Multitask Metric Learning: Theory and Algorithm

Boyu Wang[1,3] **Hejia Zhang**[1] **Peng Liu**[2] **Zebang Shen**[3] **Joelle Pineau**[4]
[1]Princeton University [2]University of Toronto [3]University of Pennsylvania [4]McGill University

Abstract

In this paper, we study the problem of multitask metric learning (mtML). We first examine the generalization bound of the regularized mtML formulation based on the notion of algorithmic stability, proving the convergence rate of mtML and revealing the tradeoff between the tasks. Moreover, we also establish the theoretical connection between the mtML, single-task learning and pooling-task learning approaches. In addition, we present a novel boosting-based mtML (mt-BML) algorithm, which scales well with the feature dimension of the data. Finally, we also devise an efficient second-order Riemannian retraction operator which is tailored specifically to our mt-BML algorithm. It produces a low-rank solution of mtML to reduce the model complexity, and may also improve generalization performances. Extensive evaluations on several benchmark data sets verify the effectiveness of our learning algorithm.

1 Introduction

Many machine learning algorithms rely on distance metrics for an appropriate definition of similarity/dissimilarity over the input space (e.g., k-nearest neighbor (k-NN) classifier, K-means clustering). Starting from [39], distance metric learning has been actively studied as a promising approach to learning problem-specific distance metrics [38, 18, 33, 17]. More specifically, given the data points $\{x_i\}_{i=1}^n \in \mathbb{R}^d$, the objective of metric learning is to learn a Mahalanobis distance parameterized by a *symmetric positive definite* (SPD) matrix M: $d_M(x_i, x_j) = \sqrt{(x_i - x_j)^\top M(x_i - x_j)}$ such that additional constraints (e.g., pairwise similarity) are satisfied. In addition, it has been shown that by taking advantage of the idea of *multitask learning* [9], the performance of metric learning algorithms can be improved by jointly solving multiple related problems, where the underlying assumption is that some common knowledge or structure can be shared across the tasks [29].

While the empirical results presented in recent works have verified the effectiveness of the multitask metric learning (mtML), its theoretical justification is rarely investigated. The only related work we are aware of is [34], where a mtML algorithm is proposed using the idea of group lasso [42, 28], and the generalization guarantee is analyzed based on the notion of algorithmic robustness [40]. However, the theoretical analysis in [34] only reveals the benefits of sparse learning, not providing any insight into why and when mtML will improve the learning performances. As the first contribution, we theoretically analyze a widely used multitask learning strategy based on the notion of *algorithmic stability* [7]. Specifically, we first answer the question of **why mtML can work** by showing that by controlling the tradeoff between the similarity and the diversity of the model, the convergence rate of the stability coefficient of mtML can increase from $\mathcal{O}(\frac{1}{n})$ to $\mathcal{O}(\frac{1}{Tn})$, with the risk of increasing training loss, where T is the number of tasks. More important, we also try to address the question of **when mtML can work** by showing that the generalization bound of mtML is related to the learning performances of the pooling-task approach and the single-task approach on the training set, which provides a new insight into mtML.

Given the theoretical analysis, our second contribution is algorithmic: we study an instantiation of mtML and present an efficient boosting-based method that alleviates the computational issue of metric learning. In contrast to previous approaches which formulate the mtML problem as semidefinite programming and do not scale well with the dimension of the data, our method, inspired by [33], decomposes a positive semidefinite matrix as a positive linear combination of trace-one rank-one (TORO) matrices as weak learners, and simultaneously learns the metric matrices from

Proceedings of the 22[nd] International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

multiple tasks via a scalable boosting-based algorithm. As a result, each boosting step only requires the computation of the largest eigenvalue and its corresponding eigenvector, which can be solved efficiently.

The third contribution of this paper is also algorithmic, motivated by learning low-rank metric matrices, which is particularly useful for reducing the model complexity and improving the generalization performance. Although there have been some methods proposed in the literature [15, 13, 12, 19], they cannot directly applied to the setting of mtML. Inspired by recent advances in optimization over a matrix manifold [1, 35, 31], we propose a fixed-rank learning algorithm based on a novel second-order Riemannian retraction operator, which is tailored specifically to our boosting learning procedure. By leveraging the fact that each weak learner is a TORO matrix, the presented operator is efficient and ensures that the metric matrices reside on the low-rank manifolds at each boosting step.

2 Multitask Metric Learning

Let $\mathfrak{S} = \{\mathcal{S}_t\}_{t=1}^T$ be T related tasks, where $\mathcal{S}_t = \{z_i^t = (x_i^t, y_i^t)\}_{i=1}^{n_t}$ is the d-dimensional data set drown from a distribution $\mathcal{D}_t$ for the t-th task, and n_t is the number of training data points of the t-th task. For simplicity, we assume that $n_t = n, \forall t = 1, \ldots, T$. To jointly learn from multiple tasks, we investigate a widely used mtML formulation [29, 41, 6], which assumes that for the t-th task, the metric matrix can be decomposed as $M_t = H_0 + H_t$, where H_t is the task-specific metric, and H_0 is the global metric measuring the commonalities among the tasks. Consequently, the goal of mtML is to find the matrices $\{H_t\}_{t=0}^T$ which minimize the following objective function:

$$\min_{\{H_t\}_{0=1}^T} \frac{1}{T}\sum_{t=1}^T \left(\mathcal{L}_{\mathcal{S}_t}(H_0 + H_t) + \mathcal{R}_t(H_0, H_t)\right), \quad (1)$$
$$\text{s.t.} \quad H_t \in \mathbb{S}_+^d, \ \forall t = \{0, \ldots, T\}$$

where $\mathcal{L}_{\mathcal{S}_t}(H_0 + H_t) = \frac{1}{|\mathcal{C}_t|}\sum_{(z_i^t, z_j^t)\in\mathcal{C}_t} \ell\left(H_0 + H_t, z_i^t, z_j^t\right)$ is the empirical loss over the t-th task, $\ell(\cdot)$ is a loss function of the constraints (e.g., pairs, triplets), $\mathcal{C}_t$ is the set of constraints,[1] and $|\mathcal{C}_t|$ is the number of constraints. $\mathcal{R}_t$ is a regularizer to control the model complexity, and $\mathbb{S}_+^d$ denotes the set of SPD matrices defined over $\mathbb{R}^{d\times d}$. In this paper, we study the widely used Frobenius norm regularizer (e.g., [14, 18, 29, 30]). In addition, to control the model diversity, one should impose different regularization strengths on H_0 and $H_{t>0}$. To this end, we study the regularizer $\mathcal{R}_t(H_0, H_t) = \lambda_0||H_0||_{\mathsf{F}}^2 + \lambda_t||H_t||_{\mathsf{F}}^2$, where $||\cdot||_{\mathsf{F}}$ is the Frobenius norm of a matrix, λ_0 and $\lambda_{t>0}$ are the trade-off regularization parameters. If $\lambda_0 \to \infty$, (1) reduces to *single-task* (stML) approaches, which solve T tasks individually, and if $\lambda_{t>0} \to \infty$, (1) reduces to *pooling-task* (ptML) approaches, which simply treat the T tasks as a single one. For simplicity, we assume $\lambda_{t>0} = \lambda$.

