Appears in Proc. 23rd European Conf. on Machine Learning (2012)

# Mirror Descent for Metric Learning: A Unified Approach

Gautam Kunapuli and Jude Shavlik

University of Wisconsin-Madison, USA

**Abstract.** Most metric learning methods are characterized by diverse loss functions and projection methods, which naturally begs the question: is there a wider framework that can generalize many of these methods? In addition, ever persistent issues are those of scalability to large data sets and the question of kernelizability. We propose a unified approach to Mahalanobis metric learning: an online regularized metric learning algorithm based on the ideas of composite objective mirror descent (COMID). The metric learning problem is formulated as a regularized positive semi-definite matrix learning problem, whose update rules can be derived using the COMID framework. This approach aims to be scalable, kernelizable, and admissible to many different types of Bregman and loss functions, which allows for the tailoring of several different classes of algorithms. The most novel contribution is the use of the trace norm, which yields a sparse metric in its eigenspectrum, thus simultaneously performing feature selection along with metric learning.

## 1 Introduction

The concept of similarity, or metric, is central to many well-known algorithms such as $k$-means clustering [1], $k$-nearest neighbors [2], multi-dimensional scaling [3] and semi-supervised clustering [4]. While there are many approaches to metric learning, a large body of work is focussed on learning the Mahalanobis distance, which amounts to learning a feature-space transformation and computing the distance in the transformed space. Among these approaches are the work of Xing et al., [5], the large-margin nearest neighbor (LMNN) algorithm [6], information-theoretic metric learning (ITML) [7] and BoostMetric [8]. In addition to the batch approaches above, online algorithms such the pseudo-metric online learning algorithm (POLA) [9] have also been developed. These approaches have been applied successfully to a diverse range of real-world applications such as face verification [10] and road-lane detection [4].

The goal of a metric learning approach is to learn a *distance function*, typically from additional information about the data set. In the supervised and semi-supervised classification setting, the notion of similarity or dissimilarity can be inferred from the class information available from the labels. Thus, if two data points are in the same class, they are assumed to be similar to each other, while two points in different classes are assumed to be dissimilar. The learned metric should ensure that distance between dissimilar points is larger than distance between similar points. Such a metric, can then be used different semi-supervised and unsupervised learning methods such as $k$-means clustering.

In this work, we consider the Mahalanobis metric learning problem applied to $k$-nearest neighbors classification. The Mahalanobis metric is a distance function we learn that is of the form $d(\mathbf{x}, \mathbf{z}) = \|L\mathbf{x} - L\mathbf{z}\|_2$. Thus, we hope to learn a transformation of the data $L$ that separates dissimilar points and brings similar points closer, and we measure distance in this transformed space.

For the remainder of this section, we discuss the problem setting and in Section 2, we introduce composite mirror descent for metric learning. We derive the general update rules, and discuss their implementation details from the perspective of efficiency in Sections 3 and 4. The kernel version of this approach is introduced in Section 5. This method is closely related to several other well-known metric learning approaches and this aspect is discussed in Section 6. In Section 7, we compare the mirror descent approach with some well-known metric learning methods on different data sets, and conclude in Section 8.

## 1.1 Problem Setting

We wish to learn a Mahalanobis metric $d(\mathbf{x},\, \mathbf{z})$ over a feature space $\mathcal{X} \subseteq \mathbb{R}^n$. The metric is a distance function that is used to measure similarity between two instances $\mathbf{x}$ and $\mathbf{z}$ in feature space and satisfies three conditions: $d(\mathbf{x},\, \mathbf{z}) \geq 0$ (non-negativity), $d(\mathbf{x},\, \mathbf{z}) = d(\mathbf{z},\, \mathbf{x})$ (symmetry), and $d(\mathbf{x},\, \mathbf{z}) \geq d(\mathbf{x},\, \mathbf{w}) + d(\mathbf{w},\, \mathbf{z})$ (sub-additivity). We formulate the problem in the spirit of Shalev-Shwartz et al., [9], where the goal is to incrementally learn a metric, given triplets of the form $(\mathbf{x}_t, \mathbf{z}_t, y_t)_{t=1}^T$. The label $y_t = 1$ indicates that training point $\mathbf{x}_t$ is similar to $\mathbf{z}_t$ and $y = -1$ indicates dissimilarity.

The metric we learn is of the form $d(\mathbf{x},\, \mathbf{z}) = \|L(\mathbf{x}{-}\mathbf{z})\|_2$, where $L \in \mathbb{R}^{n\times n}$ is a linear transformation. Since learning this metric directly is difficult owing to non-convexity, we consider instead:

$$d_M(\mathbf{x},\, \mathbf{z})^2 = (\mathbf{x}-\mathbf{z})' L' L(\mathbf{x}-\mathbf{z}) = (\mathbf{x}-\mathbf{z})' M(\mathbf{x}-\mathbf{z}), \tag{1}$$

with $M \in \mathbb{S}^n_+$, the cone of positive semi-definite (psd) matrices. Given $T$ labeled pairs of points $(\mathbf{x}_t, \mathbf{z}_t, y_t)_{t=1}^T$, we learn $(M, \mu)$ such that similar points are transformed to be closer to each other, which dissimilar points are transformed to be farther from each other. This condition can be formulated via the constraints

$$\begin{aligned} \forall(\mathbf{x}, \mathbf{z}, y = +1) \;&\Rightarrow\; d_M(\mathbf{x},\, \mathbf{z})^2 \leq \mu - 1, \\ \forall(\mathbf{x}, \mathbf{z}, y = -1) \;&\Rightarrow\; d_M(\mathbf{x},\, \mathbf{z})^2 \geq \mu + 1, \end{aligned} \tag{2}$$

which can be written simply as $y_t(\mu - d_M(\mathbf{x}_t,\, \mathbf{z}_t)^2) \geq 1$. Note that we cannot have $\mu < 1$ as it implies via the constraints (2) that the distance is negative. We define the margin function for a pair of instances $\mathbf{x}_t$ and $\mathbf{z}_t$, given a label $y_t$, as

$$m(\mathbf{x}_t, \mathbf{z}_t, y_t) = y_t(\mu - d_M(\mathbf{x}_t,\, \mathbf{z}_t)^2) = y_t\left(\mu - (\mathbf{x}_t - \mathbf{z}_t)' M(\mathbf{x}_t - \mathbf{z}_t)\right). \tag{3}$$

This lets us define several loss functions, for instance, the hinge-loss: $\ell_t(M, \mu) = \max\{\,0,\, 1 - m(\mathbf{x}_t, \mathbf{z}_t, y_t)\,\}$. The behavior of such loss functions can be observed in the one-dimensional example in Figure 1. When points $z$ near $x = -0.5$ are labeled similar (Figure 1, left), and their distance measured through the metric $M$ is under the threshold $\mu$, the loss is zero or small. Similarly labeled points $z$ that are far away from $x = -0.5$ are penalized, with the penalty increasing with the distance. In contrast, dissimilarly labeled points near $x = -0.5$ suffer a high loss (Figure 1, right), while those that are sufficiently far away according to the threshold $\mu$ are not penalized. It should be noted that $\mu$ controls the width of sensitivity around $x$. Loss functions are discussed further in Section 2.1.

