scalable. ULP uses the unlabeled data in the most effectively way to reduce the need for a large amount of labeled data. Second, the essence of finding information relevant to a query or to a user is to formulate a distance function that best describes how the user perceives similarity. ULP outputs such a function, which can then be used in tasks of information organization and retrieval. (The labeled and unlabeled data can be the history of a user's document access, or his/her desk-top profile.) ULP learns a kernel based on data; and therefore, it can work more effectively

than the traditional way of selecting a kernel in a data independent way.

In the rest of this paper, we discuss in more details on the four major synergistic steps of ULP. Because of space limitations, we refer to the reader our published papers for more details.
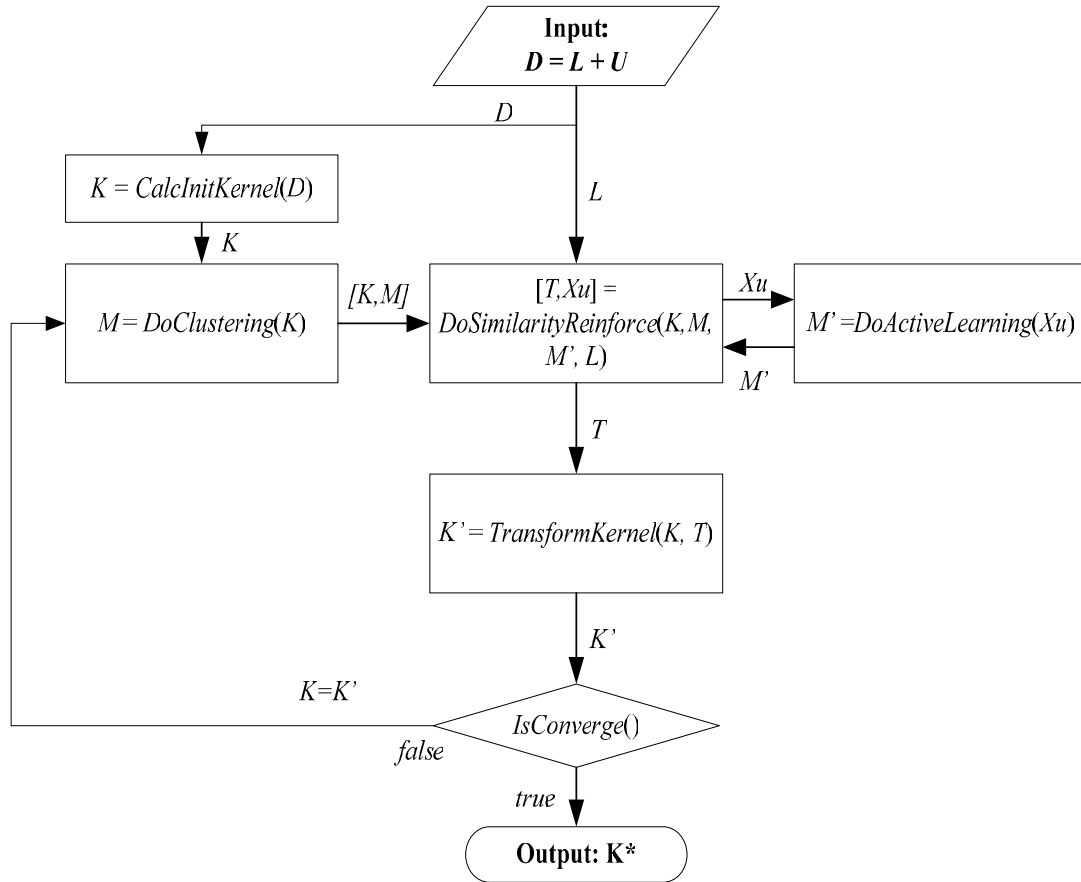

Figure 1. The Architecture of Unified Learning Paradigm.