[1] For simplicity, our analysis focuses on pair-based constraints, but the conclusion is also applicable to triplet-based metric learning algorithms with slightly modification of the lemmas and theorems. See the supplementary materials for more details.

3 Theory

Before we present a specific algorithm to solve (1), we first theoretically analyze the properties of mtML by adapting the notion of algorithmic stability [7, 18] to the setting of metric learning. The main advantage of exploiting algorithmic stability rather than the Rademacher complexity, the growth function, or the VC-dimension is that it can provide algorithm-dependent learning bounds and hence lead to tighter generalization guarantees [20]. We start by introducing several definitions used in our analysis.

Definition 1 (**σ-admissibility**). *A loss function $\ell(M, z, z')$ is σ-admissible with respect to M, if for any two matrices M and M', and any pairs of examples z and z' , there exists $\sigma > 0$ such that*

$$|\ell(M, z, z') - \ell(M', z, z')| \leq \sigma\left|(x - x')^\top M(x - x') - (x - x')^\top M'(x - x')\right|.$$

Definition 2 (**Uniform stability**). *(See [18], Section 3) A metric learning algorithm has β-uniform stability, with $\beta \geq 0$, if*

$$\sup_{z,z'\sim\mathcal{D}} |\ell(M_{\mathcal{S}}, z, z')) - \ell(M_{\mathcal{S}^i}, z, z')| \leq \beta, \qquad \forall \mathcal{S}, \mathcal{S}^i$$

where $\mathcal{S}^i$ is the training sample $\mathcal{S}$ with the i-th example z_i replaced by an independent and identically distributed (i.i.d.) example z_i', $M_{\mathcal{S}}$ and $M_{\mathcal{S}^i}$ are the matrices learned from $\mathcal{S}$ and $\mathcal{S}^i$ respectively.

The following lemma gives the generalization bounds of bounded β-uniformly stable metric learning algorithms. Due to space constraints we delegate all the proofs to the supplementary materials.

Lemma 1. *Let ℓ be a loss function bounded by $B \geq 0$, and let $\mathcal{S}$ be a training set of n data points drawn from a distribution $\mathcal{D}$, and $M_{\mathcal{S}}$ be the matrix learned over $\mathcal{S}$ by a β-uniformly stable metric learning algorithm. Let $\mathcal{L}_{\mathcal{D}}(M_{\mathcal{S}}) = \mathbb{E}_{z,z'\sim\mathcal{D}}[\ell(M_{\mathcal{S}}, z, z')]$ be the expected loss of $M_{\mathcal{S}}$ over $\mathcal{D}$. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the following holds:*

$$\mathcal{L}_{\mathcal{D}}(M_{\mathcal{S}}) - \mathcal{L}_{\mathcal{S}}(M_{\mathcal{S}}) \leq 2\beta + (2n\beta + 2B)\sqrt{\frac{\log\frac{1}{\delta}}{2n}}.$$

Given Lemma 1, we can derive meaningful generalization guarantees of mtML by upper bounding β and B. The following theorem shows that the mtML algorithms solving (1) is β-uniformly stable.

Theorem 1. *The mtML algorithms solving (1) is β-uniformly stable, with*

$$\beta \leq \frac{\sigma^2 R^4}{\lambda_0 T n} + \frac{\sigma^2 R^4}{\lambda n},$$

where $R = \max_{x,x'} ||x - x'||$.

Remark 1. *Theorem 1 explains why mtML can work – the stability coefficient β of mtML converges in the order of $\mathcal{O}(\frac{1}{Tn})$ as $\lambda \to \infty$. However, it results in ptML, which simply put all the tasks together and may have high training loss $\mathcal{L}(M_S)$. On the other hand, $\lambda_0 \to \infty$ leads to the stML solution with the convergence rate in the order of $\mathcal{O}(\frac{1}{n})$, and therefore there is no benefit of multitask learning. In other words, Theorem 1 formalizes the intuition that multitask learning can be viewed as a tradeoff between single-task and pooling-task learning. To obtain good generalization performances, one should the control balance between the model diversity and training loss across the tasks by tuning the regularization parameters λ_0 and λ.*

Remark 2. *When $T = 1$, we obtain the stability coefficient β_{st} of the stML with the same regularization strength. If we further set $\lambda_0 = \lambda$, we can show that the ratio between the coefficient β of mtML and β_{st} is $\frac{T+1}{2T} \approx \frac{1}{2}$, which indicates that mtML converges twice as fast as stML in this case.*

Next, we upper bound the loss function ℓ by the following theorem.

Theorem 2. *Let $\{H_t\}_{t=0}^T$ be the optimal solution of the mtML problem (1), H_0^* be the optimal solution of ptML with $\mathcal{R}(H_0) = \lambda_0 ||H_0||_{\mathsf{F}}^2$, and $\{H_t^*\}_{t=1}^T$ be the optimal solution of stML with $\mathcal{R}(H_t) = \tilde{\lambda}||H_t||_{\mathsf{F}}^2$, where $\tilde{\lambda} = \frac{\lambda_0 \lambda}{\lambda_0 + \lambda}$. Then, for any task j, the norm of M_j is bounded by*

$$||M_j||_{\mathsf{F}} = ||H_0 + H_j||_{\mathsf{F}} \leq \mathcal{M}_j \triangleq \sqrt{||H_j^*||_{\mathsf{F}}^2 + \frac{\mathcal{G}}{\tilde{\lambda}}},$$

where $\mathcal{G} = \sum_{t=1}^T \Delta\mathcal{L}_{\mathcal{S}_t}$, $\Delta\mathcal{L}_{\mathcal{S}_t} = \big[\big(\mathcal{L}_{\mathcal{S}_t}(H_0^*) + \lambda_0 ||H_0^*||_{\mathsf{F}}^2\big) - \big(\mathcal{L}_{\mathcal{S}_t}(H_t^*) + \tilde{\lambda}||H_t^*||_{\mathsf{F}}^2\big)\big]$, is the overall gap between the pooling-task and single-task learning approaches.

