



Multitask Learning over Graphs

Brad Baker
Riyasat Ohib

Introduction: MTL over Graphs



- Deals with the problem of simultaneously learning several related tasks.
- MTL is an approach to inductive transfer learning.
- This helps improve generalization performance relative to learning each task separately by using the domain information contained in the training signals of related tasks as an inductive bias.



Introduction

1. Previous work usually focuses on the assumption that all data are available beforehand at a fusion center.
2. Whereas this paper deals with learning multiple tasks from streaming data over distributed (or networked) systems.
3. The working hypothesis for these strategies is that agents are allowed to cooperate with each other to learn distinct, though related, tasks
4. It also explains how and when cooperation over multitask networks outperform non-cooperative strategies.



Multitask Network Models

Multi-task Network Models

- Network of a collection of N autonomous agents, connected through a topology.
- Neighborhood of agent k is N_k , these are connected to k by an edge.
- A real-valued, strongly convex, and differentiable cost $J_k(w_k)$ is associated with each agent k .
- Objective/task at agent k is to estimate parameter vector w_k^0 of size $M_k \times 1$ that minimizes $J_k(w_k)$.

$$w_k^0 = \operatorname{argmin}_{w_k} J_k(w_k)$$

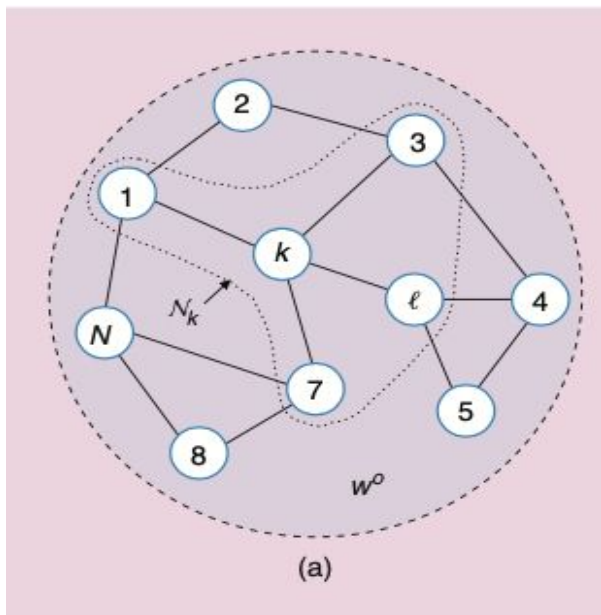


Relations of the minimizers across the Agents

Depending on how the minimizers across the agents relate to each other the paper distinguishes between three categories of networks:

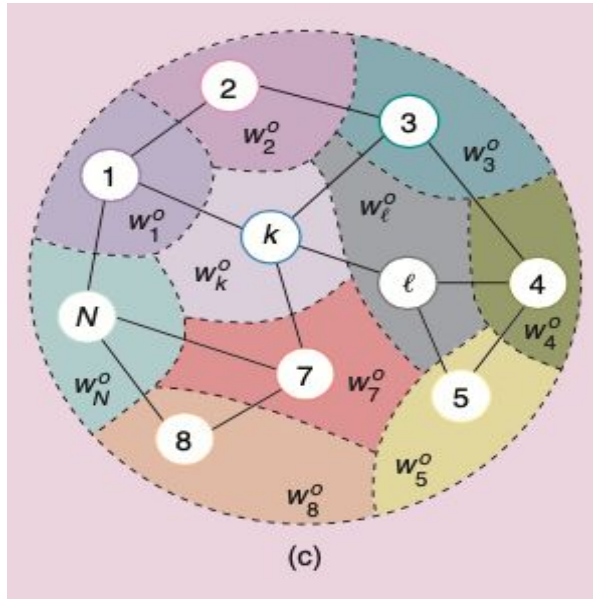
1. Single-task network.
2. Clustered multi-task network
3. Multitask network

Single-Task Network



All costs $J_k(w_k)$ are minimized at the same location w^0 , particularly, $w_k^0 = w_o$ for all k .