2. ULP ARCHITECTURE

Figure 1 illustrates the architecture of our proposed Unified Learning Paradigm. Basically, the ULP scheme comprises five main components: *clustering module, active learning module, similarity reinforcement module, kernel transformation module,* and *convergence evaluation module*. The clustering module selects the set of unlabeled instances in the kernel matrix that are either stable or the most unstable according to different parameter settings, and then transmits their corresponding membership information M to the similarity reinforcement module. In the initial step, the similarity reinforcement module collects the information provided by kernel matrix K, data membership M, and the original data labels L (if they are available). Then it produces the possible kernel transformation function T as well as the most uncertain data subset Xu that requires active learning. Based on users' relevance feedback, the active learning module than returns the membership information

M' back to the reinforcement module to learn a new T, and possibly a new Xu. When the similarity reinforcement module gains enough confidence in its produced result, it sends the kernel transformation function T to the kernel transformation module to generate a modified kernel K'. Note that K' is assumed to better reveal the intrinsic similarities of D. K' then passes through the convergence evaluation module for a convergence test. If it passes the test, the ULP algorithm ends; otherwise K' is sent back to the clustering module for another iteration.

Next, we discuss the individual modules in detail.

2.1 THE CLUSTERING MODULE

In this module, we attempt to find out those salient data instances by measuring the stability of the membership of unlabeled data. The salient instances include either the data that suggest the underlying structure of the dataset (i.e. the most stable) or those that result in the largest information gain if we manually label them (i.e., the most unstable).

For example, assume we are given a set of data instances $D = \{x_1, \ldots, x_n\}$ and we want to learn the membership matrix $M = [m_{ij}]_{n \times n}$ from the data, in which the elements are set initially to zero. Suppose we run a clustering algorithm on D. A clustering partition C_p can be obtained. If two data instances x_i and x_j are grouped in the same cluster in C_p, we update the corresponding element in the membership matrix $m_{ij} = m_{ij} + 1$. This clustering procedure is run a number of times by adopting some kinds of perturbation. For instance, we can change the k value of the k-mean step in the spectral-cluster algorithm, or we can change the parameter of a selected kernel. For each clustering result, we can access its quality (or confidence) by using a measure such as eigengap. The overall membership matrix is obtained by weighting all the clustering results with clustering quality. Eventually, a membership stability matrix can be obtained from the clustering module. The resultant membership knowledge, after being processed by the similarity reinforcement module, is important for both active learning and kernel transformation purposes.

There are several research topics in this step:
      a) What clustering algorithm(s) [1] should be employed?
      b) How might we best measure membership stability and select the salient data instances?

      c) How many unstable instances should be selected?

2.2. THE SIMILARITY REINFORCEMENT MODULE

From the clustering module, we can obtain sets of similar pairs and dissimilar pairs according to the membership M of clustering results. However, M may be noisy because of the limitation of extracted features. To reduce the factor of noise, we adopt the similarity reinforcement module [2]. Another important role for this module is to deduce a kernel transformation function T based on the information supplied by the clustering module, the active learning module, the labeled data L, and even the history information (history of kernel matrix updates).

A possible solution of this module is to propagate the similarity from the labeled data to the unlabeled data. When this module is uncertain about the labels of some data instances, it simply gives these data to the active learning module to solicit relevance feedback. And the feedback information (i.e., another group of labeled data) can be leveraged for learning a better T.

Some interesting research topics pertaining to this module are:
1) How do we formulate T?
2) How might the similarity information best be propagated from labeled data to unlabeled data?
3) How should Xu be selected for conducting active learning?

2.3. THE ACTIVE LEARNING APPROACH

In some challenging learning tasks, active learning is essential for bridging the *knowledge gap* of the given data [3]. Although we can acquire the membership knowledge from the *clustering* and the *similarity reinforcement* steps, we cannot guarantee that the knowledge is noise-free and complete. In order to remedy this shortcoming, we employ active learning.

Typically we employ active learning to solicit feedback to confirm the similarity score in order to bridge the knowledge gap. When we choose a "question" to ask the user, we need to balance two sub-goals: maximizing information gain and maximizing generalization. For example, to maximize the information gain, we can consider choosing the data instances that are most effective to propagate the similarity to other pairs; we can also pick the instances that are most uncertain according to the membership knowledge. Or to maximize generalization, we can choose the data samples that have

the greatest potential to increase the generalization performance.