Remark 3. *Theorem 2 connects the generalization bound of mtML with the stML and ptML, and indicates when mtML can work (note that $\mathcal{M}_j$ does not depend on the training performance of mtML) – the model complexity (hence the upper bound B of the loss function, as well as generalization gap) of the j-th task is low, when both 1) the intrinsic complexity of the (single-task) learning problem, measured by $||H_j^*||_{\mathsf{F}}^2$, and 2) the diversity between the tasks, measured by $\mathcal{G}$, are low. Large value of $\mathcal{G}$ indicates low similarity between the tasks, and in this case, mtML may have higher generalization error than stML even though it has faster convergence rate for β. On the other extreme, if the performance gap $\mathcal{G}$ is small, one can expect that mtML can have a tight model complexity bound and also achieve fast convergence rate by Theorem 1. In other words, Theorem 2 formulates the intuition that mtML can work when tasks are similar to each other in the way that stML and ptML have similar learning performances.*

As a concrete example, we combine Lemma 1, Theorem 1 and Theorem 2, and note that hinge loss is σ-admissible with $\sigma = 1$ [26], and upper bounded by $\mathcal{M}_j R^2 + 1$. Then, we the following corollary gives the generalization bound of the mtML algorithm solving (1) with hinge loss [29].

Corollary 1. *Let $\{H_t\}_{t=0}^T$ (hence $\{M_t\}_{t=1}^T$) be the optimal solution of the mtML problem (1) with hinge loss. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the following holds for any task j:*

$$\mathcal{L}_{\mathcal{D}_j}(M_j) - \mathcal{L}_{\mathcal{S}_j}(M_j) \leq \left(\frac{2R^4}{\lambda_0 T n} + \frac{2R^4}{\lambda n}\right) + \left(\frac{2R^4}{\lambda_0 T} + \frac{2R^4}{\lambda} + 2(\mathcal{M}_j R^2 + 1)\right)\sqrt{\frac{\log \frac{1}{\delta}}{2n}}.$$

4 Multitask Boosted Metric Learning (mt-BML)

In this section, we propose an efficient and scalable boosting-based algorithm, termed *multitask boosted metric learning* (mt-BML), to solve the problem (1). Specifically, we consider the relative constraint defined over triplets as in [38] and use the exponential loss

$$\min_{\{H_t\}_{t=0}^T} \log\left(\sum_{t=1}^T \sum_{c=1}^{|\mathcal{C}_t|} \exp -\rho_c^t\right) + \sum_{t=1}^T \mathcal{R}_t(H_0, H_t) \quad \text{s.t.} \quad H_t \in \mathbb{S}_+^d, \tag{2}$$

where we have simplified the notations by introducing $\rho_c^t = \langle X_c^t, H_0 + H_t \rangle$, with $X_c^t = (x_i^t - x_k^t)(x_i^t - x_k^t)^\top - (x_i^t - x_j^t)(x_i^t - x_j^t)^\top$, $(x_i^t, x_j^t, x_k^t) \in \mathcal{C}_t$. $\langle X, H \rangle = \mathrm{tr}(XH^\top)$, $\mathrm{tr}(\cdot)$ being the trace of a matrix, is the inner product of X and H. In addition, to simplify the algorithm derivation, instead of using a Frobenius norm regularizer, we adopt a trace regularizer $\mathcal{R}_t(H_0, H_t) = \lambda_0 \mathrm{tr}(H_0) + \lambda_t \mathrm{tr}(H_t)$ as in [32]. While mt-BML does not use a Frobenius norm as a stabilizer, it still archives good generalization performances in practice, as shown in the experiments.

As a positive semidefinite matrix can be decomposed by a positive linear combination of TORO matrices,

we can reformulate each H_t as $H_t = \sum_{m=1}^{M} w_m^t P_m^t$, with $\text{rank}(P_m^t) = 1$, $\text{tr}(P_m^t) = 1$ and $w_m^t \geq 0$. Then we have $\rho_c^t = {\zeta_c^t}^\top w^0 + {\xi_c^t}^\top w^t$, where $w^t = [w_1^t, \ldots, w_M^t]^\top$, $\zeta_c^t = [\langle X_c^t, P_1^0\rangle, \ldots, \langle X_c^t, P_M^0\rangle]^\top$, and $\xi_c^t = [\langle X_c^t, P_1^t\rangle, \ldots, \langle X_c^t, P_M^t\rangle]^\top$. Now Eq. (2) becomes

$$\begin{aligned}\min_{w,\rho} &\log\left(\sum_{t=1}^{T}\sum_{c=1}^{|\mathcal{C}_t|}\exp -\rho_c^t\right) + T\lambda_0\sum_{m=1}^{M} w_m^0 \\ &+ \sum_{t=1}^{T}\lambda_t\sum_{m=1}^{M} w_m^t \\ &\text{s.t.} \quad \rho_c^t = {\zeta_c^t}^\top w^0 + {\xi_c^t}^\top w^t,\ P_m^t \in \Omega_1, w^t \succeq 0,\end{aligned} \tag{3}$$

where Ω_1 is the set of TORO matrices. The Lagrange function of (3) is

$$\begin{aligned}L(w,\rho,\mu,\nu) &= \log\left(\sum_{t=1}^{T}\sum_{c=1}^{|\mathcal{C}_t|}\exp -\rho_c^t\right) \\ &+ \sum_{t=1}^{T}\lambda_t\sum_{m=1}^{M} w_m^t + T\lambda_0\sum_{m=1}^{M} w_m^0 \\ &+ \sum_{t=1}^{T}\sum_{c=1}^{|\mathcal{C}_t|}\mu_c^t\Big(\rho_c^t - {\zeta_c^t}^\top w^0 - {\xi_c^t}^\top w^t\Big) - \sum_{t=0}^{T}{\nu^t}^\top w^t,\end{aligned} \tag{4}$$

where $\nu \succeq 0$, and μ, ν are the sets of Lagrange multipliers. Then, the dual function is given by

$$\begin{aligned}\inf_{w,\rho} L = &\left[\inf_{\rho}\log\left(\sum_{t=1}^{T}\sum_{c=1}^{|\mathcal{C}_t|}\exp -\rho_c^t\right) + \sum_{t=1}^{T}\sum_{c=1}^{|\mathcal{C}_t|}\mu_c^t\rho_c^t\right] \\ &+ \inf_{\{w^t\}_{t=1}^T}\left[\sum_{t=1}^{T}\lambda_t\sum_{m=1}^{M} w_m^t - {w^t}^\top\Big(\sum_{c=1}^{|\mathcal{C}_t|}\mu_c^t\xi_c^t + \nu^t\Big)\right] \\ &+ \inf_{w^0}\left[T\lambda_0\sum_{m=1}^{M} w_m^0 - {w^0}^\top\Big(\sum_{t=1}^{T}\sum_{c=1}^{|\mathcal{C}_t|}\mu_c^t\zeta_c^t + \nu^0\Big)\right]\end{aligned} \tag{5}$$