Multi-task Network



The individual costs are minimized at distinct, though related, locations $\{w_k^o\}$



Minimizing the Cost

$$w_k^0 = \operatorname{argmin}_{w_k} J_k(w_k)$$

- Each agent k can minimize the cost on its own. However, since the objectives across the network relate to each other, it is expected that by properly promoting these relationships, one may improve the network performance.
- One important question is how to design cooperative strategies that can lead to better performance versus non cooperative methods, where each agent attempts to determine its parameters on its own.



Noncooperative learning under streaming data

- The agents being considered operate in the streaming data setting. That is, each agent k receives at time instance i , one realization $x_{k,i}$ of a random data x_k .
- Goal of the agent k is to estimate the vector w_k^0 that minimizes its risk function:

$$J_k(w_k) = \mathbb{E}_{x_k} Q_k(w_k; x_k)$$

- defined in terms of some loss function $Q_k(\cdot)$
- The expectation is computed over the distribution of the data x_k .
- However, in the stochastic setting where the agent will operate, the distribution of the data is generally unknown.



Noncooperative learning under streaming data

- This means that the risks $J_k(\cdot)$ and their gradients $\nabla_{w_k} J_k(\cdot)$ are unknown.
- Approximate gradient vectors $\widehat{\nabla_{w_k} J_k}(\cdot)$ is used.
- The resulting stochastic gradient descent algorithm:

$$w_{k,i} = w_{k,i-1} - \mu \widehat{\nabla_{w_k} J_k}(w_{k,i-1})$$

where $w_{k,i}$ is the estimate of w_k^o at iteration I and $\mu > 0$ is the small step-size parameter/learning rate.

- The Gradient approximation at the i -th iteration is

$$\widehat{\nabla_{w_k} J_k}(w_{k,i}) = \nabla_{w_k} Q_{w_k}(w_k; x_{k,i})$$

Example with Logistic Regression

- $\gamma_k(i) \pm 1$ be a streaming sequence of Binary class variables.
- $\mathbf{h}_{k,i}$ is the streaming sequence of $M_k \times 1$ real random feature vector.
- In these problems, agent k seeks to estimate the vector w_k^0 that minimizes the regularized logistic risk function:

- $$J_k(w_k) = \mathbb{E} \ln(1 + \exp(-\gamma_k(i) \mathbf{h}_{k,i}^T w_k)) + \frac{\rho}{2} \|w_k\|^2$$

- Here, $\rho > 0$ is the regularization parameter.
- Once, w_k^0 is found, then $\gamma_k(i) \pm 1 = \text{sign}(\mathbf{h}_{k,i}^T w_k)$ can be used as the decision rule to classify new features.
- The stochastic gradient algorithm for this:

$$w_{k,i} = (1 - \mu\rho)w_{k,i-1} + \mu\gamma_k(i)\mathbf{h}_{k,i} \left(\frac{1}{1 + \exp(-\gamma_k(i)\mathbf{h}_{k,i}^T w_k)} \right)$$