In addition to learning a metric that minimizes this notion of loss, we also incorporate regularization into the problem so that the resulting metric has sparsity. It is well-known that $\ell_1$-regularization yields sparse solutions [11]; analogously, minimizing the trace-norm of $M$ i.e., the sum of the singular values of $M$ yields sparsity in the spectrum of $M$, thus minimizing the rank of $M$ [12]. Given $T$ samples, the overall problem is one of regularized loss minimization, which leads to an optimization problem of the form

$$\min_{M \succeq 0, \mu \geq 1} \; \frac{1}{T} \sum_{t=1}^{T} \ell_t(M, \mu) \,+\, \rho\, r(M), \tag{4}$$

where the loss function $\ell_t : \mathbb{S}^n_+ \times \mathbb{R} \to \mathbb{R}$ and the regularization function $r : \mathbb{S}^n_+ \to \mathbb{R}$ are both convex and $\rho \in \mathbb{R}_+$ is the regularization parameter. Note that the minimization step in (4) contains a matrix projection into $\mathbb{S}^n_+$, which is a consequence of constraining $M \succeq 0$, and a scalar projection of $\mu \geq 1$. Before describing the proposed approach, we introduce some notation.

## 1.2 Notation and Background

Scalars are denoted in lower-case ($\mu$), vectors in bold face ($\mathbf{v}$), and matrices in upper case ($M$). The vector $\mathbf{e}$ denotes a vector of ones, while $I$ denotes the identity matrix, with the dimension of either being apparent from the context.

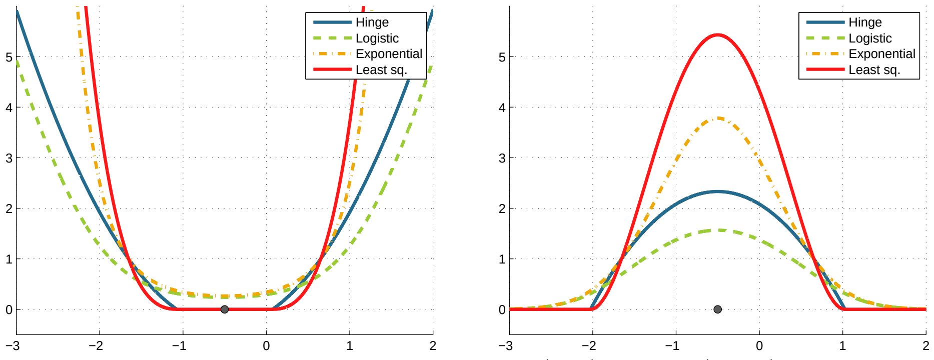

**Fig. 1.** Behavior of different loss functions: (**left**) $y = +1$; (**right**) $y = -1$

For a vector $\mathbf{v}$, the plus function $\mathbf{v}_+$ is defined as the componentwise maximum with respect to zero i.e., $(v_j)_+ = \max\{0, v_j\}$, for the $j$-th component of $\mathbf{v}$. The step function $\mathbf{v}_\star$ is defined componentwise as $(v_j)_\star = v_j$ if $v_j > 0$ and 0 if $v_j \le 0$. The inner product of matrices $X$ and $Y$ is defined as using the trace: $\langle\, X,\, Y \,\rangle = \operatorname{tr} X'Y$. We can also compute matrix gradients; one particularly useful definition is $\nabla_X \langle\, X,\, Y \,\rangle = Y$. We write the singular value decomposition of $X = U\Sigma V$, while for symmetric matrices, we can write $X = V\Lambda V'$. Matrix functions $f(X)$ (e.g., $\exp X$, $\log X$) can be computed via the eigen-decomposition of $X$, that is, $f(X) = Vf(\Lambda)V'$. The Frobenius norm of $X$ is denoted $\|X\|_F = \sqrt{\langle\, X,\, X \,\rangle}$; the trace norm of $X$ is denoted $|||X||| = \mathbf{e}'\boldsymbol{\sigma}$, where $\boldsymbol{\sigma}$ are the singular values of $X$. For symmetric matrices, $|||X||| = \mathbf{e}'|\boldsymbol{\lambda}|$, where $\boldsymbol{\lambda}$ are the eigenvalues of $X$.

The Bregman divergence [13] with respect to a strictly convex function $\psi$ is defined as $B_\psi(\mathbf{x}, \mathbf{z}) = \psi(\mathbf{x}) - \psi(\mathbf{z}) - \nabla\psi(\mathbf{z})'(\mathbf{x} - \mathbf{z})$. For example, the function $\psi(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$ is a Bregman function which induces the squared-Euclidean norm, $B_\psi(\mathbf{x}, \mathbf{z}) = \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2$. This definition can naturally be extended to convex functions defined over matrices i.e., $B_\psi(X, Z) = \psi(X) - \psi(Z) - \langle\, \nabla\psi(Z),\, (X - Z) \,\rangle$. A well-known example is the squared-Frobenius norm $B_\psi(X, Z) = \frac{1}{2}\|X - Z\|_F^2$, induced by $\psi(X) = \frac{1}{2}\|X\|_F^2$.

## 2 Mirror Descent for Metric Learning

The mirror descent algorithm [14] is an iterative proximal-gradient method for minimizing a convex function, $\phi : \Omega \to \mathbb{R}$. Based on this approach, an update in the online setting, with function $\phi_t$ is

$$\mathbf{w}_{t+1} \;=\; \operatorname*{arg\,min}_{\mathbf{w}\in\Omega}\; B_\psi(\mathbf{w},\, \mathbf{w}_t) + \eta\, \nabla'\phi_t(\mathbf{w}_t)(\mathbf{w} - \mathbf{w}_t). \tag{5}$$

Recently, Duchi et al., [15] generalized mirror descent to the case where the functions $\phi_t = \ell_t + r$ are composite, consisting of loss and regularization terms:

$$\mathbf{w}_{t+1} \;=\; \operatorname*{arg\,min}_{\mathbf{w}\in\Omega}\; B_\psi(\mathbf{w},\, \mathbf{w}_t) + \eta\, \nabla'\ell_t(\mathbf{w}_t)(\mathbf{w} - \mathbf{w}_t) + \eta\, r(\mathbf{w}). \tag{6}$$

The subtle, yet significant difference between (5) and (6) is that the entire composite function $\phi_t$ is not linearized. Rather, only $\ell_t$ is linearized; this leads to the composite mirror descent algorithm (COMID). The reason for partial linearization is because general mirror descent applied to $\ell_1$-regularization does not lead to sparse updates, whereas the COMID update does.

We utilize the framework of Duchi et al., to formulate metric learning as an online problem. However, we compute matrix update rules directly rather than derive passive-aggressive-like online updates of [9]. The goal is to optimize the objective (4) in an online manner i.e., at each iteration $t = 1, \ldots, T$, the algorithm receives a labeled pair of points $(\mathbf{x}_t, \mathbf{z}_t, y_t)$, which has an associated loss function $\ell_t(M, \mu)$, and the estimates $M_{t+1}$ and $\mu_{t+1}$ are calculated using a composite mirror descent update rule. Since we are interested in sparse updates as well, we use the trace norm, $|||M|||$, the effect of which is controlled via a regularization parameter $\rho > 0$. We derive generalized update rules for a general loss function and Bregman divergence. At each step, we compute updates given a learning rate $\eta > 0$, and regularization parameter $\rho > 0$ according to

$$M_{t+1} = \underset{M \succeq 0}{\arg\min}\; B_\psi(M,\, M_t) + \eta \,\langle\, \nabla_M \ell_t(M_t, \mu_t),\, M - M_t \,\rangle + \eta\, \rho\; |||\, M |||, \quad (7)$$

$$\mu_{t+1} = \underset{\mu \geq 1}{\arg\min}\quad B_\psi(\mu,\, \mu_t) + \eta\, \nabla_\mu \ell_t(M_t, \mu_t)'\; (\mu - \mu_t). \quad (8)$$