It can be observed that problem (5) can be minimized with respect to $w^0, \{w^t\}$, and ρ separately, which gives the Lagrange dual problem of (3) (see the supplementary materials for the detailed derivations):

$$\max_{\mu} -\sum_{t=1}^{T}\sum_{c=1}^{|\mathcal{C}_t|}\mu_c^t\log\mu_c^t, \tag{6}$$

$$\text{s.t.} \quad \mu \succeq 0,\ \sum_{t=1}^{T}\sum_{c=1}^{|\mathcal{C}_t|}\mu_c^t = 1, \tag{C1}$$

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{|\mathcal{C}_t|}\mu_{c,m}^t\zeta_{c,m}^t \leq \lambda_0, \tag{C2}$$

$$\sum_{c=1}^{|\mathcal{C}_t|}\mu_{c,m}^t\xi_{c,m}^t \leq \lambda_t, \quad \forall t = 1, \ldots, T \tag{C3}$$

where $\zeta_{c,m}^t$ and $\xi_{c,m}^t$ are, respectively, the m-th element of ζ_c^t and ξ_c^t, and the formulation of $\mu_{c,m}^t$ will be detailed later. As the number of possible TORO matrices is infinite, neither the primal (3) nor dual (6) can be solved directly. Following the intuition of *column generation* [21], we use an AdaBoost-like algorithm which sequentially adds TORO matrices P_m^t to the current solution. More specifically, at the m-th iteration, a TORO matrix that most violates the constraints (C2) or (C3) is added to the current optimization problem. In other words, we need to solve

$$P_m^\star = \arg\max_{\{P_m^t\}_{t=0}^T}\left\{f_m^t(P_m^t)\right\}, \quad \text{s.t. } P_m^t \in \Omega_1 \tag{7}$$

where

$$f_m^t(P_m^t) = \begin{cases}\left\langle\left(\frac{1}{\lambda_0 T}\sum_{t=1}^{T}\sum_{c=1}^{|\mathcal{C}_t|}\mu_{c,m}^t X_c^t\right), P_m^0\right\rangle, & t = 0 \\ \left\langle\left(\frac{1}{\lambda_t}\sum_{c=1}^{|\mathcal{C}_t|}\mu_{c,m}^t X_c^t\right), P_m^t\right\rangle, & t > 0\end{cases}$$

where $H_{t,m} = \sum_{i=1}^{m} w_i^t P_i^t$ is the solution of H_t at the m-th boosting step. Note that as $\{P_m^t\}$ are TORO matrices, the solution of (7) can be obtained by eigenvalue decomposition. Specifically, let

$$\Sigma_t = \begin{cases}\left(\frac{1}{\lambda_0 T}\sum_{t=1}^{T}\sum_{c=1}^{|\mathcal{C}_t|}\mu_{c,m}^t X_c^t\right), & t = 0 \\ \left(\frac{1}{\lambda_t}\sum_{c=1}^{|\mathcal{C}_t|}\mu_{c,m}^t X_c^t\right), & t > 0\end{cases}, \tag{8}$$

and $\sigma(\Sigma_t)$ be the largest eigenvalue of Σ_t. Then we have

$$P_m^\star = P_m^{\hat{t}} = p_m^{\hat{t}}{p_m^{\hat{t}}}^\top, \tag{9}$$

where $\hat{t} = \arg\max_t \sigma(\Sigma_t)$, and $p_m^{\hat{t}}$ is the corresponding eigenvector. From Eq. (8) it can be observed that $P_m^{t>0}$ (hence $H_{t>0}$) is learned only from the triplets of the t-th task, while P_m^0 (hence H_0) is learned from the triplets of all the tasks. Consequently, the tasks interact with each other when Σ_0 is selected during the boosting procedure.

Once we add a new basis to the current solution, its coefficient $w_m^{\hat{t}}$ can be obtained by minimizing the cost function of primal problem (3). Specifically, the cost function with respect to w_m^0 is (note that $w_m^{t>0} = 0$ if P_m^0 is selected, and we have omitted the terms irrelevant to w_m^0)

$$\log\left(\sum_{t=1}^{T}\sum_{c=1}^{|\mathcal{C}_t|}\exp -\left(\rho_{c,m-1}^t + w_m^0\zeta_{c,m}^t\right)\right) + \lambda_0 w_m^0.$$

Setting the derivative of with respect to w_m^0 to zero, we need to find w_m^0 such that

$$\sum_{t=1}^{T}\sum_{r=1}^{|\mathcal{C}_t|}\mu_{c,m-1}^t(\zeta_{r,m}^t - \lambda_0)\exp -(w_m^0\zeta_{c,m}^t) = 0. \tag{10}$$

Algorithm 1 Multitask Boosted Metric Learning (mt-BML)

Input: Training set $\mathfrak{S}$, regularization parameters $\{\lambda_t\}_{t=0}^T$, number of boosting iterations M.

1: Initialize $\mu_{c,0}^t = \frac{1}{|\mathcal{C}|}$, where $|\mathcal{C}| = \sum_{t=1}^T |\mathcal{C}_t|$.
2: **for** $m = 1, \ldots, M$ **do**
3: Compute $\{p_m^t\}_{t=0}^T$ by solving Eq (8). $\mathcal{O}(|\mathcal{C}|d + Td^2)$
4: Obtain a new base learner $P_m^{\hat{t}}$ using Eq (9). $\mathcal{O}(d^2)$
5: Compute $w_m^{\hat{t}}$ by finding the root of (10) or (11).
6: Call `SROMU` to map the metrics back to a low-rank manifold (optional). $\mathcal{O}(dk)$ or $\mathcal{O}(Tdk)$
7: Update the weight $\mu_{c,m}^t$ for each triplet X_c^t using Eq (12). $\mathcal{O}(|\mathcal{C}|d)$ or $\mathcal{O}(|\mathcal{C}_{\hat{t}}|d)$
8: **end for**

Output: Metric matrices M_t, where $M_t = H_0 + H_t$, with $H_t = \sum_{m=1}^M w_m^t P_m^t,\ \forall t = 0, \ldots, T$.

Similarly, we need to solve

$$\sum_{c=1}^{|\mathcal{C}_{\hat{t}}|} \mu_{c,m-1}^{\hat{t}} (\xi_{c,m}^{\hat{t}} - \lambda_{\hat{t}}) \exp -(w_m^{\hat{t}} \xi_{c,m}^{\hat{t}}) = 0, \qquad (11)$$

for $w_m^{\hat{t}>0}$ (and $w_m^{t\neq\hat{t}} = 0$). Note that as the cost function is convex with respect to w_m^t, there exists a unique global solution for (10) and (11), which can be found by any root-finding algorithm.

It remains to show the formulation of $\mu_{c,m}^t$, which can be derived from the KKT conditions of (4). Setting the derivative of (4) with respect to ρ_c^t to zero gives

$$\mu_c^t = \frac{\exp -\rho_c^t}{\sum_{s=1}^T \sum_{k=1}^{|\mathcal{C}_t|} \exp -\rho_k^s},$$

which suggests that once we add a new matrix $P_m^{\hat{t}}$ to the current solution, $\mu_{c,m}^t$ can be updated by

$$\mu_{c,m}^t = \begin{cases} \frac{\mu_{c,m-1}^t \exp(-w_m^{\hat{t}} \langle X_c^t, P_m^{\hat{t}} \rangle)}{Z_m}, & \hat{t} = 0 \text{ or } t \\ \frac{\mu_{c,m-1}^t}{Z_m}, & \text{otherwise.} \end{cases}, \qquad (12)$$

where Z_m is simply a normalization factor to ensure that $\sum_{t=1}^T \sum_{c=1}^{|\mathcal{C}_t|} \mu_{c,m}^t = 1$. As each base learner is obtained by computing the largest eigenvector of Σ_t, which is a weighted linear combination of X_c^t, as shown in Eq. (8), the dual variable $\mu_{c,m}^t$ plays a role of the weight of each triplet X_c^t as in AdaBoost. The pseudo-code of mt-BML is summarized in Algorithm 1.

4.1 Learning Low-Rank Metric

In practice, a low-rank solution of metrics is usually preferred since it can reduce the memory and model complexity, and therefore may avoid overfitting and improve the generalization performances. In addition, it also corresponds to dimensionality reduction which projects the data to a lower dimensional space with more compact representation. In mt-BML, a low-rank solution can be obtained by limiting the number of boosting iterations. One fundamental problem with this approach is that it can only control the sum of the rank of H_t, but not the rank of each specific H_t. In practice, it may happen that H_t is never learned for some specific tasks t. Moreover, our ultimate objective is to limit the rank of M_t, rather than H_t. In this section, we tackle the low-rank mtML problem by explicitly enforcing fixed-rank constraints on M_t. By exploiting the characteristics of mt-BML, we present an efficient operator that maps metrics back to a low-rank manifold at each boosting iteration.

Let $M \in \mathbb{S}_+^{d,k}$ and $Z \in \mathbb{R}^{d\times d}$, where $\mathbb{S}_+^{d,k}$ is the set of positive semidefinite matrices of rank k. We aim to devise an operator $\mathfrak{M}_M(Z) = M_*$ such that $M_* \in \mathbb{S}_+^{d,k}$, while keep M_* as close to $M + Z$ as possible. Naive approaches (e.g., projected gradient descent) involving repeated singular value decomposition (SVD) are computationally intractable for high-dimensional data, and tend to deviate from the original direction [27]. To circumvent this problem, more efficient algorithms have been proposed in the literature of optimization over Riemannian manifolds [35], which typically consist of two consecutive operations: 1. *Projection* which linearly maps $Z \in \mathbb{R}^{d\times d}$ to the *tangent space* $\mathcal{T}_M\mathbb{S}_+^{d,k} : Z_\mathcal{T} = \mathcal{P}_M(Z) \in \mathcal{T}_M\mathbb{S}_+^{d,k}$. (see the supplementary materials for the formal definition of $\mathcal{T}_M\mathbb{S}_+^{d,k}$). 2. *Retraction* which maps $Z_\mathcal{T}$ back onto the manifold: $Z_\mathcal{R} = \mathcal{R}_M(Z_\mathcal{T}) \in \mathbb{S}_+^{d,k}$. However, a direct implementation of these two operations is computationally unaffordable since a generic retraction is called *exponential mapping* [1], which is still expensive.

As each iteration of mt-BML is a rank-one update: $Z = wP \in \mathbb{S}_+^{d,1}$, it allows us to design an efficient operator that directly maps $M+Z$ back onto $\mathbb{S}_+^{d,k}$ without computing projection and retraction separately. The following Lemma gives an efficient mapping and rigorously proves that our retraction operator is a second-order approximation of exponential mapping.

Lemma 2. *Let $M = UU^\top \in \mathbb{S}_+^{d,k}$, with $U \in \mathbb{R}^{d\times k}$, and $M^\dagger$, $U^\dagger$ be their pseudoinverses. Given a matrix $Z = zz^\top \in \mathbb{S}_+^{d,1}, z \in \mathbb{R}^d$, the operator $\mathfrak{M}_M(Z) = M_* = U_*U_*^\top$, where $U_* = U + V$, with*

$$V = zz^\top {U^\dagger}^\top - \frac{1}{2} P_U zz^\top {U^\dagger}^\top - \frac{1}{2} zz^\top M^\dagger zz^\top {U^\dagger}^\top + \frac{3}{8} P_U zz^\top M^\dagger zz^\top {U^\dagger}^\top \qquad (13)$$

maps $M + Z$ back to the manifold $\mathbb{S}_+^{d,k}$, where $P_U = UU^\dagger = U(U^\top U)^{-1}U^\top$. In addition, we have the projection $Z_\mathcal{T} = \mathcal{P}_M(Z) = P_U zz^\top P_U + P_{U_\perp} zz^\top P_U + P_U zz^\top P_{U_\perp} \in \mathcal{T}_M\mathbb{S}_+^{d,k}$, where $P_{U_\perp} = I - P_U$, and the

Algorithm 2 Symmetric Rank-One Matrix Update on Riemannian Manifold (SROMU)

Input: $M = UU^\top \in \mathbb{S}_+^{d,k}$, $U^\dagger$, $Z = zz^\top \in \mathbb{S}_+^{d,1}$.

1: Compute V using Eq. (13). $\mathcal{O}(dk)$
2: Compute $U_* = U + V$. $\mathcal{O}(d)$
3: Compute $U_*^\dagger$, the rank-one update of the pseudoinverse of U, given $U, U^\dagger$, and Z. $\mathcal{O}(dk)$
4: Compute $P = UV^\top + VU^\top + VV^\top$ $\mathcal{O}(dk)$

Output: The matrix U_*, such that $U_*U_*^\top = M_* \in \mathbb{S}_+^{d,k}$, with its pseudoinverse, $U_*^\dagger$, and the base learner P.

retraction $\mathcal{R}_M : Z_\mathcal{T} \mapsto U_*U_*^\top$ is a second-order Riemannian retraction on $\mathbb{S}_+^{d,k}$. Namely, 1) $\mathcal{R}_M(0) = M$; 2) local rigidity: the derivative of the curve $\gamma_{Z_\mathcal{T}} : \tau \mapsto \mathcal{R}_M(\tau Z_\mathcal{T})$ with respect to τ satisfies $\dot\gamma_{Z_\mathcal{T}}(0) = Z_\mathcal{T}$; and 3) zero initial acceleration condition: $\mathcal{P}_M\left(\frac{d^2\mathcal{R}_M(\tau Z_\mathcal{T})}{d\tau^2}|_{\tau=0}\right) = 0, \forall Z_\mathcal{T} \in \mathcal{T}_M\mathbb{S}_+^{d,k}$.

Remark 4. *Lemma 2 implies that the mapping defined by $\mathfrak{M}_M(Z)$ (i.e., Eq (13)) is sufficiently accurate (but much computationally cheaper), maintains the convergence property of exponential mapping, and enjoys the convergence properties of second-order algorithms [1].*

Based on Lemma 2, we present the *Symmetric Rank-One Matrix Update on Riemannian Manifold* (SROMU) to efficiently maintain the rank of a matrix during the boosting procedure of mt-BML, as shown in Algorithm 2. To map the update back to $\mathbb{S}_+^{d,k}$, we only need to track $U^\dagger$. Since $\text{rank}(V) = 1$, efficient algorithms are available to update the pseudoinverse of rank-one perturbed matrix [25]. As $M_* = M + UV^\top + VU^\top + VV^\top$, we can define the new base learner as $P_m^{\hat t} = UV^\top + VU^\top + VV^\top$ ($w_m^{\hat t} = 1$) for $\hat t > 0$. When $\hat t = 0$ is selected, we update the metric matrices of all the tasks to ensure that $M_t \in \mathbb{S}_+^{d,k}$. Our update rule is based on [35], and is similar in spirit to some previous work [31, 19], yet with key differences: by exploiting of the rank-one and symmetric structure of Z, $\mathfrak{M}_M(Z)$ is simple and elegant. Moreover, it is tailored specifically to mt-BML, producing a rank-2 base learner P, which makes the weights of triplets still updated efficiently (i.e., Eq. 12) after retraction at each boosting iteration.

Computational Complexity At each boosting iteration, the computation cost of the eigenvalue decomposition is $\mathcal{O}(d^2)$ to find the largest eigenvalue and the corresponding eigenvector for each task, and therefore the overall time complexity of line 3 of Algorithm 1 is $\mathcal{O}(Td^2)$. On the other hand, other mtML algorithms based on semidefinite programming (e.g., mt-LMNN [29]) take at least $\mathcal{O}(d^3)$ to project the metric matrix back to $\mathbb{S}_+^d$ for each task. Therefore, the overall complexity is $\mathcal{O}(Td^3)$ at each iteration. As V is a rank-one matrix, Algorithm 2 is generally fast, and the overall complexity is $\mathcal{O}(dk)$, while naive approaches using SVD take $\mathcal{O}(d^2k)$.

5 Related Work

The first attempt to utilizing the multitask learning framework for metric learning is presented in [29], termed as mt-LMNN, which is a multitask extension of the large margin nearest neighbor (LMNN) algorithm [38]. It makes use of the idea that there is a global model parameter shared across the tasks [14, 10], which is equivalent to the assumption that the model parameters of different tasks come from a common prior. Later, this intuition is also adopted in [37] to share the knowledge among semantic labels and social tagging information for image understanding. In [43], the task covariance matrix is proposed to measure more sophisticated task relatedness structures for multitask metric learning, and then extended to the setting of transfer learning by regarding it as a special case of multitask learning. In [41], the von Neumann divergence is introduced to regularize the relationship among multiple tasks, as well as to preserve the data geometry. Recently, the idea of sharing the global parameter across the tasks is revisited in [6] for multitask face matching via coupled projection.

From the theoretical perspective, there have been several studies either on metric learning or multitask learning. For example, by utilizing the notions of VC-dimension and covering number, the generalization bound of multitask learning has been investigated in [3] under the assumption that the tasks share some common hypothesis space. Later, Ando and Zhang [2] present a more general (although related) analysis of structural learning from multiple tasks by leveraging both labeled and unlabeled data. Their analysis is based on Rademacher complexity and a different covering number definition. Based on the notion of Rademacher complexity, improved bounds are presented under the assumption that the tasks are in related in the way such that they share a common linear operator which is chosen to preprocess data [22], or the linear model parameters lie in a low-dimensional subspace [23, 24]. The latter work is also extended to nonlinear model by gradient boosting [36]. Recently, algorithm-dependent bounds have also been analyzed in [20], based on the notion of algorithmic stability.

As for metric learning, the theoretical results are relatively rare. The main obstacle of deriving bounds for metric learning algorithms is that the aforementioned technical approaches are based on the assumption that training examples are i.i.d., which does not hold for the constraints (e.g., pairs, triplets) used for

training a metric. The first attempt to analyzing the generalization bounds for metric learning is presented in [18], where the uniform stability is defined for metric learning. Later, this work is extended to the problem of similarity learning by making use of framework of good similarities [5]. More recently, the notion of algorithmic stability is also used for deriving the generalization bounds of metric hypothesis transfer learning [30]. Other technical approaches to the theoretical analysis of metric learning include Rademacher complexity [16, 8] and algorithmic robustness [4]. In the context of mtML, the only related work we are aware of is [34], where a mtML algorithm is proposed using the idea of group lasso [42, 28], and the generalization guarantee is also analyzed based on the notion of algorithmic robustness [40]. However, the analysis in [34] only reveals the benefits of sparse learning, not providing any insight into the mtML itself.

Low-rank metric learning has been actively studied over the recent years. Inspired by recent advances in optimization over matrix manifolds [1, 35], there have been some studies on devising efficient Riemannian retraction operator, which project the matrix back on the low-rank manifold at each step of optimization procedure [11, 31, 19, 27]. Our method is similar in spirit to these work, yet with key differences: by exploiting of the rank-one and symmetric structure of the base learner of mt-BML, our Riemannian retraction operator is simple and elegant. Moreover, it is tailored specifically to mt-BML, producing a new rank-2 base learner after projection, which makes the weights of triplets still updated efficiently after retraction.

6 Experiments

In this section, we evaluate mt-BML on four benchmark data sets of multitask learning, including Isolet, CoIL, Letter, and USPS.[2] We use the same approach to generate the triplets as in [38, 33]. With respect to the hyper-parameters, we simply set the number of iterations $M = 500$, $\lambda_{t>0} = 0.01$, and tune λ_0 in the grid $\{0.001, 0.002, \ldots, 0.02\}$ by cross-validation. After the metric learning step, the testing examples are classified by using 3-NN classifier. We run the experiments 20 times by randomly splitting training/testing data set and the average results are reported.