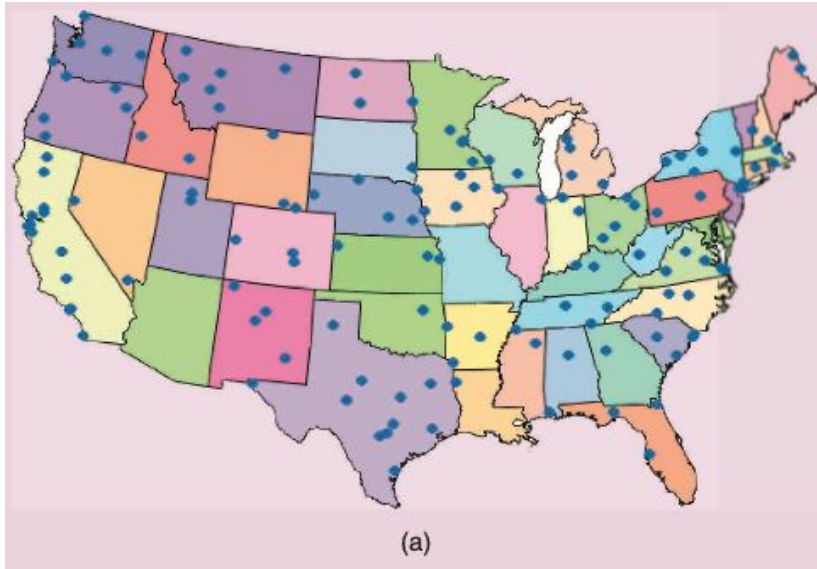
Multitask-learning framework

- Let $\mathcal{W} \triangleq \text{col}\{w_1, \dots, w_N\}$ denote the collection of parameter vectors from across the network.
- The following global optimization problem for the multitask formulation:

$$\mathcal{W}^* = \arg \min_{\mathcal{W}} J^{glob}(\mathcal{W}) = \sum_{k=1}^N J_k(W_k) + \frac{\eta}{2} \mathcal{R}(\mathcal{W})$$

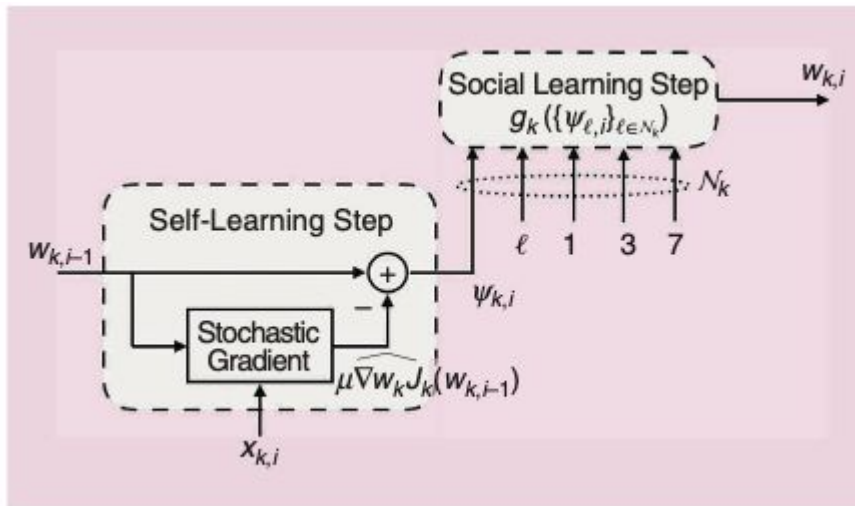
- subject to $\mathcal{W} \in \Omega$
- Here \mathcal{R} is a convex regularization function promoting the relationship between tasks, Ω is a closed convex set defining the feasible region of the parameter vectors, and $\eta > 0$ controls the importance of the regularization.
- The choice of the regularizer $\mathcal{R}(\cdot)$ and the set Ω depends on prior information about how the multitask models relate to each other.

Example 1: Weather Forecasting



- Agents: $N = 139$ Weather Stations.
- Feature vector $h_{k,i}$ of collected data (might be temp, wind speed, dew point etc.) at sensor k at day i .
- $\gamma_k(i)$ denotes binary variable, if rain occurs $\gamma_k(i) = 1$, else $\gamma_k(i) = -1$.
- Laplacian regularizer $S(\mathcal{W})$.
- By choosing $\mathcal{R}(\mathcal{W}) = S(\mathcal{W})$ and $\Omega = \mathbb{R}^{MN}$.
- This multitask formulation for the weather forecasting application takes into account the smoothness prior over the graph.

Multitask Strategies



$$\psi_{k,i} = w_{k,i-1} - \mu \widehat{\nabla}_{w_k} J_k(w_{k,i-1})$$

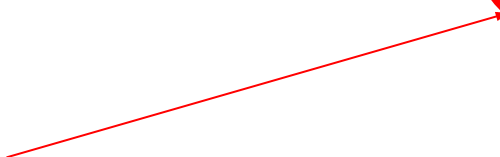
$$w_{k,i} = g_k(\{\psi_{\ell,i}\}_{\ell \in N_k}).$$



Regularized Multitask Estimation

$$w^* = \arg \min_w J^{\text{glob}}(w) = \sum_{k=1}^N J_k(w_k) + \frac{\eta}{2} \mathcal{R}(w)$$