After the active learning module has collected feedback information, the feedback knowledge can be used to refine the similarity correlation between data instances.

2.4. THE KERNEL TRANSFORMATION MODULE

Kernel functions or kernel matrices are essential for many machine learning tasks [4]. It is still a challenging open research problem to devise effective kernel functions or matrices from labeled and unlabeled data. Our solution is to study the kernel transformation techniques by combining the prior knowledge of labeled data and the membership knowledge of unlabeled data exploited from the clustering results and active learning.

From the *similarity reinforcement* step, we have learned the transformation knowledge T which provides the guide to learn the kernel functions or matrices. In order to implement the kernel transformation toward an effective kernel functions, two problems should be considered. One is the positive semi-definite (PSD) issue: It is important for a distance metric to satisfy the PSD condition as explained in [6]. Another is the generalization performance of the kernel functions. How to make the kernel functions generalize well to the unseen data is important for developing the kernels.

In sum, from the *kernel transformation* step, we can learn new kernel functions [5] or matrices [6] according to the explicit or implicit similarity knowledge from the data. Our preliminary results [5,6] show both avenues to be effective.

2.5. THE CONVERGENCE EVALUATION MODULE

In the ULP framework, when a new kernel function or matrix is obtained, we run the whole procedure iteratively until the convergence conditions are satisfied. There are several factors to determine the convergence conditions. One is to ascertain whether our similarity knowledge matches the user feedback. Furthermore, we can verify whether the knowledge gap has been bridged, by measuring the completeness of similarity linkages between the labeled and unlabeled data.

After the convergence conditions are satisfied, we have obtained an improved kernel matrix K^*. One more possible step is to deduce a kernel function based on it [5]. Alternatively, we can use the matrix to conduct generalization [6].

3. CONCLUSIONS

In this paper we outlined a foundational framework for learning the kernel functions or matrices from labeled and unlabeled data in combinations of unsupervised learning, supervised learning, active learning, and reinforcement learning. We presented our preliminary results through references, and discussed our future work. We believe that ULP is an essential tool to work with large-scale, personalized information management. Our other endeavors for speeding up kernel methods in large-scale settings, which complements ULP, can be found in [8,9].

4. REFERENCES

[1] A K Jain and M N Murty. Data Clustering: A Review. *ACM Computing Surveys,* 31(3), pp.264-323, 1999

[2] L. P. Kaelbling, M. L. Littman, A. W. Moore. Reinforcement Learning: A Survey (1996). *Journal of Artificial Intelligence Research*, vol.4, pp.237-285, 1996

[3] S. Tong and E. Y. Chang, Support Vector Machine Active Learning for Image Retrieval, In *Proceedings ACM International Conference on Multimedia*, pp.107-118, Ottawa, October 2001

[4] N. Cristianini, J. Kandola, A. Elisseeff and J. Shawe-Taylor On Kernel-Target Alignment, *Journal of Machine Learning Research*, 1, 2002

[5] G. Wu, E. Y. Chang, and N. Panda, Formulating Distance Functions via the Kernel Trick, In *ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, Chicago, August 2005.

[6] G. Wu, Z. Zhang, and E. Y. Chang, An Analysis of Transformation on Non-Positive Semidefinite Similarity Matrix for Kernel Machines, *UCSB Technical Report*, March 2005.

[7] F.R. Bach and M.I. Jordan. Learning spectral clustering. In *Advances in Neural Information Processing Systems*, 16, 2004.

[8] Exploiting Geometric Property for Support Vector Machine Indexing, N. Panda and E. Y. Chang, *SIAM International Conference on Data Mining (SDM)*, Newport Beach, April 2005.

[9] Kronecker Factorization for Speeding up Kernel Machines, G. Wu, Z. Zhang, and E. Y. Chang, *SIAM International Conference on Data Mining (SDM)*, Newport Beach, April 2005.