This metric learning formulation[1] has several advantages:

1. **General framework**. Different classes of algorithms can be designed based on different Bregman and loss functions. For example, using the Euclidean distance as the Bregman function produces additive updates, while using relative entropy produces multiplicative updates.
2. **Scalable to large data sets**. As we show below, the matrix updates involve the computation of a rank-1 update to the current eigendecomposition of $M$, which is highly efficient. Furthermore, the rank-one eigen-update scheme discussed here is embarrassingly parallelizable.
3. **Trace-norm regularization produces sparse metric**. The trace-norm regularization ensures that at each step, the updated $M = L'L$ is sparse in its spectrum of singular/eigenvalues. The trace-norm, like the $\ell_1$ norm, introduces sparsity into the eigenvalues of $M$. The solution typically has $r < n$ non-zero eigenvalues: $\tilde{L} = V_r \Lambda_r V_r'$ and the approach performs input-space feature selection.
4. **Theoretical regret guarantees**. In the online optimization setting, for a strongly-convex Bregman function $B_\psi$ and a Lipschitz continuous loss function $\ell_t$, COMID has $O(\sqrt{T})$ regret [15]. Furthermore, strong convexity of the composite function ensures $O(\log T)$ regret.
5. **Kernelizable for nonlinear metric learning**. Many existing distance learning methods are not intuitively kernelizable. Recently, Chatpatanasiri et al., [16] showed various techniques of kernelizing some popular metric learning approaches. Their results are easily extended to this approach in order to learn nonlinear metrics.

## 2.1 Loss Functions

Recall that the margin for a labeled pair of points $(\mathbf{x}_t, \mathbf{z}_t, y_t)$ is defined as $m_t(\mathbf{u}_t, y_t) = y_t(\mu - \mathsf{tr}\, M \mathbf{u}_t \mathbf{u}_t')$, with $\mathbf{u}_t = \mathbf{x}_t - \mathbf{z}_t$. When a pair $(\mathbf{x}_t,\, \mathbf{z}_t)$ are similar with $y_t = 1$, any loss function should produce zero (or small loss) if the distance according to the learned metric is less than a threshold i.e., $\mathsf{tr}\, M \mathbf{u}_t \mathbf{u}_t' \leq \mu$. For dissimilar points $(y_t = -1)$, when $\mathsf{tr}\, M \mathbf{u}_t \mathbf{u}_t' \leq \mu$, the loss suffered is higher. Many such loss functions can be used with the update rules (7–8). Table 1 and Figure 1 show some common loss functions. It is interesting to note that all the loss functions in Table 1 have gradients of the form $\alpha \mathbf{u}\mathbf{u}'$. As we show in Section 4, we can exploit this fact to implement update rules more efficiently.

[1] As the trace-norm allows us to learn a low-rank matrix $M$, this problem is an instance of *pseudo-metric* learning; directions in the null space of $M$ cannot be differentiated.

**Table 1.** Examples of some loss functions, and their gradients with respect to $M$ and $\mu$. For a pair of instances, $\mathbf{u}_t = \mathbf{x}_t - \mathbf{z}_t$, and $m_t(\mathbf{u}_t; M, \mu) = y_t(\mu - \mathsf{tr}\, M\mathbf{u}_t\mathbf{u}_t')$.

| **Loss** | $\ell_t(M_t,\, \mu_t)$ | $\nabla_M\, \ell_t(M_t,\, \mu_t)$ | $\nabla_\mu\, \ell_t(M_t,\, \mu_t)$ |
|---|---|---|---|
| Hinge | $(1 - m_t)_+$ | $(1 - m_t)_\star\ (y_t\mathbf{u}_t\mathbf{u}_t')$ | $-(1 - m_t)_\star\, y_t$ |
| Modified Least Sq. | $\frac{1}{2}\,(1 - m_t)_+^2$ | $(1 - m_t)_+\,(y_t\mathbf{u}_t\mathbf{u}_t')$ | $-(1 - m_t)_+ y_t$ |
| Exponential | $\exp(-m_t)$ | $\exp(-m_t)\,(y_t\mathbf{u}_t\mathbf{u}_t')$ | $-\exp(-m_t)y_t$ |
| Logistic | $\log\,(1 + \exp(-m_t))$ | $\frac{\exp(-m_t)}{1+\exp(-m_t)}\,(y_t\mathbf{u}_t\mathbf{u}_t')$ | $-\frac{\exp(-m_t)}{1+\exp(-m_t)}y_t$ |

### 2.2 Bregman Divergences

Bregman divergences have been extensively studied in literature; see, for instance, Censor and Zenios [17]. The strong convexity of Schatten and entropic matrix functions, which we use here, was recently established by Kakade et al., [18]. Slightly abusing notation, we use $B_\psi(\cdot,\, \cdot)$ for both scalars and matrices.

The squared $p$-norms $\psi(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_p^2$ are strongly convex and induce Bregman divergences. For a matrix $X$, the definition can be extended to Schatten $p$-norms [19], a family of unitary norms defined by applying the $p$-norm to its singular values: $\psi(X) = \frac{1}{2}\|\boldsymbol{\sigma}(X)\|_p^2$. With $p = 2$, we obtain the *squared-Euclidean distance* $B_\psi(\mathbf{x}, \mathbf{z}) = \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2$ for scalars, and the *squared-Frobenius distance* i.e., $B_\psi(X, Z) = \frac{1}{2}\|X - Z\|_F^2$ for matrices.

The function $\psi(\mathbf{x}) = \sum_i x_i \log x_i - x_i$ induces the *Kullback-Liebler (KL) divergence*, $B_\psi(\mathbf{x}, \mathbf{z}) = \sum_i x_i \log \frac{x_i}{z_i} - x_i + z_i$. For a matrix $X$, if $\lambda_i$ is the $i$-th eigenvalue, the Bregman function can be analogously extended: $\psi(X) = \sum_i \lambda_i \log \lambda_i - \lambda_i = \mathsf{tr}\, X \log X - X$ giving us the *von Neumann divergence*, $B_\psi(X, Y) = \mathsf{tr}\,(X \log X - X \log Y - X + Y)$.

## 3 Deriving Update Rules for $M_{t+1}$ and $\mu_{t+1}$

The update rule (7) can be broken down into two separate updates:

$$M_{t+\frac{1}{2}} = \arg\min_M\ B_\psi(M,\, M_t)\ +\ \eta\,\langle\, \nabla_M \ell_t(M_t, \mu_t),\, M - M_t\,\rangle, \tag{9}$$

$$M_{t+1} = \arg\min_{M \succeq 0}\ B_\psi(M,\, M_{t+\frac{1}{2}})\ +\ \eta\rho\ |||\, M\, |||\,. \tag{10}$$

The gradient condition of (9): $0 \in \nabla\psi(M_{t+\frac{1}{2}}) - \nabla\psi(M_t) + \eta\,\nabla_M \ell_t(M_t, \mu_t)$, gives us the intermediate solution: $M_{t+\frac{1}{2}} = \nabla\psi^{-1}\,(\,\nabla\psi(M_t) - \eta\,\nabla_M \ell_t(M_t, \mu_t)\,)$, which can be used to solve (10). The latter is closely related to the trace-norm minimization problem, which was solved by Cai et al., [20]:

$$\underset{X}{\text{minimize}}\ B_\psi(X, Y)\ +\ \tau\ |||\, X\, |||\,. \tag{11}$$

Cai et al., showed that when $\psi(X) = \frac{1}{2}\,\|X\|_F^2$, the optimal solution to (11) is $\Sigma_\tau(Y)$, where $\Sigma_\tau$ is the *singular-value thresholding/shrinkage operator*. For $X \in \mathbb{R}^{m\times n}$, with SVD $X = U\,\mathsf{diag}(\boldsymbol{\sigma})\,V'$, the singular-value shrinkage operator is defined as $\Sigma_\tau(X) = U\, \mathsf{diag}(\boldsymbol{\sigma}_\tau)\, V'$, where $(\sigma_\tau)_i = (\sigma_i - \tau)_+$. Thus, $\Sigma_\tau$ shrinks all singular values by $\tau > 0$, and cuts off those below the specified threshold to zero i.e., those $\sigma_i \leq \tau$. This problem (10) differs from (11) in two key ways:

- The solution is derived for $X, Y \in \mathbb{R}^{m\times n}$, and assumes unitarily invariant Bregman functions $\psi(X)$. It relies on an elegant result by Lewis [21] that shows that for a *unitarily invariant* matrix function $\psi(X)$ (i.e., $\psi(PXQ) = \psi(X)$, for any $P, Q$ unitary), the subdifferential can be calculated as $\nabla\psi(X) = U\, \mathsf{diag}(\nabla\psi(\boldsymbol{\sigma}))\, V'$. However, all Bregman functions are *not* unitarily invariant[2], and consequently, it is not possible to characterize the subgradients in our general case. Fortunately, we are interested in symmetric $X \in \mathbb{S}^n_+$, and in these cases, an analogous result by Lewis [22] characterizes subgradients of *spectral functions* $\psi(X)$ as $\nabla\psi(X) = V\, \mathsf{diag}(\nabla\psi(\boldsymbol{\lambda}))\, V'$, given the eigen-decomposition $X = V \mathsf{diag}(\boldsymbol{\lambda}) V'$. The symmetry of $X$ ensures that $\psi(X)$ are *orthogonally invariant* (i.e., $\psi(QXQ') = \psi(X)$, for any orthogonal $Q$).
- No positivity constraints $X \succeq 0$ are imposed in (11). In our case, since $X$ is a pseudo-metric, we need to ensure that it is positive semidefinite. As we show below, we can derive a closed-form solution, taking into account that $X \succeq 0$, using the *modified eigenvalue thresholding operator*, $S_\tau(X) = V\, \mathsf{diag}(\boldsymbol{\lambda}_\tau)\, V'$, where $(\lambda_\tau)_i = (\lambda_i - \tau)_+$, ensuring that *all eigenvalues below the threshold $\tau$ are cut off*, including all negative eigenvalues.

**Proposition 1.** *The optimal solution to (10) is given by*

$$M_{t+1} \;=\; \nabla\psi^{-1}\left(S_{\eta\rho}(\nabla\psi(M_{t+\frac{1}{2}}))\right) \;=\; V\, \nabla\psi^{-1}\left(\, \mathsf{diag}(S_{\eta\rho}(\boldsymbol{\lambda}))\,\right) V', \tag{12}$$

*where* $\nabla\psi(M_{t+\frac{1}{2}}) \;=\; \nabla\psi(M_t) - \eta\, \nabla_M \ell_t(M_t, \mu_t) \;=\; V\, \mathsf{diag}(\boldsymbol{\lambda})\, V'$.

Note here that the computation of $\nabla\psi(M_{t+\frac{1}{2}})$ involves a symmetric rank-one update because the gradient of the loss function $\nabla_M \ell_t$ is a rank-one matrix (see Table 1). The update essentially consists of computing a symmetric rank-one update to the current eigendecomposition of the pseudo-metric and then cutting off all eigenvalues $< \eta\rho$. We discuss this step further in Section 4. Finally, a closed-form update can be derived for $\mu_{t+1}$ as well, from the formulation (8). In this case, the intermediate solution $\mu_{t+\frac{1}{2}}$ is projected onto $\mu \geq 1$.

**Proposition 2.** *The optimal solution to (8) is given by*

$$\mu_{t+1} = \max\left(\, \nabla\psi^{-1}\left(\nabla\psi(\mu_t) - \eta\, \nabla_\mu \ell_t(M_t, \mu_t)\right),\;\; 1\right). \tag{13}$$

## 4 Implementing Update Rules for $M_{t+1}$

At the $t$-th iteration, with $M_t = V_t \nabla\psi(\Lambda_t) V_t'$, we have:

$$\begin{aligned}
&\textsf{(Intermediate gradient)} & \nabla\psi(M_{t+\frac{1}{2}}) \;&=\; V_t \nabla\psi(\Lambda_t) V_t' - \alpha \mathbf{u}_t \mathbf{u}_t' \\
&\textsf{(EVD of intermediate gradient)} & \nabla\psi(M_{t+\frac{1}{2}}) \;&=\; V_{t+1}\, \Lambda_{t+1}\, V_{t+1}' \\
&\textsf{(Matrix update/thresholding)} & M_{t+1} \;&=\; V_{t+1}\, \nabla\psi^{-1}\left(\, S_{\eta\rho}(\Lambda_{t+1})\,\right) V_{t+1}'
\end{aligned}$$

[2] An example is the entropy function that induces the von Neumann divergence: rotations applied by arbitrary matrices $P$, $Q$ could change the sign of the eigenvalues.

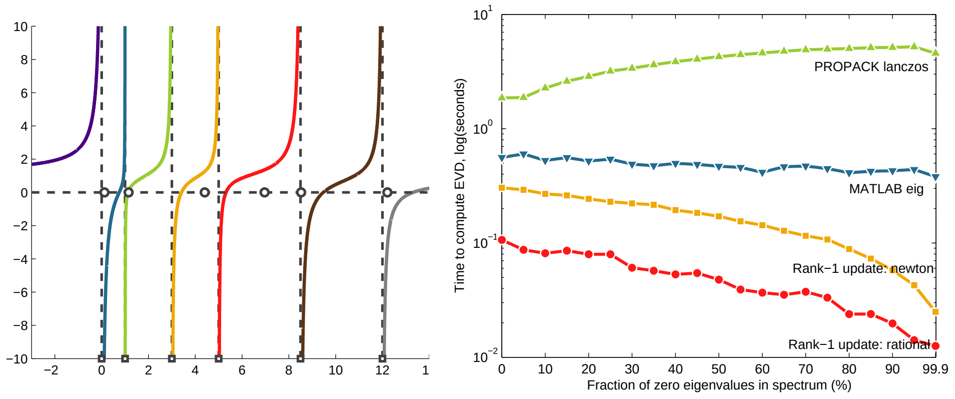

**Fig. 2.** (**left**) The interleaving of eigenvalues $\lambda_i$ ($\square$) of a matrix $M \in \mathbb{S}^6$ with the eigenvalues $\mu_i$ ($\circ$) of a rank-one perturbation $\tilde{M} = M + \rho\mathbf{u}\mathbf{u}'$; (**right**) Comparing the efficiency of Newton's method and the rational interpolation method with Lanczos and `MATLAB`'s `eig` which uses QR + Householder. The approaches are compared on 20 random matrices in $\mathbb{S}_+^{500}$, over increasing sparsity in the spectrum of the matrix.

From Table 1, we observe that the gradient of the loss function is generally of the form $\ell_t = \alpha\mathbf{u}_t\mathbf{u}_t'$, where $\mathbf{u}_t = \mathbf{x}_t - \mathbf{z}_t$. The first two steps constitute a *rank-one update* of the eigendecomposition at the current iteration, which is the most expensive step. While there are several well-known approaches (power iteration, Lanczos, QR + Householder; see [23]), we discuss a different approach that exploits the the *eigenvalue interlacing property* [23, Chapter 8] (Figure 2, left) to significantly improve efficiency.