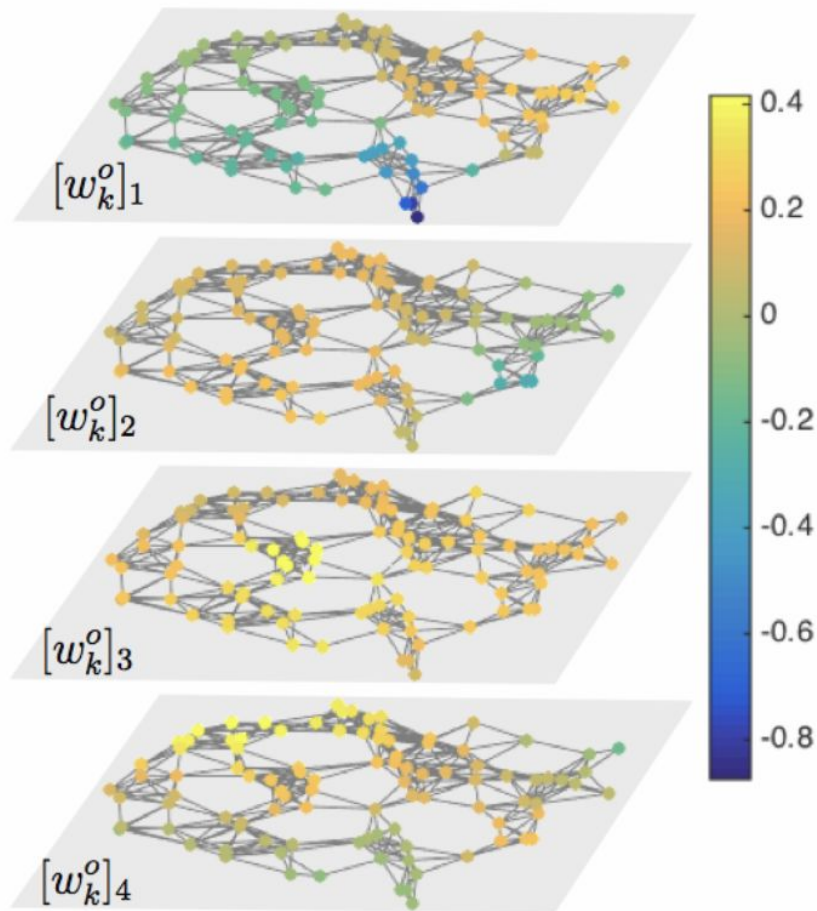
Idea: Incorporating
information about graph and
task-relatedness into the
regularization term



Multitask Learning under Smoothness

$$S(w) = w^T \mathcal{L} w = \frac{1}{2} \sum_{k=1}^N \sum_{l \in N_k} c_{kl} ||w_k - w_l||^2$$

Penalty for
distance from
neighbors



Multitask Learning under Smoothness

$$w_{k,i} = \psi_{k,i} - \mu\eta \sum_{l \in \mathcal{N}_k} c_{kl} (\psi_{k,i} - \psi_{l,i})$$

See: performance result 2 -- basically, the smoothness constraint allows us to constrain the variance during learning without a huge increase in bias.