Performance Comparison We first evaluate mt-BML against several baseline algorithms, including mt-LMNN [29], mt-GPML [41] and mt-SCML [34] as mtML baselines, as well as st-LMNN [38] and BoostMetric [33] as stML baselines. We vary the ratio of training examples from 0.1 to 0.5, and the results are shown in Figure 1. Overall, the mtML algorithms outperform st-LMNN, especially when the sample size is small. BoostMetric also achieve comparable performances with mt-LMNN, mt-GPML and mt-SCML even without sharing the knowledge from other tasks. On the other hand, mt-BML inherits the benefits from both BoostMetric and multitask learning, and outperforms other baselines. Specifically, mt-BML consistently attains the lowest error rates on CoIL and Letter data sets with different ratios of training examples. It also outperforms other baseline on the Isolet and USPS data sets when the ratio of training examples is smaller than 0.4, and achieves comparable performances with other algorithms when the ratio is between 0.4 and 0.5.

Low-Rank mt-BML Next, we evaluate low-rank mt-BML (mt-LRBML) against several other low-rank metric learning algorithms, including single task LRBML (st-LRBML) by setting $\lambda_0 = 10^4$, ITML [13], LORETA [31], and FRML [19]. We fix the ratio of training examples to be 0.5, vary the rank of metric, and the results are shown in Figure 2. It can be observed that compared with single task low-rank learning algorithms, mt-LRBML substantially improve the learning performances on both data sets across all values of rank, and it outperforms other algorithms even with very low rank. Roughly speaking, the error rates decrease with the increase of the value of rank, but low-rank metrics still achieve comparable performances. For example, the lowest error rate is achieved by the low-rank solution on the Letter data set (rank $k = 110$). In addition, the results are saturated for larger value of the rank k. For example, for the USPS data set, the error rate of mt-BML is reduced by 1.14% when the rank k varies from 15 (1.63%) to 30 (0.49%), but by 0.21% when rank k varies from 30 to 64 (0.28%). In other words, mt-LRBML obtains more compact representations without losing much discriminative power of the data.

Influence of λ_0 In this section, we empirically examine the influence of the regularization parameter λ_0 with different ratios of training examples, as shown in Figure 3. It can be observed that the performance of mt-BML is relatively insensitive to the values of λ_0. We conjecture that this is due to the fact that the boosting procedure itself has the mechanism to balance the trade-off between the tasks by adjusting the weights of examples according to the learning performance each base learner. Further investigation of the learning process could be an interesting direction for the future work. Comparing Figure 3 with the Figure 1, we also observe that mt-BML outperforms other baselines over a wide range of values of λ_0.

Running Time Comparison We verify the efficiency of mt-BML and mt-LRBML with rank $k = 10$ on

[2] See the supplementary materials for more details of the data sets and additional experimental results.

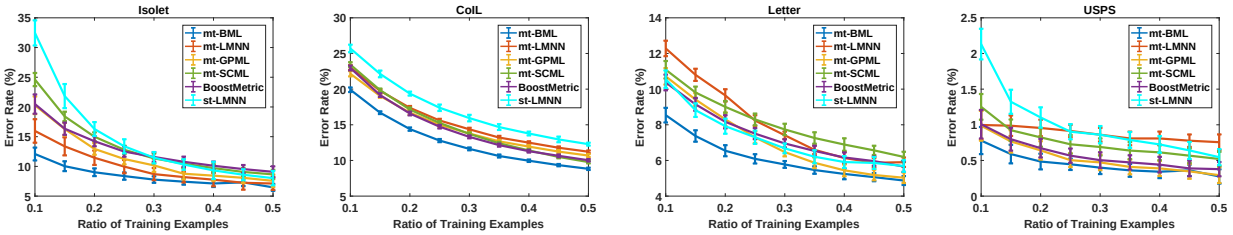

Figure 1: Test error rates (%) of the algorithms with different ratios of training examples.

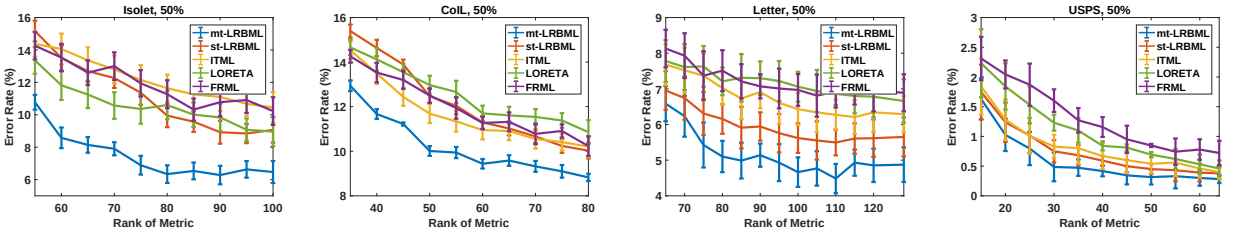

Figure 2: Test error rates (%) with different values of rank.

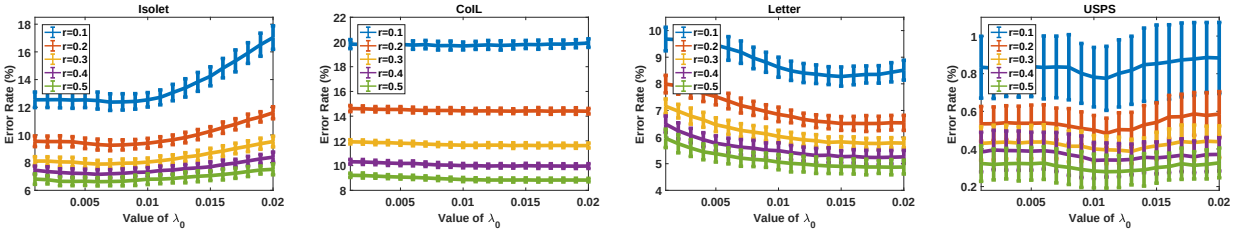

Figure 3: Test error rates with different values of regularization parameter λ_0.

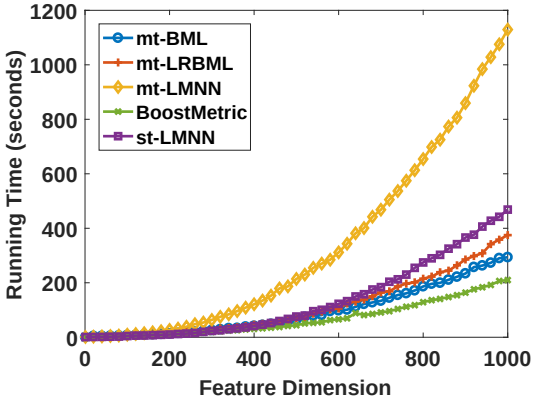

Figure 4: Running time comparison.

the normally distributed synthetic data set. For each task, we keep the number of triplets fixed to 72, and vary the feature dimension from 1 to 1000. As mt-BML avoids explicitly solving the optimization problem with the semidefiniteness constraint by sequentially performing eigenvalue decomposition, it has much lower computational cost than mt-LMNN especially when the feature dimension is high, as shown in Figure 4. The running time of mt-BML is slightly more than BoostMetric, which is mainly because of the extra eigenvalue decomposition of Σ_0. At each boosting iteration, mt-LRBML calls `SROMU` to project the metrics back to the low-rank manifolds, which requires extra computational cost of order $\mathcal{O}(dk)$ or $\mathcal{O}(Tdk)$. As a result, it is slightly slower than mt-BML, but still much more scalable than mt-LMNN.

