Distance Metric Tracking

Kristjan Greenewald *[1], Stephen Kelley[2], and Alfred O. Hero III[1]

[1] University of Michigan, 500 S State Street, Ann Arbor, MI 48109
[2]MIT Lincoln Laboratory, 244 Wood Street, Lexington, MA 02420

March 2, 2016

Abstract

Recent work in distance metric learning has produced numerous methods aimed at learning transformations of data that best align with provided sets of pairwise similarity and dissimilarity constraints. The learned transformations lead to improved retrieval, classification, and clustering algorithms due to the more accurate distance or similarity measures. Here, we introduce the problem of learning these transformations when the underlying constraint generation process is dynamic. These dynamics can be due to changes in either the ground-truth labels used to generate constraints or changes to the feature subspaces in which the class structure is apparent. We propose and evaluate an adaptive, online algorithm for learning and tracking metrics as they change over time. We demonstrate the proposed algorithm on both real and synthetic data sets and show significant performance improvements relative to previously proposed batch and online distance metric learning algorithms.

1 Introduction

The effectiveness of many machine learning and data mining applications rely on an appropriate measure of pairwise distance between data points that accurately reflects the objective, e.g., prediction, clustering or classification. In settings with clean, appropriately-scaled spherical Gaussian data, standard Euclidean distance can be utilized. However, when the data is heavy tailed, multimodal, contaminated by outliers, irrelevant or replicated features, or observation noise, Euclidean inter-point distance can be problematic, leading to bias or loss of discriminative power.

As a result, many unsupervised, data-driven approaches for identifying appropriate distances between points have been proposed. These methodologies, broadly taking the form of dimensionality reduction or data "whitening", aim to utilize the data itself to learn a transformation of the data that embeds it into a space where Euclidean distance

*Corresponding author

is appropriate. Examples of such unsupervised techniques include Principal Component Analysis [2], Multidimensional Scaling [13], covariance estimation [13, 2], and manifold learning [19]. Such unsupervised methods do not have the benefit of human input on the distance metric, and overly rely on prior assumptions, e.g., local linearity or smoothness.

This paper proposes methods for distance metric learning. In this problem one seeks to learn linear transformations of the data that are well matched to a particular task specified by the user. In this case, point labels or constraints indicating point similarity or dissimilarity are used to learn a transformation of the data such that similar points are "close" to one another and dissimilar points are distant in the transformed space. Learning distance metrics in this manner allows a more precise notion of distance or similarity to be defined that is related to the task at hand.

Many supervised and semi-supervised distance metric learning approaches have been developed [17]. This includes online algorithms [18] with regret guarantees for situations where similarity constraints are received in a stream. In this paper, we propose a new way of formulating the distance metric learning task. We assume the underlying ground-truth distance metric from which constraints are generated is evolving over time. This problem formulation suggests an adaptive, online approach to track the underlying metric as constraints are received. We present an algorithm for tracking distance metrics based on recent advances in composite objective mirror descent for metric learning [10] (COMID) and the Strongly Adaptive Online Learning (SAOL) framework proposed in [7].

1.1 Related Work

Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) are classic examples of linear transformations for projecting data into more interpretable low dimensional spaces. Unsupervised PCA seeks to identify a set of axes that best explain the variance contained in the data. LDA takes a supervised approach, minimizing the intra-class variance and maximizing the inter-class variance given class labeled data points.

Much of the recent work in Distance Metric Learning has focused on learning Mahalanobis distances on the basis of pairwise similarity/dissimilarity constraints. These methods have the same goals as LDA; pairs of points labeled "similar" should be close to one another while pairs labeled "dissimilar" should be distant. MMC [25], a method for identifying a Mahalanobis metric for clustering with side information, uses semidefinite programming to identify a metric that maximizes the sum of distances between points labeled with different classes subject to the constraint that the sum of distances between all points with similar labels be less than some constant.

Large Margin Nearest Neighbor (LMNN) [23] similarly uses semidefinite programming to identify a Mahalanobis distance, however it modifies the constraints to only take into account a small, local neighborhood for each point. In this setting, the algorithm minimizes the sum of distances between a given point and its similarly labeled neighbors while forcing differently labeled neighbors outside of its neighborhood. This method has been shown to be computationally efficient [24] and, in contrast to the similarly motivated Neighborhood Component Analysis [11], is guaranteed to converge to a globally optimal solution. Additionally, constraining the optimization based only on a small neighborhood of points enables effective processing of multi-modal classes.

Information Theoretic Metric Learning (ITML) [8] is another popular Distance Metric Learning technique. ITML minimizes the Kullback-Liebler divergence between an initial guess of the matrix that parameterizes the Mahalanobis distance and a solution that satisfies a set of constraints. The constraints in this setting are based on similarity and dissimilarity pairs and are constructed such that similar pairs be within some closeness constant and dissimilar pairs be more distant than some larger constant. Online and non-linear extensions to the ITML methodology are presented as well.

In a dynamic environment, it is necessary to be able to compute multiple estimates of the changing metric at different times, and to be able to compute those estimates online. