The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Lifelong Learning Networks: Beyond Single Agent Lifelong Learning

Mohammad Rostami
Department of Electrical and Systems Engineering
University of Pennsylvania
mrostami@seas.upenn.edu

Eric Eaton
Department of Computer and Information Science
University of Pennsylvania
eeaton@seas.upenn.edu

Abstract

Lifelong machine learning (LML) is a paradigm to design adaptive agents that can learn in dynamic environments. Current LML algorithms consider a single agent that has centralized access to all data. However, given privacy and security constraints, data might be distributed among multiple agents that can collaborate and learn from collective experience. Our goal is to extend LML from a single agent to a network of multiple agents that collectively learn a series of tasks.

Introduction

Collective learning in a network of collaborating agents can potentially improve learning speed and performance of all agents. It is the product of individual agents, each with their own interests and constraints, sharing and accumulating learned knowledge over time. Recent work in LML (Ruvolo and Eaton 2013) has explored the notion of a single agent accumulating knowledge over time. Such an individual LML agent reuses knowledge from previous tasks to improve its learning on new tasks. This LML process improves performance over all tasks. Although current work in LML focuses on a single learning agent that incrementally perceives all task data, many real-world applications involve scenarios in which multiple agents must collectively learn a series of tasks. Consider the following scenarios:

- Task data could only be partially accessible by each agent. For example, financial decision support agents may have access only to one data view or a portion of the data.
- Local data processing can be inevitable, such as when health care regulations prevent medical data from being shared between learning systems or for security concerns.
- Data communication may be costly. For instance, home service robots must process perceptions locally due to the volume of perceptual data, or wearable devices may have limited communication bandwidth or battery power.
- Data must be broked and processed in parallel. Big data often necessitates parallel processing in the cloud across multiple virtual agents, i.e. CPUs or GPUs.

Inspired by these scenarios, we want to explore the idea of *lifelong learning networks*. We consider multiple LML agents, each facing their own series of tasks, that transfer knowledge to collectively improve performance and learning speed. To develop algorithms for lifelong learning networks, we follow a parametric approach and formulate the problem as an online problem over the network. For each agent, the corresponding task model parameters are represented as a task-specific sparse combination of atoms of its local knowledge base (Kumar and Daume III 2012). Each agent seeks to learn parametric models for its series of (potentially unique) tasks. The network topology imposes communication constraints among the agents. The agents share their knowledge bases with their neighbors, update them to incorporate the learned knowledge representations of their neighboring agents, and come to a local consensus. We use techniques from distributed optimization to solve this global optimization problem. This allows for transferring the learned local knowledge bases without sharing the specific learned model parameters among neighboring agents.

Problem Formulation

We consider a network of N collaborating LML agent, where each agent receives a series of sequential tasks over time, t. There is also some true underlying hidden knowledge base for all tasks, and each agent learns a local view of this knowledge base based on its own task distribution. We represent the communication mode of these agents by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of static nodes $\mathcal{V} = \{1, \ldots, N\}$ denotes the agents and the set of potentially dynamic edges $\mathcal{E}_t \subset \mathcal{V} \times \mathcal{V}$, with $|\mathcal{E}_t| = e_t$, specifies possibility of communication between the agents. We assume for each edge $(i, j) \in \mathcal{E}_t$, the nodes i and j are connected or they can communicate information. The neighborhood $\mathcal{N}(i)$ of node i is the set of all nodes that are connected to it. We use the graph structure to formulate an LML problem on this network. Although each agents learns its knowledge base locally, we assume local knowledge bases of neighboring nodes to be similar to encourage collaboration.

Each agent in the network is an LML agent that receives a set of T related (but different) supervised regression or classification tasks, each with training data, i.e. $\{\mathcal{Z}_i^{(t)} = (X_i^{(t)}, \boldsymbol{y}_i^{(t)})\}_{t=1}^{T}$, where $X_i^{(t)} \in \mathbb{R}^{d \times M}$ is task data and $\boldsymbol{y}_i^{(t)} \in \mathcal{Y}^M$ is the corresponding targets. The mapping from each data point $\boldsymbol{x}_m$ to the corresponding target can be parametrized as $y_m = f(\boldsymbol{x}_m; \boldsymbol{\theta}^{(t)})$, where $\boldsymbol{\theta}^{(t)} \in \mathbb{R}^d$. The goal for each agent is to learn $\boldsymbol{\theta}^{(t)}$ for all of its tasks.

To model task relations, the GO-MTL algorithm (Kumar and Daume III 2012) uses classic Empirical Risk Minimization (ERM) and formulates a problem by assuming that the task parameters for a single agent can be decomposed into a shared knowledge dictionary base $\boldsymbol{L}_i \in \mathbb{R}^{d\times u}$ to facilitate knowledge transfer and task-specific sparse coefficients $\boldsymbol{s}_i^{(t)} \in \mathbb{R}^u$, such that $\boldsymbol{\theta}_i^{(t)} = \boldsymbol{L}_i \boldsymbol{s}_i^{(t)}$. Following these notations, We formulate the following learning problem on $\mathcal{G}$:

$$\min_{\boldsymbol{L}_i} \{ \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N} \min_{\boldsymbol{s}_i^{(t)}} (\|\boldsymbol{\alpha}_i^{(t)} - \boldsymbol{L}_i \boldsymbol{s}_i^{(t)}\|^2_{\boldsymbol{\Gamma}_i^{(t)}}$$

$$\mu \|\boldsymbol{s}_i^{(t)}\|_1) + \lambda \|\boldsymbol{L}_{\boldsymbol{i}}\|^2_{\mathsf{F}} \}, \text{ s.t. } \boldsymbol{L}_i = \boldsymbol{L}_j,\ \forall (i,j) \in \mathcal{E}_t \quad , (1)$$

where $\|\boldsymbol{x}\|^2_{\boldsymbol{A}} = \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x}$, $\boldsymbol{\alpha}^{(t)} \in \mathbb{R}^d$ is the single task ridge estimator, and $\boldsymbol{\Gamma}^{(t)}$ is the Hessian of the loss function which measure the data fidelity. More details on derivation of eq. (1) are explained in (Rostami et al. 2017).

In order to deal with the dynamic nature and time-dependency of the objective (1), we restrict our investigation to static networks, i.e. $\mathcal{E}_t = \mathcal{E}$. We also assume that the agents are synchronous and at each time step t, each agent receives a task and solve the inner optimization on $\boldsymbol{s}_i^{(t)}$ locally using local task data. Then, through information exchanges during that time step, the local dictionaries are updated such that the agents reach consensus. To split the constrained objective (1) into a sequence of local unconstrained agent-level problems, we use the extended ADMM algorithm (Han and Yuan 2012). As a result, the agents can learn tasks in an online sequential manner, enabling lifelong learning, and meanwhile share their learned knowledge with the neighboring agent. We call our approach the Collective Lifelong Learning Algorithm (CoLLA).

Experimental Results

We compared our approach against: 1) Single task learning (STL), a lower-bound to measure the effect of positive transfer among the tasks, 2) ELLA (Ruvolo and Eaton 2013) as single agent LML algorithm, to demonstrate that collaboration between the agents improves overall performance, 3) offline CoLLA, as an upper-bound to our online distributed algorithm, and finally 4) GO-MTL, as an absolute upper-bound. We used computer survey dataset in our experiments (Lenk et al. 1996). The goal in this dataset is to predict the likelihood of purchasing one of 20 different computers by 190 subjects. Each subject is assumed to be a task and its ratings determines the task data points. We considered 19 agents and randomly allocated ten tasks to each. We randomly split the data for each task evenly into training and testing sets. We used root mean-squared error on the testing set to measure performance of the algorithms. We used the improvement in the initial performance on a new task due to transfer ("jumpstart") as our comparison criterion against STL. We used this metric because collaboration is most effective in initial tasks. Both ELLA and CoLLA converge to the same asymptotic solution and hence the jumpstart is an informative metric for our framework.

Method \ Structure	LT	CP	ST	RD
CoLLA	43.09	46.05	42.09	44.78
ELLA	37.99	-	-	-
Offline CoLLA	61.71	-	-	-
GO-MTL	61.81	-	-	-

Table 1: Jumpstart comparison (improvement in percentage) on the Computer Survey dataset given various graph structure: LT (linear tree), CP (complete), ST (star), RD (random)

For a connected graph, ADMM guarantees asymptotic convergence of our algorithm to the solution of centralized learning. Thus, for further investigation, we also studied the effect of the graph structure. We performed experiments on four graph structures: linear tree (LT) ($\mathcal{E} = \{(i, i+1) | 1 \le i < N\}$), star graph (ST), complete graph (CP), and random graph (RD). The star graph structure connects all agents through a central server, and the random graph was formed by randomly selected half of the edges of a complete graph. Performance results for these structures are presented in Table 1. From the table, we can conclude COLLA's effectiveness in collaboratively learning knowledge bases suitable for transfer when compared to ELLA and as expected it is upper-bounded by centralized schemes (Offline version of CoLLA and GO-MTL). The results also indicate that network structures with more edges have faster learning rate is faster (note that the complete graph structures outperforms other structures). This result signals that more communication and collaboration between the agents can increase learning speed.

We conclude that collaboration among the agents not only can lead to the asymptotic performance on the learned tasks comparable to centralized scheme, but enables the agent to learn faster using less amount of data in initial tasks. Our future plan is to investigate LML in dynamic networks with asynchronous agents to improve our framework.

References

Han, D., and Yuan, X. 2012. A note on the alternating direction method of multipliers. *Journal of Optimization Theory and Applications* 155(1):227–238.

Kumar, A., and Daume III, H. 2012. Learning task grouping and overlap in multi-task learning. In *ICML*, 1383–1390.

Lenk, P. J.; DeSarbo, W. S.; Green, P. E.; and Young, M. R. 1996. Hierarchical bayes conjoint analysis: Recovery of partworth heterogeneity from reduced experimental designs. *Marketing Science* 15(2):173–191.

Rostami, M.; Kolouri, S.; Kim, K.; and Eaton, E. 2017. Multi-agent distributed lifelong learning for collective knowledge acquisition. *arXiv preprint arXiv:1709.05412*.

Ruvolo, P., and Eaton, E. 2013. ELLA: An efficient lifelong learning algorithm. *ICML* 28:507–515.