Figure from Nassif et al. 2018

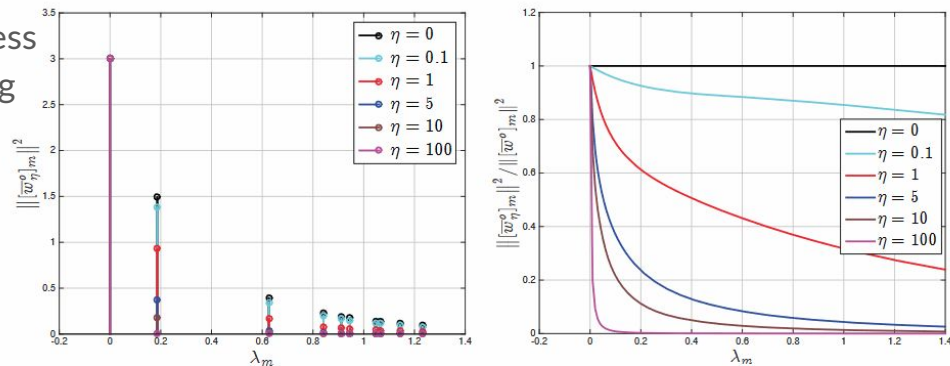


Fig. 1. MSE networks ($R_{u,k} = R_u \forall k$). (Left) Graph frequency content of \bar{w}_η^o . (Right) The ratio $\|[\bar{w}_\eta^o]_m\|^2 / \|[\bar{w}^o]_m\|^2$ for $\lambda_m \in [0, 1.4]$ from (28).

Graph Spectral Regularization

Idea: directly constrain the spectrum of the graph

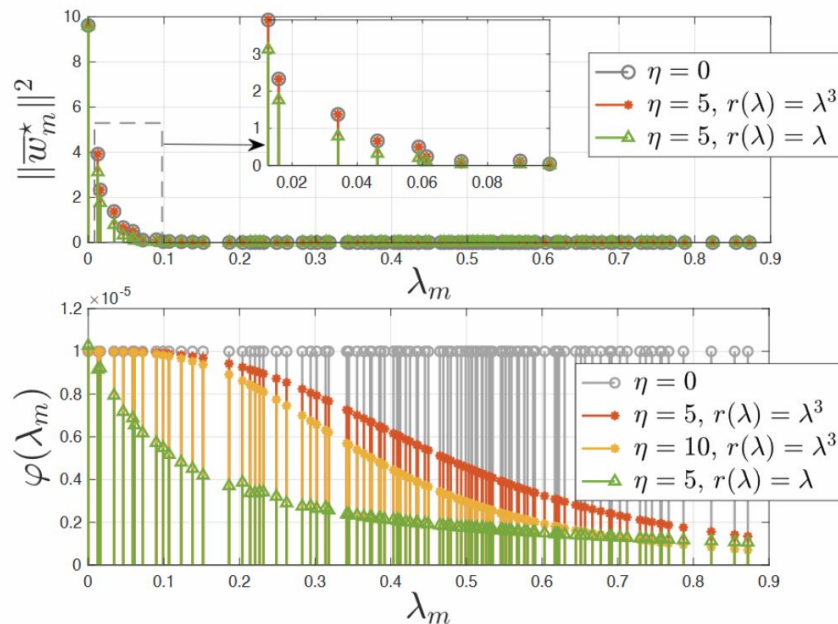
$$\mathcal{R}(w) = w^\top r(\mathcal{L})w = w^\top (r(L) \otimes I_M)w$$

Regularization on
graph eigenvalues

$$\begin{cases} \psi_{k,i}^s = \beta_{S-s} \psi_{k,i} + \sum_{\ell \in \mathcal{N}_k} c_{k\ell} (\psi_{k,i}^{s-1} - \psi_{\ell,i}^{s-1}), & s = 1, \dots, S \\ w_{k,i} = \psi_{k,i} - \mu \eta \psi_{k,i}^S \end{cases}$$

Engineering choice:
How do we choose r ?