If $M_t = V\,\mathsf{diag}(\boldsymbol{\lambda})\,V'$, with eigenvalues $\lambda_1 \leq \ldots \leq \lambda_n$, then a symmetric rank-one update is $M_{t+1} = M_t + \alpha\mathbf{u}\mathbf{u}'$, with $M_{t+1} = W\,\mathsf{diag}(\boldsymbol{\mu})\,W'$, and eigenvalues $\mu_1 \leq \ldots \leq \mu_n$. The eigenvalues are related by the secular equation

$$f(\mu) := 1 - \alpha\mathbf{u}'(\mu I_n - \mathsf{diag}(\boldsymbol{\lambda}))^{-1}\mathbf{u} = 0,$$

and the eigenvalues *interlace* i.e., if $\alpha > 0$, $\lambda_1 \leq \mu_1 \leq \lambda_2 \leq \mu_2 \ldots \leq \lambda_n \leq \mu_n$; and if $\alpha < 0$, $\mu_1 \leq \lambda_1 \leq \mu_2 \leq \lambda_2 \leq \ldots \leq \mu_n \leq \lambda_n$. The eigenvectors can also be easily updated as

$$\mathbf{w}_i = \mathbf{v}_i(\mu I_n - \mathsf{diag}(\boldsymbol{\lambda}))^{-1}\mathbf{u},$$

where $\mathbf{w}_i$ and $\mathbf{v}_i$ are the $i$-th columns of $W$ and $V$ respectively. Now, since we know that the $i$-th eigenvalue of $M_{t+1}$, $\mu_i \in (\lambda_i, \lambda_{i+1})$, for $\alpha > 0$ (and $\mu_i \in (\lambda_{i-1}, \lambda_i)$, for $\alpha < 0$) any root-finding method such as Newton-Raphson safeguarded with bisection search can be used to find $\mu_i$. Another consequence of interlacing is that if $\lambda_i = \lambda_{i+1} = \ldots = \lambda_{i+k} = \lambda$, i.e., there are $k$ repeated eigenvalues, then we can avoid the computation of $k-1$ eigenvalues (and eigenvectors) in the update since $\mu_i = \mu_{i+1} = \ldots = \mu_{i+k-1} = \lambda$ as well. This is particularly suitable for our purposes since we seek to introduce more zero eigenvalues into the spectrum of the metric. We discuss some specific details of our implementation:

– **Numerical stability**. General root-finding approaches introduce numerical instability, which propagates into the computation of the eigenvectors leading to non-orthogonality. This issue is addressed by the rational interpolation approach of Gu and Eisenstat [24], which we implement. The approach is based on the observation that while Newton's method uses a local linear approximation of the secular equation, since the secular equation is rational, better stability can be obtained through a local rational approximation.

- **Learning rate**. We use an adaptive learning rate, $\eta_t = \eta/\sqrt{t}$, which gives $O(\sqrt{T})$ regret. The approach requires the user to select the learning rate $\eta$ and the parameter $\rho$, which controls the sparsity of the learned metric.
- **Low Rank Learning**. As, the von Neumann divergence is undefined for low-rank matrices, we compute updates using the *reduced eigendecomposition*, $M_t = \tilde{V}_t \tilde{\Lambda}_t \tilde{V}_t'$, where $\tilde{V}_t$ and $\tilde{\Lambda}_t$ correspond only to the $r$ non-zero eigenvalues. This is similar to the approach of Kulis et al., [25] for low-rank kernel learning. As a result of applying $\nabla\psi^{-1} = \exp(\cdot)$ to the updated eigenvalues in (12), the smallest eigenvalues in the updated matrix will all be 1, resulting in a full-rank matrix. However, we are still able to perform feature selection in this case by selecting the $r$ largest eigenvalues, similar to feature selection in principal components analysis (PCA).

The complete algorithm is described below.

**Algorithm 1** Mirror Descent for Metric Learning

```
 1: input: data $(\mathbf{x}_t, \mathbf{z}_t, y_t)_{t=1}^T$, parameters $\rho$, $\eta > 0$
 2: choose: Bregman functions $\psi(M)$; $\psi(\mu)$, loss function $\ell(M, \mu; \mathbf{x}, \mathbf{z}, y)$
 3: initialize: $M_0 = I_n$, $\mu_0 = 1$
 4: for $(\mathbf{x}^t, \mathbf{z}_t, y_t)$ do
 5:     let $\mathbf{u}_t = \mathbf{x}_t - \mathbf{z}_t$,   $\eta_t = \eta/\sqrt{t}$
 6:     compute gradients of loss $\nabla_M \ell_t = \alpha_t \mathbf{u}_t \mathbf{u}_t'$ and $\nabla_\mu \ell_t = -\alpha_t$ (see Table 1)
 7:     write $\nabla\psi(M_t) = V_t \nabla\psi(\Lambda_t) V_t'$
 8:     compute symmetric rank-one update $V_{t+1}\,\Lambda_{t+1}\,V_{t+1}' = V_t \nabla\psi(\Lambda_t) V_t' - \alpha \mathbf{u}_t \mathbf{u}_t'$
 9:     shrink the eigenvalues $M_{t+1} = V_{t+1}\,\nabla\psi^{-1}\left(S_{\eta\rho}(\Lambda_{t+1})\right) V_{t+1}'$
10:     margin update $\mu_{t+1} = \max\left(\nabla\psi^{-1}\left(\nabla\psi(\mu_t) - \eta\,\nabla\ell_t(M_t, \mu_t)\right),\ 1\right)$
11: end for
```

## 5 Kernel MDML

There are two primary approaches to kernelizing metric learning algorithms: one based on the direct application of the kernel trick, and the other based on the application of the Kernel Principal Components Analysis (KPCA) framework [16]. We use the first approach here. Consider the (possibly infinite-dimensional) nonlinear mapping $\phi : \mathbb{X} \to \mathbb{F}$, that maps all data $\mathbf{x}$ in the input space $\mathbb{X}$ to a high-dimensional feature space $\mathbb{F}$. Associated with this map is a kernel function $\kappa(\cdot, \cdot)$ that can compute inner-products in $\mathbb{F}$ without explicit transformation. Let $X \in \mathbb{R}^{\ell \times n}$ denote the matrix of all examples and $\Phi$ denote the matrix of corresponding high-dimensional vectors obtained from applying the mapping $\phi$ to the data. In *feature space*, the squared Mahalanobis distance is computed as

$d^2(\phi(\mathbf{x}),\, \phi(\mathbf{z})) \;=\; \|L(\phi(\mathbf{x}) - \phi(\mathbf{z}))\,\|_2^2 \;=\; (\phi(\mathbf{x}) - \phi(\mathbf{z}))' L' L(\phi(\mathbf{x}) - \phi(\mathbf{z}))$. If we parameterize $L' \;=\; \Phi G'$, we have that

$$d^2(\phi(\mathbf{x}),\, \phi(\mathbf{z})) \;=\; (\phi(\mathbf{x}) - \phi(\mathbf{z}))' \Phi G' G \Phi(\phi(\mathbf{x}) - \phi(\mathbf{z})).$$

This allows us to kernelize the equation above which leads to a metric in the feature space, $d_\kappa$, expressed in terms of input-space vectors as:

$$d_\kappa^2(\mathbf{x},\, \mathbf{z}) \;=\; (\boldsymbol{\kappa}(X, \mathbf{x}) - \boldsymbol{\kappa}(X, \mathbf{z}))'\, M\, (\boldsymbol{\kappa}(X, \mathbf{x}) - \boldsymbol{\kappa}(X, \mathbf{z})), \tag{14}$$

where $\boldsymbol{\kappa}(X, \mathbf{x})$ is the column of the kernel matrix corresponding to $\mathbf{x}$, and where, with a slight abuse of notation, we have set $M = G'G$. Now, the margin in feature space can be redefined as,

$$m_\kappa(\mathbf{x}_t, \mathbf{z}_t, y_t) \;=\; y_t\,(\,\mu \;-\; (\boldsymbol{\kappa}(X, \mathbf{x}) - \boldsymbol{\kappa}(X, \mathbf{z}))'\, M\, (\boldsymbol{\kappa}(X, \mathbf{x}) - \boldsymbol{\kappa}(X, \mathbf{z}))\,)\,. \tag{15}$$

As before, we can define $\mathbf{u}_t = \boldsymbol{\kappa}(X, \mathbf{x}) - \boldsymbol{\kappa}(X, \mathbf{z})$. Finally, once the matrix $M$ is learned, the Mahalanobis distance of some test point $\tilde{\mathbf{x}}$ with respect to a data point $\mathbf{x}$ can easily be computed as:

$$d_\kappa^2(\tilde{\mathbf{x}},\, \mathbf{x}) \;=\; \boldsymbol{\kappa}(X, \tilde{\mathbf{x}}) - \boldsymbol{\kappa}(X, \mathbf{x}))'\, M\, (\boldsymbol{\kappa}(X, \tilde{\mathbf{x}}) - \boldsymbol{\kappa}(X, \mathbf{x}). \tag{16}$$

## 6 Related Work

Prior approaches to learning the Mahalanobis metric include work by Xing et al., [5], the SVM-based approach of Schultz and Joachims [26] and large-margin nearest neighbors (LMNN) [6]. Davis et al., formulate the metric learning problem as minimizing the Burg divergence subject to similarity constraints, an approach called information theoretic metric learning (ITML) [7]. The BoostMetric approach developed by Shen et al., generalizes the well known AdaBoost algorithm to use a metric as a weak learner rather than a classifier [8]. This results in the optimization of the exponential loss of the margin function, which is solved via coordinate descent. Recently, Guillaumin et al., [27] proposed a metric learning approach that uses logistic regression loss. Many of these algorithms can be viewed as cases of the MDML approach presented here. The MDML approach is also closely related to low rank kernel learning, which was studied by Kulis et al., [25], where the nearness of kernels is measured using the von Neumann and Burg divergences.

There also exist several metric learning approaches such as discriminant adaptive nearest neighbor classification (DANN) [28], neighborhood components analysis (NCA) [29] and relevant components analysis (RCA) [30] that can perform feature selection, in addition to learning a metric. Other online algorithms for supervised learning of the Mahalanobis metric include the work of Shalev-Shwartz et al., [9] and Jain et al, [31].

**Table 2.** UCI data sets

| Data set | #train | #test | #dim | #trn pairs | # classes |
|---|---|---|---|---|---|
| `iris` | 105 | 45 | 4 | 630 | 3 |
| `wine` | 123 | 55 | 13 | 738 | 3 |
| `scale` | 436 | 189 | 4 | 2616 | 3 |
| `segment` | 147 | 63 | 19 | 882 | 7 |
| `breast` | 397 | 172 | 30 | 2382 | 2 |
| `ionosphere` | 245 | 106 | 34 | 1470 | 2 |

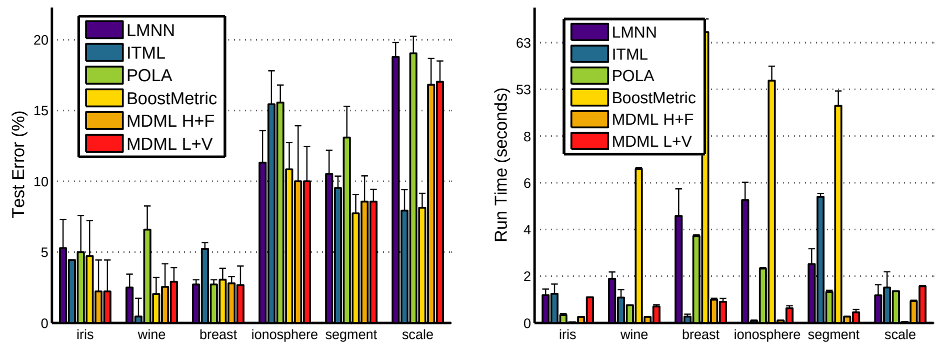

**Fig. 3.** Comparing test error (**left**) and run times (**right**) on six UCI data sets.

## 7 Experiments

In this section, we compare the MDML approach with some current metric learning approaches on various data sets. We consider two classes of algorithms: an additive algorithm that arises from using the hinge loss with the Frobenius divergence (MDML H+F) and a multiplicative algorithm that arises from using the logistic loss with the von Neumann divergence (MDML L+V).

### 7.1 Benchmark Data Sets

We consider four well-known batch and online metric learning approaches: LMNN[3], ITML[4], BoostMetric[5] and POLA [9]. The latter, as well as the MDML approaches were implemented in `MATLAB`.

The performance of these methods on six data sets from the UCI repository[6]. The statistics of these data sets are described in Table 2. All data sets were normalized to zero mean and unit standard deviation. The experimental results are averaged over 10 runs; for each run, the data was split uniformly randomly into training and test sets: 70% was used for training and the remaining 30% was used for testing. The various parameters in ITML, LMNN, MDML H+F and MDML L+V were selected through 10-fold cross validation.

For each data set, we generate triplets and similar/dissimilar pairs for the methods here based on the approach described by Weinberger et al., [6]. To

[3] `http://www.cse.wustl.edu/∼kilian/code/code.html`
[4] `http://www.cs.utexas.edu/∼pjain/itml/`
[5] `http://code.google.com/p/boosting/`
[6] `http://archive.ics.uci.edu/ml/`

summarize, for each data point $\mathbf{x}_t$, $k$ similarly labeled nearest neighbors (targets) and $k$ differently labeled nearest neighbors (impostors) are selected, and triplets are constructed appropriately. We chose $k = 3$ for the generation of triplets and labeled pairs. Once the models were learned, test data were classified using 3-nearest neighbors classification as well.

Figure 3 shows the performance of these approaches with respect to test error and running time.The generalization performance of the MDML approaches is consistently comparable to that of other metric learning approaches. However, the significance of the MDML approaches becomes apparent when considering the running times. BoostMetric is the most expensive approach here, even among the batch approaches, which is not surprising considering it is an ensemble approach. The generalization performance of BoostMetric is good overall, but the performance comes at a significantly higher computational cost. While the MDML approaches outperform the batch methods computationally, they are also faster than POLA, which is an online approach.

We also studied the feature selection performance of the MDML approaches on these benchmark datasets. These results are shown in Figure 4. While it is clear that increasing values of $\rho$ force more features to zero, it is interesting to note that, in many cases, with an appropriate choice of the learning rate $\eta$, it is possible to learn a highly sparse metric whose generalization performance does not degrade significantly. As hoped, the algorithm performs input-space feature selection while learning a pseudo-metric, which can be helpful to practitioners when trying to learn interpretable models. A similar trend is observed when counting the number of eigenvalues that account for 90% of the cumulative energy of the metric. Again, the MDML approaches are able to accumulate more information into a smaller subset of features. While Boostmetric and LMNN are able to perform well by this measure, it should be noted again, that this comes at a higher computational expense.

