Clustered Multi-Task Learning Via Alternating Structure Optimization: Supplemental Material

A. Algorithm Details

Alternating Optimization Method

Algorithm 1 Alternating Optimization Algorithm (altCMTL) for Convex CMTL

Input: W_0, $\gamma_0 \in \mathbb{R}$, and max iteration number q.
Output: M, W.
for $i = 1$ to q **do**
    Update M_i by solving Eq. (20).
    Update W_i by solving Eq. (19).
    if stopping criteria satisfied **then** break the for loop.
end-for
Set $M = M_{i+1}, W = W_{i+1}$.

Accelerated Projected Gradient Method

Algorithm 2 Accelerated Projected Gradient Algorithm (apgCMTL) for Convex CMTL

Input: Z_0, $\gamma_0 \in \mathbb{R}$, and max iteration number q.
Output: Z.
Set $Z_1 = Z_0$, $t_{-1} = 0$, and $t_0 = 1$.
for $i = 1$ to q **do**
    Set $\mu_i = (t_{i-2} - 1)/t_{i-1}$, $S_i = Z_i + \mu_i(Z_i - Z_{i-1})$.
    while (true)
        Compute $Z^* = \operatorname{argmin}_{Z \in \mathcal{C}} \mathcal{M}_{\gamma,S}(Z)$.
        if $f(Z^*) \leq \mathcal{M}_{\gamma,S_i}(Z^*)$ **then** break the while loop
            else set $\gamma_i = \gamma_i \times 2$.
        end-if
    end-while
    Set $Z_{i+1} = Z^*$ and $\gamma_{i+1} = \gamma_i$.
    if stopping criteria satisfied **then** break the for loop.
    Set $t_i = \frac{1+\sqrt{1+4t_{i-1}^2}}{2}$.
end-for
Set $Z = Z_{i+1}$.

The optimization problem in Eq. (23) admits an analytical solution via solving a simple convex projection problem. The main result is summarized in the following theorem.

Theorem 6.1. *Given an arbitrary symmetric matrix $\hat{M}_S \in \mathbb{R}^{m\times m}$ in Eq. (23), let $\hat{M}_S = P\hat{\Sigma}P^T$ be its eigen-decomposition, where $P \in \mathbb{R}^{m\times m}$ is orthogonal, and $\hat{\Sigma} = diag(\hat{\sigma}_1, \cdots, \hat{\sigma}_m) \in \mathbb{R}^{m\times m}$ is diagonal with the eigenvalues on its main diagonal. Let $\Sigma^* = diag(\sigma_1^*, \cdots, \sigma_m^*) \in \mathbb{R}^{m\times m}$, where $\{\sigma_i^*\}_{i=1}^m$ is the optimal solution to the following optimization problem:*

$$\min_{\{\sigma_i\}} \sum_{i=1}^{m} (\sigma_i - \hat{\sigma}_i)^2, \quad \textit{s.t.} \ \sum_{i=1}^{m} \sigma_i = k,\ 0 \leq \sigma_i \leq 1,\ i = 1, \cdots, m. \tag{25}$$

Then the global minimizer to Eq. (23) is given by $M^* = P\Sigma^* P^T$.

To prove the above theorem, we first introduce the following lemma:

Lemma 6.2. *Let $O_{(1)}$ be the optimal objective value of the optimization problem:*

$$\min_{T} \|T - E\|_F^2, \quad \textit{s.t.} \ \operatorname{tr}(T) = k,\ 0 \preceq T \preceq I, \tag{26}$$

and let $O_{(2)}$ be the optimal objective value of the optimization problem:

$$\min_{\hat{E}} \|\hat{E} - E\|_F^2, \quad \textit{s.t.} \quad \mathrm{tr}\left(\hat{E}\right) = k,\ \hat{E} = \mathrm{diag}(\hat{e}_1, \cdots, \hat{e}_d),\ 0 \leq \hat{e}_i \leq 1. \tag{27}$$

Then $O_{(1)} = O_{(2)}$.

Proof. According to the definition, the feasible domain of optimization (27) is a subset of the feasible domain of optimization (26), which indicates $O_{(2)} \geq O_{(1)}$. Let T^* be the optimal solution to the optimization (26), and $\hat{T} = \mathrm{diag}(T^*)$ be the diagonal matrix by setting the non-diagonal elements of T^* to 0. It is evident that $\mathrm{tr}\left(\hat{T}\right) = k$ and $0 \preceq \hat{T} \preceq I$ hold. Notice that $\hat{T}$ is a feasible point in optimization (27), we have following inequality:

$$O_{(1)} = \|T^* - E\|_F^2 \geq \|\hat{T} - E\|_F^2 \geq O_{(2)},\ O_{(1)} \geq O_{(2)}. \tag{28}$$

Therefore $O_{(1)} = O_{(2)}$ must hold. This completes the proof. □

We are now ready to prove Theorem 6.1:

Proof of Theorem 6.1. For an arbitrary M_Z feasible for Eq. (23), we denote its eigen-decomposition by $M_Z = Q\Lambda Q^T$, where $Q \in \mathbb{R}^{m\times m}$ is orthogonal, $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_m) \in \mathbb{R}^{m\times m}$ is diagonal with the eigenvalues on its main diagonal. Because of the unitary invariant property of Frobenius norm, the optimization in Eq. (23) can be equivalently represented as:

$$\begin{aligned} \min_{\Lambda, Q} \quad & \left\| P^T Q \Lambda Q^T P - \hat{\Sigma} \right\|_F^2, \\ \text{s.t.} \quad & \mathrm{tr}\,(\Lambda) = k,\ \Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_m),\ 0 \leq \lambda_i \leq 1, \quad Q^T Q = Q Q^T = I_m. \end{aligned} \tag{29}$$

According to Lemma 6.2, the optimization problems in Eq. (29) and Eq. (25) have the same optimal objective value. It is easy to verify that the optimization problem in Eq. (29) is strictly convex, and that the pair $\Lambda = \Sigma^*, Q = P$ is a feasible solution. This means that $\Lambda = \Sigma^*, Q = P$ is the unique global minimizer to Eq. (29). Thus $M^* = P\Sigma^* P^T$ is the unique global minimizer to Eq. (23). □

Direct Gradient Descent Method

Algorithm 3 Direct Gradient Descent Algorithm (graCMTL) for Convex CMTL

Input: W_0, $\gamma_0 \in \mathbb{R}$, and max iteration number q.
Output: M, W.
for $i = 1$ to q **do**
    Compute M_t^* by solving Eq. (20) using W_{t-1}.
    Compute the gradient direction $\nabla_W g_{\text{CMTL}}(W_{t-1}) = 2(\eta I + M_t^*)^{-1} W_{t-1}^T$.
    Perform a line search to determine L_t.
    $W_t = W_{t-1} + L_t \nabla_W g_{\text{CMTL}}(W_{t-1})$.
    if stopping criteria satisfied **then** break the for loop.
end-for
Set $M = M_{i+1}, W = W_{i+1}$.

B. Construction of Synthetic Cluster-Structured Data

Denote by $\widehat{w}_i^c$ the i-th task from the c-th cluster. $\widehat{w}_i^c$ can be expressed as the sum of the cluster center w^c and the task-specific component w_i^c, i.e., $\widehat{w}_i^c = w^c + w_i^c$. The cluster center w^c is generated as follows: (1) set the value of the first 20 entries in w^c as zero; (2) select $(d-20)/5$ entries from the other $d-20$ entries, and generate non-zero values from $\mathcal{N}(0, 900)$ for the selected $(d-20)/5$ entries. Note that we keep w^c orthogonal to the other cluster centers by selecting the appropriate locations of the non-zero entries. The task-specific component w_i^c is generated as follows: (1) generate non-zero values from $\mathcal{N}(0, 16)$ for the first 20 entries; (2) generate non-zero values from $\mathcal{N}(0, 16)$ for the locations corresponding to the non-zero entries of w^c. For each task we generate 60 sample pairs

(the data point and the response). Denote the data matrix and the response vector by X_i and y_i, respectively. The entries in X_i are generated from $\mathcal{N}(0,1)$, and the entries in y_i are generated as $y_i = X_i w_i^c + \xi_i$, where $\xi \sim \mathcal{N}(0, 0.1)$ represents the noise vector.

C. Effectiveness Comparison on Sarcos Dataset

The Sarcos data is collected for an inverse dynamics prediction problem for a seven degrees-of-freedom anthropomorphic robot arm. This data consists of 48933 observations corresponding to 7 joint torques; each of the observations is described by 21 features including 7 joint positions, 7 joint velocities, and 7 joint accelerations. The prediction of each joint torque corresponds to one task. Because using a few training samples already gives good performance, we vary the training sample size in the set $\{10, 20, 50, 100\}$. The results are presented in Table 2. We can observe (1) when the training sample size is relatively small, cCMTL outperforms other competing methods; (2) when the training ratio is large, cCMTL and RidgeSTL are comparable; (3) RegMTL performs poorly in all settings.

Table 2: Performance comparison on the Sarcos data in terms of nMSE and aMSE. Smaller nMSE and aMSE indicate better performance. The regularization parameters of all methods are tuned using 5-fold cross validation. The mean and standard deviation are calculated based on 10 random repetitions.

Measure	Sample	RidgeSTL	RegMTL	cCMTL
nMSE	10	2.3668 ± 0.3033	2.2310 ± 0.5008	2.1984 ± 1.0083
	20	0.7409 ± 0.1461	0.8787 ± 0.2007	0.6710 ± 0.0740
	50	0.2562 ± 0.0366	0.4387 ± 0.0509	0.2522 ± 0.0258
	100	0.1735 ± 0.0068	0.3717 ± 0.0411	0.1784 ± 0.0097
aMSE	10	0.9151 ± 0.0718	0.9399 ± 0.1622	1.0803 ± 0.6384
	20	0.4129 ± 0.0976	0.4796 ± 0.1267	0.3703 ± 0.0427
	50	0.1457 ± 0.0259	0.2581 ± 0.0374	0.1432 ± 0.0198
	100	0.0992 ± 0.0031	0.2133 ± 0.0242	0.1014 ± 0.0041