Federated Kernelized Multi-Task Learning

Sebastian Caldas
Carnegie Mellon University

Virginia Smith
Carnegie Mellon University

Ameet Talwalkar
Carnegie Mellon University
Determined AI

Federated Learning is the effort of training machine learning models over distributed networks of devices, pushing computation to the edge and enabling data to remain local.

Challenges:

Statistical
- Non IID
- Unbalanced data

Systems
- Distributed
- Heterogeneity

Privacy
- Local data
- Secure messages

Current frameworks, however, either do not address all of the systems challenges of the federated scenario or are limited in their expressive power.

Through Kernelized Multi-Task Learning (KMTL) each device can locally learn its own non-linear model, and improve it through a global structure learned centrally.

$$\min_{\mathbf{W},\boldsymbol{\Omega}} \sum_{t=1}^{m} \frac{1}{n_t} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t^T \phi(\mathbf{x}_t^i), y_t^i) + \frac{\lambda}{2}\,\mathrm{tr}\left(\mathbf{W}\boldsymbol{\Omega}\mathbf{W}^T\right)$$

$$\boldsymbol{\Omega}^{-1} \succeq 0, \mathrm{tr}(\boldsymbol{\Omega}^{-1}) = 1$$

$\boldsymbol{\Omega}$ captures the underlying structure among tasks

This way, we tackle the statistical challenges that other solutions fail to address:

Global model

- Usual solution
- Assumes IID

Local models

- Fails to exploit structure
- Overexerts nodes

We augment KMTL in order to address the systems and privacy challenges of the federated setting.

Kapuccino solves for $\mathbf{W}$ and $\boldsymbol{\Omega}$ in an alternating fashion:

$\mathbf{W}$ is solved distributedly by exploiting the dual formulation of KMTL and designing subproblems for each node to solve.

dual: $\min_{\boldsymbol{\alpha}} \frac{1}{2\lambda}\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} + \sum_{t=1}^{m}\frac{1}{n_t}\sum_{i=1}^{n_t}\ell_t^*(-\alpha_t^i)$

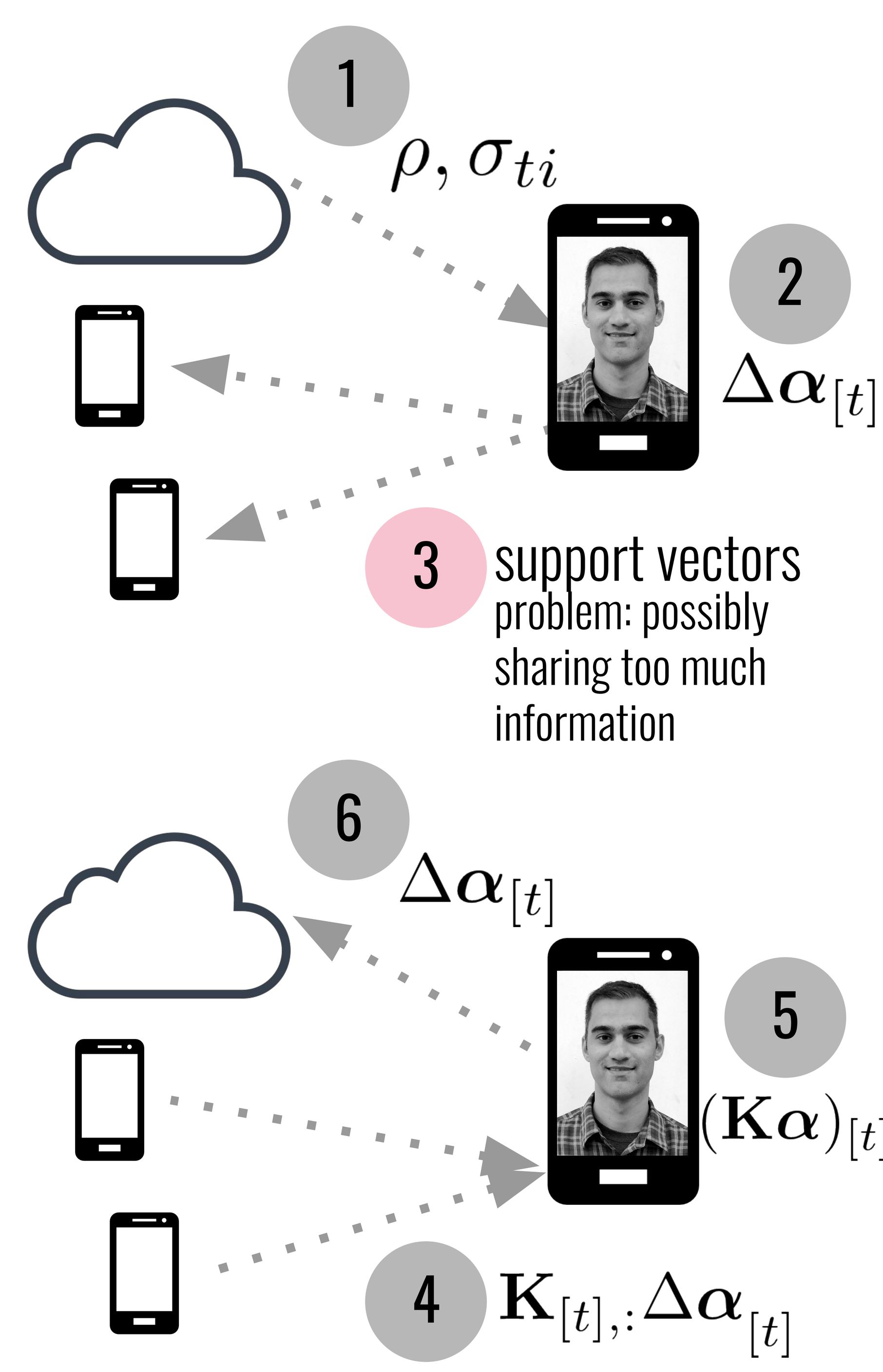
subproblems:

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

ρ captures the degree of separability of $\mathbf{K}$

dependency on $\mathbf{K}$!

"Datacenter" solution



Federated solution (in progress)

Distributedly (and privately) calculate centroids of $\mathbf{K}$'s columns

Centrally construct a low-rank approximation of $\mathbf{K}$ using the Nÿstrom method

Communicate this approximation to the different nodes

$\boldsymbol{\Omega}$, which in this case has a closed-form solution, is updated centrally based on the local updates $\Delta\boldsymbol{\alpha}_{[t]}$.

Preliminary results show that MTL outperforms global and local solutions. An even greater improvement is expected with the introduction of KMTL at the expense of higher communication and storage costs.

Solution	Communication cost	Storage cost
Mocha/MTL	O(n)	O(n_t)
Kapuccino/Datacenter	O(# supp vectors + n)	O(n_t)
Kapuccino/Federated	O(n)	O(rank^2 + n · rank)

Communication and storage costs per $\mathbf{W}$ update for different frameworks.

References

Smith, et al. Federated Multi-Task Learning. NIPS, 2017.

Liu, et al. Distributed Multi-task Relationship Learning. Conference on Knowledge Discovery and Data Mining, 2017.

Hsieh, et al. Communication-Efficient Distributed Block Minimization for Nonlinear Kernel Machines. Conference on Knowledge Discovery and Data Mining, 2017.