7 Conclusions

In this paper, we have investigated both theoretical and algorithmic aspects of mtML algorithms. By examining the generalization bound, we theoretically justify the benefits of a widely used mtML formulation. On the algorithmic side, we present mt-BML, which scales well with the feature dimension. In addition, we also investigate the issues of low-rank learning in the setting of mtML. The extensive evaluations verify the efficiency and effectiveness of our algorithms.

We have proposed to use *performance gap* to measure the similarity between domains in the context of multitask learning. In the future, we are interested in extending this notion to other learning paradigms, such as transfer learning and meta learning.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009.

[2] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

[3] J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12(149-198):3, 2000.

[4] A. Bellet and A. Habrard. Robustness and generalization for metric learning. *Neurocomputing*, 151:259–267, 2015.

[5] A. Bellet, A. Habrard, and M. Sebban. Similarity learning for provably accurate sparse linear classification. In *Proceedings of the International Conference on Machine Learning*, pages 1491–1498, 2012.

[6] B. Bhattarai, G. Sharma, and F. Jurie. CP-mtML: Coupled projection multi-task metric learning for large scale face retrieval. In *Proceedings of the IEEE Conference Computer Vision and Pattern Recognition*, pages 4226–4235, 2016.

[7] O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[8] Q. Cao, Z.-C. Guo, and Y. Ying. Generalization bounds for metric and similarity learning. *Machine Learning*, 102(1):115–132, 2016.

[9] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[10] O. Chapelle, P. Shivaswamy, S. Vadrevu, K. Weinberger, Y. Zhang, and B. Tseng. Boosted multi-task learning. *Machine Learning*, 85(1-2):149–173, 2011.

[11] G. Chechik, D. Weinshall, and U. Shalit. Online learning in the manifold of low-rank matrices. In *Advances in Neural Information Processing Systems*, 2011.

[12] J. V. Davis and I. S. Dhillon. Structured metric learning for high dimensional problems. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 195–203. ACM, 2008.

[13] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proceedings of the International Conference on Machine Learning*, pages 209–216. ACM, 2007.

[14] T. Evgeniou and M. Pontil. Regularized multi-task learning. In *SIGKDD*, pages 109–117, 2004.

[15] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems*, pages 513–520, 2005.

[16] Z.-C. Guo and Y. Ying. Guaranteed classification via regularized similarity learning. *Neural Computation*, 26(3):497–522, 2014.

[17] M. Harandi, M. Salzmann, and R. Hartley. Joint dimensionality reduction and metric learning: A geometric take. In *Proceedings of the International Conference on Machine Learning*, 2017.

[18] R. Jin, S. Wang, and Y. Zhou. Regularized distance metric learning: Theory and algorithm. In *Advances in Neural Information Processing Systems*, pages 862–870, 2009.

[19] D. Lim and G. Lanckriet. Efficient learning of mahalanobis metrics for ranking. In *Proceedings of the International Conference on Machine Learning*, pages 1980–1988, 2014.

[20] T. Liu, D. Tao, M. Song, and S. J. Maybank. Algorithm-dependent generalization bounds for multi-task learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):227–241, 2017.

[21] M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.

[22] A. Maurer. Bounds for linear multi-task learning. *Journal of Machine Learning Research*, 7:117–139, 2006.

[23] A. Maurer, M. Pontil, and B. Romera-Paredes. Sparse coding for multitask and transfer learning. In *Proceedings of the International Conference on Machine Learning*, pages 343–351, 2013.

[24] A. Maurer, M. Pontil, and B. Romera-Paredes. The benefit of multitask representation learning. *The Journal of Machine Learning Research*, 17(1):2853–2884, 2016.

[25] C. D. Meyer. Generalized inversion of modified matrices. *SIAM Journal on Applied Mathematics*, 24(3):315–323, 1973.

[26] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT press, 2012.

[27] Y. Mu. Fixed-rank supervised metric learning on riemannian manifold. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1941–1947, 2016.

[28] G. Obozinski, B. Taskar, and M. I. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, 2010.

[29] S. Parameswaran and K. Q. Weinberger. Large margin multi-task metric learning. In *NIPS*, pages 1867–1875, 2010.

[30] M. Perrot and A. Habrard. A theoretical analysis of metric hypothesis transfer learning. In *Proceedings of the International Conference on Machine Learning*, pages 1708–1717, 2015.

[31] U. Shalit, D. Weinshall, and G. Chechik. Online learning in the embedded manifold of low-rank matrices. *Journal of Machine Learning Research*, 13(Feb):429–458, 2012.

[32] C. Shen, J. Kim, L. Wang, and A. Hengel. Positive semidefinite metric learning with boosting. In *Advances in Neural Information Processing Systems*, pages 1651–1659, 2009.

[33] C. Shen, J. Kim, L. Wang, and A. v. d. Hengel. Positive semidefinite metric learning using boosting-like algorithms. *Journal of Machine Learning Research*, 13:1007–1036, 2012.

[34] Y. Shi, A. Bellet, and F. Sha. Sparse compositional metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2078–2084, 2014.

[35] B. Vandereycken and S. Vandewalle. A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2553–2579, 2010.

[36] B. Wang and J. Pineau. Generalized dictionary for multitask learning with boosting. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, pages 2097–2103, 2016.

[37] S. Wang, S. Jiang, Q. Huang, and Q. Tian. Multi-feature metric learning with knowledge transfer among semantics and social tagging. In *CVPR*, pages 2240–2247, 2012.

[38] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, pages 1473–1480, 2005.

[39] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, pages 521–528, 2003.

[40] H. Xu and S. Mannor. Robustness and generalization. *Machine Learning*, 86(3):391–423, 2012.

[41] P. Yang, K. Huang, and C.-L. Liu. Geometry preserving multi-task metric learning. *Machine learning*, 92(1):133–175, 2013.

[42] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68(1):49–67, 2006.

[43] Y. Zhang and D.-Y. Yeung. Transfer metric learning by learning task relationships. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1199–1208, 2010.