Federated Kernelized Multi-Task Learning

Sebastian Caldas
Carnegie Mellon University
Pittsburgh, PA
scaldas@cmu.edu

Virginia Smith
Carnegie Mellon University
Pittsburgh, PA
smithv@cmu.edu

Ameet Talwalkar
Carnegie Mellon University and
Determined AI
Pittsburgh, PA
talwalkar@cmu.edu

ABSTRACT

Federated learning poses new statistical and systems challenges in the training of machine learning models over distributed networks of devices. In this ongoing work, we develop a state of the art MTL federated system that bypasses the modelling limitations of previous efforts through the inclusion of non-linear mappings in its formulation. We address the new issues that arise due to this inclusion and that are associated with the particulars of the federated scenario, such as communication and storage costs, introducing this way the first fully practical kernelized federated framework.

1 INTRODUCTION

Mobile phones, wearable devices, and smart homes are just a few of the modern distributed networks generating massive amounts of data each day. Due to the growing storage and computational power of devices in these networks, it is increasingly attractive to store data locally and push more network computation to the edge. Training statistical models directly on these devices is explored by the burgeoning field of federated learning [12].

Learning in federated settings is fundamentally different than in traditional distributed learning environments, requiring novel machine learning methods that can simultaneously handle the following challenges: (i) statistical: data is collected in a *non-IID* manner across the network, with data on each device being generated by a distinct distribution; (ii) systems: extreme communication bottlenecks, different computational capacities across nodes, stragglers and dropped nodes; and (iii) deployment: the known 'cold-start' and 'concept drift' phenomena.

Recent work has been successful in tackling the above statistical and systems challenges through a multi-task learning (MTL) framework, learning separate but related models for each node [15]. Nonetheless, this work is restricted to linear MTL formulations, which earlier efforts have shown to have limited expressive power [13, 17]. The objective of this ongoing work is thus to develop a novel federated approach that, through the inclusion of kernels, can capture non-linear relationships both in the local models and in the relationships among them. Including this new class of models comes with its own set of challenges such as storage costs, privacy concerns and new communication overheads, which we address in order to present the first fully practical kernelized federated system.

2 RELATED WORK

In federated learning, the aim is to learn a model over data that resides on, and has been generated by, m distributed nodes, where each node $t \in [m]$ may generate data at a different pace and from a distinct distribution. It is then natural to fit separate models to the distributed data—one for each local dataset. However, structure between models frequently exists, and modeling these relationships via MTL is a natural strategy to improve performance and boost the effective sample size for each node [1, 2, 4].

Recently, Mocha has developed a federated framework that leverages MTL as a statistical modeling choice while also exploring some of the systems challenges associated with federated learning, providing both convergence guarantees and insight into practical performance [15]. Unfortunately, among other limitations, Mocha is restricted to a simple model family (regularized linear models) and is thus short of becoming a fully practical MTL federated system.

Finally, previous works have either argued for [10, 17] or empirically demonstrated [13] the benefits of introducing non-linear mappings in their MTL formulations. These works, however, either don't deal with the distributed scenario [13, 17], or consider unfeasible the distributed computation of the kernel matrix, opting instead for explicit or approximate feature mappings [10]. Furthermore, previous work does not take into account the systems challenges that arise in the federated setting, which is one of the main contributions of this work.

3 FEDERATED KERNELIZED MULTI-TASK LEARNING

Given data $\mathbf{X}_t \in \mathbb{R}^{d \times n_t}$ from m nodes, multi-task learning fits separate weight vectors $\mathbf{w}_t \in \mathbb{R}^d$ to the data for each task (node) through arbitrary convex loss functions ℓ_t. Many MTL problems can be captured via the following general formulation:

$$\min_{\mathbf{W}, \Omega} \left\{ \sum_{t=1}^{m} \frac{1}{n_t} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t^T \phi(\mathbf{x}_t^i), y_t^i) + \mathcal{R}(\mathbf{W}, \mathbf{\Omega}) \right\}, \tag{1}$$