## 7.2 Digit Recognition

We use the Optical Recognition of Handwritten Digits (`optdigits`) data set from the UCI repository for these experiments. This 64-dimensional, 10 class data set consists of 3823 training points and 1797 test points. Generating triplets using the approach by Weinberger et al., as described above, results in $34,407$ triplets for training. For the metric learning methods that take labeled pairs, this approach resulted in the generation of $11,469$ similar pairs of data and $11,469$ dissimilar pairs of data. As before, we set $k = 3$ for both training and testing. Parameters were selected using 5-fold cross validation. The results are summarized in Table 3.

We compare the different approaches on test error, run time and feature selection. As with the previous benchmark results, LMNN and BoostMetric are able to produce the best models, but again, this comes at the expense of a large computational cost, particularly in the case of BoostMetric. The MDML approaches are able to generalize well overall, but the overall run time for both methods is several orders of magnitude smaller. We also compare the ability

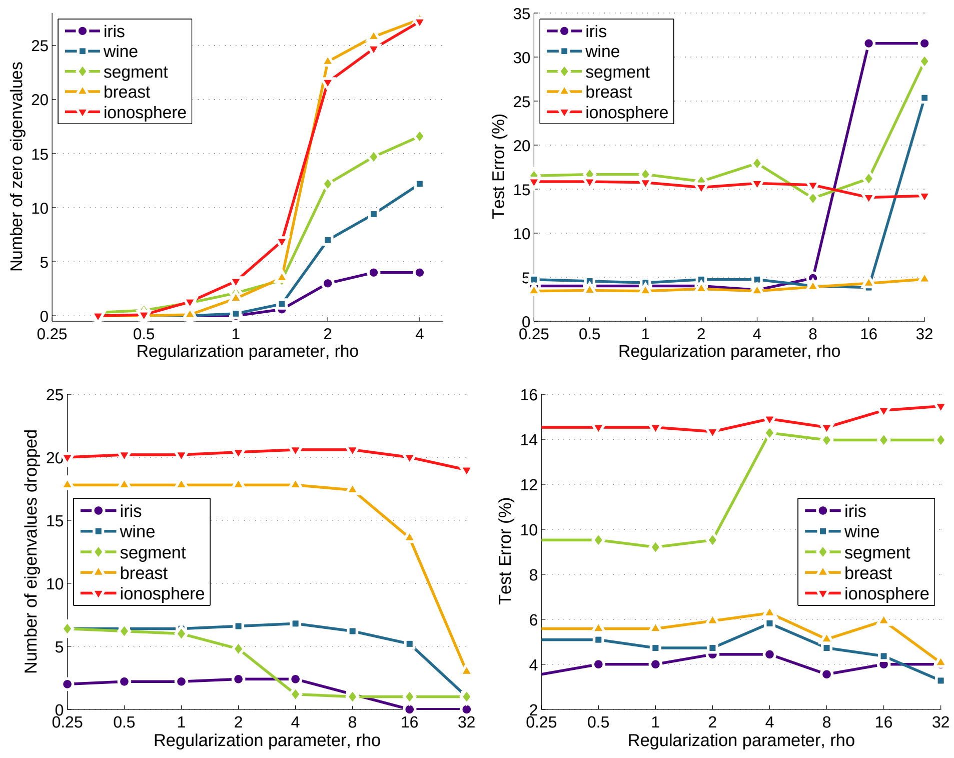

**Fig. 4.** (**top**) `MDML H+F`: (**left**) Average number of zero eigenvalues of the learned metric, with fixed learning rate $\eta$, and different regularization parameters $\rho$; (**right**) the corresponding test error. (**bottom**) `MDML L+V`: number of features dropped i.e., whose eigenvalues do not contribute to the top 90% of the cumulative energy. This experiment could not be performed for `scale` as it was extremely sensitive to parameters.

to perform feature selection across all the data sets using two measures. Given $L = V\mathsf{diag}(\boldsymbol{\lambda})V'$, the first measure is simply the number of non-zero eigenvalues of $L$ i.e., $\|\boldsymbol{\lambda}\|_0$. The second measure is the number of eigenvalues required to account for 90% of the cumulative energy of the metric. The cumulative energy of the $i$-th largest eigenvalue is $e_i = \sum_{j=1}^{i} \lambda_j$. The last column shows the number of features $r$ such that $e_r \geq 0.9 \sum_{i=1}^{n} \lambda_i$; this is used to pick a reduced subset of features during PCA. Both MDML methods are able to perform input space feature selection effectively; for von Neumann, even though a full-rank matrix is learned, we are able to pick a reduced subset because the cumulative energy is concentrated in a few eigenvalues.

## 8 Conclusions and Future Work

We have presented an incremental metric learning approach (MDML) which not only optimizes the notion of loss at every step, but is also regularized. Specifi-

**Table 3.** Performance of the different approaches on the `optdigits` data set.

| Data set | Test Error (%) | Run Time (seconds) | Non-zero features | Num. feats. for 90% energy |
|---|---|---|---|---|
| `LMNN` | 1.669 | 54.213 | 30 | 20 |
| `ITML` | 5.509 | 25.745 | 62 | 43 |
| `POLA` | 2.282 | 14.607 | 53 | 40 |
| `BoostMetric` | 1.758 | 2072.427 | 62 | 19 |
| `MDML H+F` | 1.892 | 15.232 | 26 | 22 |
| `MDML L+V` | 1.948 | 13.768 | 62 | 29 |

cally, we are interested in learning metrics that are sparse in the eigenspectrum, and to this end, the metric learning problem was regularized with the trace norm. This formulation is solved using composite mirror descent and results in a very general framework. Several different types of algorithms can be derived by choosing different loss functions and Bregman functions. Furthermore, the updates result in a symmetric rank-one update at the current step; this can be implemented very efficiently making the approach scalable to large data sets. Preliminary experimental results suggest that the approach performs comparably with current approaches with regard to generalization. However, the ability to learn a metric along with feature selection makes this approach attractive to machine learning practitioners.

These proof-of-concept results suggests several exciting directions for future research, some of which are currently under consideration. Given that the updates are embarrassingly parallelizable, an immediate target is the massive data setting, where we need to learn with millions of data points. In addition, the approach is also amenable to the addition of local geometry constraints in order to learn low-dimensional geometry-aware metrics that lead to representable models. Finally, the kernel-MDML approach is a very powerful extension to linear metric learning, with applications in colored dimensionality reduction and manifold alignment.