$$r(\lambda) = \sum_{s=0}^S \beta_s \lambda^s$$





Non-Quadratic Regularization

Subspace Constraints

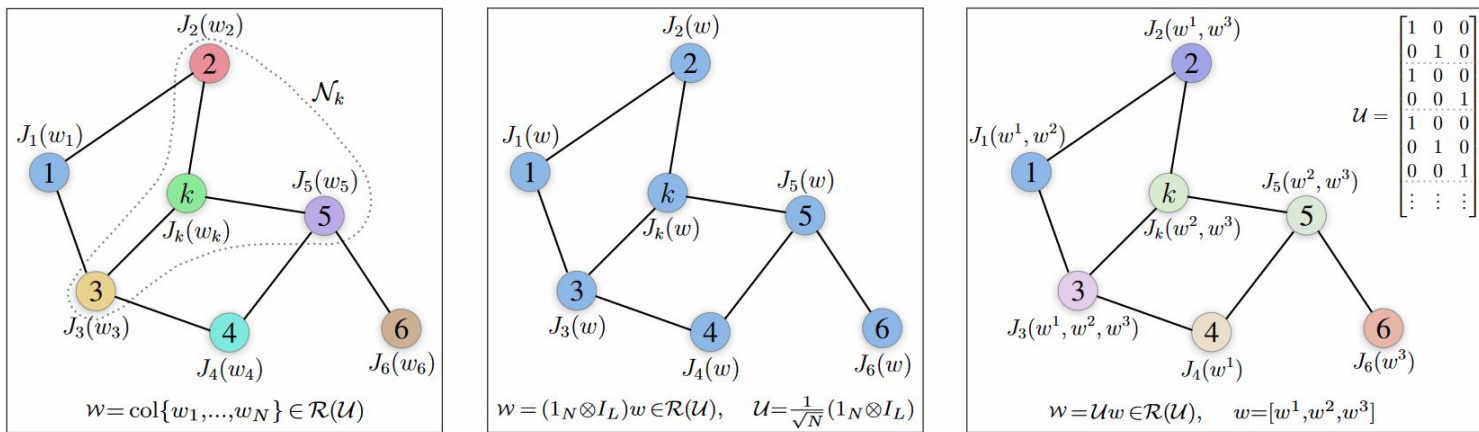


Fig. 1. Inference under subspace constraints. (Left) Illustrative scheme of problem (2): Each agent k in the network has an individual w_k to estimate, subject to subspace constraints that enforce the objectives across the network to lie in $\mathcal{R}(\mathcal{U})$. (Middle) Consensus optimization (1): Agents in the network seek to estimate an $L \times 1$ common vector w corresponding to the minimizer of the aggregate sum of individual costs. (Right) Coupled optimization (23): Different agents generally seek to estimate different, but overlapping, parameter vectors.

Clustered MultiTask Estimation

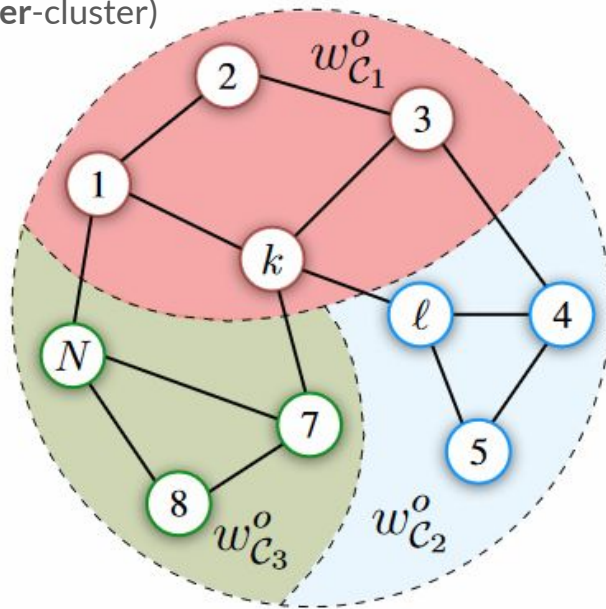
Combines subspace constraints (**intra-cluster**) with regularization (**inter-cluster**)

Basically the clusters themselves define the subspace constraints

$$\Omega = \text{Range}(\mathcal{U}), \quad \mathcal{U} = \text{diag} \left\{ \frac{1}{\sqrt{N_q}} (\mathbf{1}_{N_q} \otimes I_M) \right\}_{q=1}^Q$$

And regularization is determined by priors on inter-cluster relations

$$\mathcal{R}(\mathcal{W}) = \sum_{k=1}^N \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} \rho_{k\ell} h_{k\ell}(w_k, w_\ell)$$





Conclusion

- We can incorporate priors about task relationships into the learning mechanism
- Different priors invoke different strategies
- Regularization controls relationships between tasks
- Subspace constraints model overlapping parameters, common estimated subspaces
- Clustered multitask estimation combines the two - shared subspaces **within** clusters, and regularization **between** clusters