where $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_m] \in \mathbb{R}^{d \times m}$ is a matrix whose t-th column is the weight vector for the t-th task, $\phi(\cdot)$ is a feature mapping that can be non-linear, and the matrix $\mathbf{\Omega} \in \mathbb{R}^{m \times m}$ models relationships amongst tasks. MTL problems differ based on their assumptions on $\mathcal{R}$, which takes $\mathbf{\Omega}$ as input and promotes some suitable structure amongst the tasks. For this work, we will focus on the following popular bi-convex formulation:

$$\mathcal{R}(\mathbf{W}, \mathbf{\Omega}) = \frac{\lambda}{2} \operatorname{tr}\left(\mathbf{W}\mathbf{\Omega}\mathbf{W}^T\right), \tag{2}$$

with constant λ and constraints $\mathbf{\Omega}^{-1} \succeq 0, \operatorname{tr}(\mathbf{\Omega}^{-1}) = 1$. This formulation learns the relationships between tasks [5, 7, 17, 18] while also performing L_2 regularization on each local model. We use it because it allows for kernelizable solutions for both $\mathbf{W}$ and $\mathbf{\Omega}$ (Sections 3.1 and 3.2).

SYSML'18, February 2018, Stanford, CA USA

In order to solve the linear version of (1), existing methods use an alternating optimization procedure [10, 15, 17]. The first step fixes Ω and updates $\mathbf{W}$ in a distributed fashion, as data is horizontally split across m nodes. Meanwhile, the second fixes $\mathbf{W}$ and optimizes for Ω, which can be done centrally as Ω does not depend on the data.

3.1 Federated Update of W

To update $\mathbf{W}$ in the federated setting, we extend from previous work [8, 10, 11, 15] in order to exploit the dual formulation of (1) and separate the global problem into appropriate federated subproblems. For our particular choice of $\mathcal{R}(\mathbf{W}, \Omega)$ (2), we have the dual form

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \left\{ \frac{1}{2\lambda} \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} + \sum_{t=1}^{m} \frac{1}{n_t} \sum_{i=1}^{n_t} \ell_t^*(-\alpha_t^i) \right\} , \tag{3}$$

where $n = \sum_{t=1}^m n_t$, ℓ_t^* is the convex conjugate of ℓ_t, $\boldsymbol{\alpha}$ is the concatenation of the dual variables for all m tasks and $\mathbf{K}_{\mathrm{ix}(t,i),\mathrm{ix}(u,j)} = \frac{\sigma_{tu}}{n_t n_u} \langle \phi(\mathbf{x}_t^i), \phi(\mathbf{x}_u^j) \rangle$, where we defined $\mathrm{ix}(t,i) = i + \sum_{u=1}^t n_u$ and $\sigma_{tu} = \Omega_{tu}^{-1}$. It is clear then that we can perform the 'kernel-trick' by replacing $\langle \phi(\mathbf{x}_t^i), \phi(\mathbf{x}_u^j) \rangle$ with the kernel $k(\mathbf{x}_t^i, \mathbf{x}_u^j)$.

Subproblems. In order to solve (3) across distributed nodes, we define subproblems to be solved locally by each node. The t-th subproblem, which finds update $\Delta\boldsymbol{\alpha}_{[t]} \in \mathbb{R}^{n_t}$ to the dual variables in $\boldsymbol{\alpha}$ corresponding to node t, is given by:

$$\min_{\Delta\boldsymbol{\alpha}_{[t]}} \rho \Delta\boldsymbol{\alpha}_{[t]}^T \mathbf{K}_{[t],[t]} \Delta\boldsymbol{\alpha}_{[t]} + \sum_{i=1}^{n_t} \bar{\ell}_t(\Delta\alpha_t^i) , \tag{4}$$

where ρ is a constant that accounts for the degree of separability of $\mathbf{K}$ when we block-approximate it [15, 16] and $\bar{\ell}_t(\Delta\alpha_t^i) = n_t^{-1} \ell_t^*(\alpha_t^i + \Delta\alpha_t^i) + \lambda^{-1} (\mathbf{K}\boldsymbol{\alpha})_{[t]} \Delta\alpha_t^i$. These subproblems should only require access to data which is available locally, yet (4) depends on $\mathbf{K}$ (particularly, on the product $(\mathbf{K}\Delta\boldsymbol{\alpha})_{[t]}$), which requires access to the data in all the nodes. Nevertheless, communicating and storing the totality of the data in all the nodes is unacceptable in the federated setting.