## Proof of Proposition 1

Let $A \in \mathbb{S}^n_+$ be a positive semidefinite Lagrange multiplier for the constraint $M \succeq 0$. The matrix Lagrangian for the convex optimization problem (10) is $\mathcal{L}(M) = B_\psi(M, M_{t+\frac{1}{2}}) + \eta\rho\, |||M||| - \langle\, M,\, A\,\rangle$. The optimal primal solution $\nabla\psi(\bar{M})$ and the optimal dual solution $\bar{A}$ should satisfy the first-order necessary conditions:

$$\begin{aligned} \nabla\psi(\bar{M}) \;-\; \nabla\psi(M_{t+\frac{1}{2}}) \;+\; \eta\rho\,\partial\,|||\,\bar{M}||| \;&=\; \bar{A} \quad \text{(gradient condition)},\\ 0 \preceq \nabla\psi(\bar{M}) \;\perp\; \bar{A} &\succeq 0 \qquad\qquad \text{(complementarity)}, \end{aligned} \tag{17}$$

where $\partial|||\bar{M}|||$ denotes the set of all subgradients of $|||\bar{M}|||$ and $X \perp Y$ denotes $\langle\, X,\, Y\,\rangle = 0$. For an $m \times n$ matrix $M$, with SVD $M = U\Sigma V'$, the subgradients are given by [32]:

$$\partial\;|||\,M|||\;=\;\left\{UV' + W \,|\, W \in \mathbb{R}^{m\times n},\; U'W = 0,\; WV = 0,\; \|W\|_2 \leq 1\right\}. \tag{18}$$

We wish to show that the solution $\nabla\psi(\bar{M}) = S_{\eta\rho}\left(\nabla\psi(M_{t+\frac{1}{2}})\right)$ is optimal. To this end, decompose $\nabla\psi(M_{t+\frac{1}{2}}) = V\,\Lambda\,V'$ further, based on its eigenvalues $\lambda_i$

$$\nabla\psi(M_{t+\frac{1}{2}}) = \underbrace{V_1\Lambda_1V_1'}_{\lambda_i>\eta\rho} + \underbrace{V_2\Lambda_2V_2'}_{|\lambda_i|\leq\eta\rho} + \underbrace{V_3\Lambda_3V_3'}_{-\lambda_i<\eta\rho}. \tag{19}$$

By directly thresholding the eigenvalues, we can write the optimal solution as $\nabla\psi(\bar{M}) = V_1(\Lambda_1 - \eta\rho\, I_1)V_1'$. Then, $\partial\ |||\ \bar{M}|||$ at the optimal $\bar{M}$ is

$$\partial\ |||\ \bar{M}||| = \left\{V_1V_1' + W \,|\, W\in\mathbb{S}^n,\ V_1'W = 0,\ WV_1 = 0,\ \|W\|_2 \leq 1\right\}. \tag{20}$$

If we choose $W = \frac{1}{\eta\rho}\,V_2\Lambda_2V_2' \in \mathbb{S}^n$, we immediately have $V_1'W = 0$ and $WV_1 = 0$. Also, $\|W\|_2 = \frac{1}{\eta\rho}\|V_2\Lambda_2V_2'\|_2 = \frac{1}{\eta\rho}\,\|\Lambda_2\|_2 \leq 1$. Thus, $\partial\ |||\ \bar{M}||| = V_1V_1' + \frac{1}{\eta\rho}\,V_2\Lambda_2V_2'$. By this construction, we first have primal feasibility; in fact, $\nabla\psi(\bar{M})$ is positive definite since $\Lambda_1 \succ \eta\rho\, I_1$. From (17), it is easy to see that $\bar{A} = -V_3\Lambda_3V_3'$, which is dual feasible; in fact, it is also positive definite since $\Lambda_3 \prec -\eta\rho\, I_3$. We also have complementarity slackness since $\langle\,\nabla\psi(\bar{M}),\ \bar{A}\,\rangle = 0$, owing to eigenvector orthogonality. Thus, $\nabla\psi(\bar{M}) = S_{\eta\rho}\left(\nabla\psi(M_{t+\frac{1}{2}})\right)$ is optimal to (10), giving us $\bar{M} = \nabla\psi^{-1}\left(\,S_{\eta\rho}\left(\nabla\psi(M_{t+\frac{1}{2}})\right)\right)$. $\square$

## Acknowledgements

The authors gratefully acknowledge the support of Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181, and the National Institutes of Health under the National Library of Medicine grant no. NLM R01-LM008796. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, AFRL, or the US government. The authors would also like to acknowledge anonymous reviewers for their invaluable comments.

## References

1. MacQueen, J.: On convergence of k-means and partitions with minimum average variance. Annals of Mathematical Statistics **36** (1965) 1084ff
2. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory **IT-13**(1) (1967) 21–27
3. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling. Chapman and Hall (2001)
4. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: Proc. 18th ICML. (2001) 577–584
5. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. In: NIPS 15. (2002) 505–512
6. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. In: NIPS 19. (2006)
7. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: Proc. 24th ICML. (2007) 209–216
8. Shen, C., Kim, J., Wang, L., van den Hengel, A.: Positive semidefinite metric learning with boosting. In: NIPS 22. (2009) 629–633

9. Shalev-Shwartz, S., Singer, Y., Ng, A.Y.: Online and batch learning of pseudo-metrics. In: Proc. 21st ICML. (2004) 94–102
10. Chopra, S., Hadsell, R., Lecun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: Proceedings of Computer Vision and Pattern Recognition Conference, IEEE Press (2005) 539–546
11. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B **58** (1994) 267–288
12. Recht, B., Fazel, M., Parrilo, P.A.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Rev. **52**(3) (2010) 471–501
13. Bregman, L.M.: The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. USSR Computational Mathematics and Mathematical Physics **7** (1967) 200–217
14. Beck, A., Teboulle, M.: Mirror descent and nonlinear projected subgradient methods for convex optimization. Operations Research Letters **31** (2003) 167–175
15. Duchi, J., Shalev-Shwartz, S., Singer, Y., Tewari, A.: Composite objective mirror descent. In: COLT. (2010) 14–26
16. Chatpatanasiri, R., Korsrilabutr, T., Tangchanachaianan, P., Kijsirikul, B.: On kernelization of supervised Mahalanobis distance learners. Computing Research Repoisitory (CoRR) **abs/0804.1441** (2008)
17. Censor, Y.A., Zenios, S.A.: Parallel Optimization: Theory, Algorithms and Applications. Oxford University Press (1997)
18. Kakade, S.M., Shalev-Shwartz, S., Tewari, A.: On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. **preprint**
19. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press (1990)
20. Cai, J.F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization **20** (March 2010) 1956–1982
21. Lewis, A.S.: The convex analysis of unitarily invariant matrix functions. Journal of Convex Analysis **2** (1995) 173–183
22. Lewis, A.S.: The mathematics of eigenvalue optimization. Mathematical Programming **97** (2003) 155–176
23. Golub, G.H., Van Loan, C.F.: Matrix Computations (Johns Hopkins Studies in Mathematical Sciences). 3rd edn. The Johns Hopkins University Press (1996)
24. Gu, M., Eisenstat, S.C.: A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem. SIAM Journal on Matrix Analysis and Applications **15**(4) (1994) 1266–1276
25. Kulis, B., Sustik, M.A., Dhillon, I.S.: Low-rank kernel learning with bregman matrix divergences. J. Mach. Learn. Res. **10** (2009) 341–376
26. Schultz, M., Joachims, T.: Learning a distance metric from relative comparisons. In: NIPS 16. (2004)
27. Guillaumin, M., Verbeek, J.J., Schmid, C.: Is that you? metric learning approaches for face identification. In: ICCV. (2009) 498–505
28. Hastie, T., Tibshirani, R.: Discriminant adaptive nearest neighbor classification. IEEE Trans. on Pattern Analysis and Machine Intelligence **18**(6) (1996) 607–616
29. Goldberger, J., Roweis, S.T., Hinton, G.E., Salakhutdinov, R.: Neighbourhood components analysis. In: NIPS 17. (2004)
30. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning distance functions using equivalence relations. In: Proc. 20th ICML. (2003) 11–18
31. Jain, P., Kulis, B., Dhillon, I.S., Grauman, K.: Online metric learning and fast similarity search. In: NIPS. (2008) 761–768
32. Watson, G.A.: Characterization of the subdifferential of some matrix norms. Linear Algebra and its Applications **170** (1992) 33–45