Our proposed solution to this issue is to leverage Mocha with the recently proposed Parallel Block Minimization (PBM) algorithm [6] as shown in Algorithm 1. PBM only communicates local support vectors in each iteration, which have been found in practice to usually be orders of magnitude less than the total number of vectors for appropriate loss functions. As such, this solution is a successful federated learning strategy under the assumption of sparsity and if we allow the communication of information among nodes. However, the number of support vectors is indeed problem-dependent, leaving the need for an alternative that is guaranteed to be communication-efficient in the worst case.

An ongoing line of work that addresses this issues is the use of a low-rank approximation of the kernel matrix $\mathbf{K}$. Succinctly, a rank-l approximation of $\mathbf{K}$ could be constructed via the Nyström method [9] with l cluster centers of the original columns (which could be done once centrally). Given a small enough l, the approximation, which would take only $O(l^2 + nl)$ space, could be communicated and stored efficiently, and Mocha could be run from that point forward. Even more, it would possible to adapt this clustering in order to address the privacy concerns that inevitably come up when sharing users' data [3], and which our current PBM-based solution does not take into account.

Algorithm 1 Kernelized Federated Multi-Task Learning

```
Input: Data $\mathbf{X}_t$ for $t = 1, \ldots, m$, stored on one of $m$ nodes, and initial matrix $\Omega_0$
Starting point $(\mathbf{K}\Delta\boldsymbol{\alpha})_{[t]} := \mathbf{0} \in \mathbb{R}^{n_t}$ for $t = 1, \ldots, m$
for iterations $i = 0, 1, \ldots$ do
    Set $\rho := \max_t \sum_{u=1}^m \frac{|\sigma_{tu}|}{\sigma_{tt}}$
    Set number of federated iterations $H_i$
    Communicate $\sigma_{tt}$ to node $t = 1, \ldots, m$
    for iterations $h = 0, 1, \cdots, H_i$ do
        for nodes $t \in \{1, 2, \ldots, m\}$ in parallel do
            Solve (4) locally
            Broadcast $\{\mathbf{x}_t^i, \Delta\boldsymbol{\alpha}_t^i | \Delta\boldsymbol{\alpha}_t^i \neq 0\}$
            Compute $\mathbf{K}_{[t],:}\Delta\boldsymbol{\alpha}_{[t]}$ in parallel
            REDUCE_SCATTER to obtain $(\mathbf{K}\Delta\boldsymbol{\alpha})_{[t]}$
            Update $\boldsymbol{\alpha}_{[t]} \leftarrow \boldsymbol{\alpha}_{[t]} + \Delta\boldsymbol{\alpha}_{[t]}$
            Update $(\mathbf{K}\boldsymbol{\alpha})_{[t]} \leftarrow (\mathbf{K}\boldsymbol{\alpha})_{[t]} + (\mathbf{K}\Delta\boldsymbol{\alpha})_{[t]}$
    Update $\Omega$ centrally based on latest $\boldsymbol{\alpha}$ (Section 3.2)
return: $\Omega, \boldsymbol{\alpha}$
```

3.2 Central Update of Ω

With $\mathbf{W}$ fixed and our choice of $\mathcal{R}(\mathbf{W}, \Omega)$ (2), an analytical solution for Ω^{-1} [17] is,

$$\Omega^{-1} = \frac{(\mathbf{W}^T\mathbf{W})^{\frac{1}{2}}}{\mathrm{tr}\Big((\mathbf{W}^T\mathbf{W})^{\frac{1}{2}}\Big)} , \tag{5}$$

$$(\mathbf{W}^T\mathbf{W})_{pq} = \frac{1}{\lambda^2} \sum_{t,i} \sum_{u,j} \alpha_t^i \alpha_u^j \mathbf{K}_{\mathrm{ix}(t,i),\mathrm{ix}(u,j)} \frac{\sigma_{pt}\sigma_{qu}}{\sigma_{tu}} . \tag{6}$$

With this formulation, the tasks' clustering can also be performed in the high dimensional space specified by the kernel. A kernelized form may not exist for other formulations of $\mathcal{R}(\mathbf{W}, \Omega)$, which could be solved by using an explicit mapping $\phi(\cdot)$ (e.g. approximating infinite dimensional kernels via random features [14] at the expense of some accuracy) and updating $\mathbf{w}_t = \frac{1}{\lambda} \sum_{u=1}^m \sum_{i=1}^{n_u} \frac{\alpha_u^i}{n_u} \phi(\mathbf{x}_u^i) \sigma_{tu}$.

4 CONCLUSIONS AND FUTURE WORK

We have discussed a novel federated MTL system that captures a larger family of more expressive, non-linear models than the current state of the art [15], and have discussed how to make it robust to the systems challenges of this particular scenario. These are necessary steps towards this work's ultimate goal: to introduce the first fully practical kernelized federated framework.

Now, beyond providing a method that is theoretically sound and based on proven algorithms, a crucial part of this work is to test the framework in practice. To do this, we are currently performing extensive experimentation on a variety of real-world federated datasets. These experiments will, ultimately, also provide a much needed set of benchmarks for this field.

REFERENCES

[1] Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research* 6 (2005), 1817–1853.

[2] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2008. Convex multi-task feature learning. *Machine Learning* 73, 3 (2008), 243–272.

[3] Maria-Florina Balcan, Travis Dick, Yingyu Liang, Wenlong Mou, and Hongyang Zhang. 2017. Differentially Private Clustering in High-Dimensional Euclidean Spaces. In *International Conference on Machine Learning*. 322–331.

[4] Rich Caruana. 1997. Multitask learning. *Machine Learning* 28 (1997), 41–75.

[5] Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi-task learning. In *Conference on Knowledge Discovery and Data Mining*.

[6] Cho-Jui Hsieh, Si Si, and Inderjit S Dhillon. 2017. Communication-Efficient Distributed Block Minimization for Nonlinear Kernel Machines. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 245–254.

[7] Laurent Jacob, Jean-philippe Vert, and Francis R Bach. 2009. Clustered multi-task learning: A convex formulation. In *Neural Information Processing Systems*.

[8] Martin Jaggi, Virginia Smith, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I Jordan. 2014. Communication-Efficient Distributed Dual Coordinate Ascent. In *Neural Information Processing Systems*.

[9] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. 2012. Sampling methods for the Nyström method. *Journal of Machine Learning Research* 13, Apr (2012), 981–1006.

[10] Sulin Liu, Sinno Jialin Pan, and Qirong Ho. 2017. Distributed Multi-task Relationship Learning. *Conference on Knowledge Discovery and Data Mining* (2017).

[11] Chenxin Ma, Virginia Smith, Martin Jaggi, Michael I. Jordan, Peter Richtárik, and Martin Takáč. 2015. Adding vs. Averaging in Distributed Primal-Dual Optimization. In *International Conference on Machine Learning*.

[12] H Brendan McMahan and Daniel Ramage. 2017. http://www.googblogs.com/federated-learning-collaborative-machine-learning-without-centralized-training-data/. *Google* (2017).

[13] Keerthiram Murugesan and Jaime Carbonell. 2017. Multi-Task Multiple Kernel Relationship Learning. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 687–695.

[14] Ali Rahimi and Benjamin Recht. 2008. Random features for large-scale kernel machines. In *Advances in neural information processing systems*. 1177–1184.

[15] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. 2017. Federated Multi-Task Learning. *arXiv preprint arXiv:1705.10467* (2017).

[16] Virginia Smith, Simone Forte, Chenxin Ma, Martin Takac, Michael I Jordan, and Martin Jaggi. 2016. CoCoA: A General Framework for Communication-Efficient Distributed Optimization. *arXiv preprint arXiv:1611.02189* (2016).

[17] Yu Zhang and Dit-Yan Yeung. 2010. A Convex Formulation for Learning Task Relationships in Multi-task Learning. In *Conference on Uncertainty in Artificial Intelligence*.

[18] Jiayu Zhou, Jianhui Chen, and Jieping Ye. 2011. Clustered multi-task learning via alternating structure optimization. In *Neural Information Processing Systems*.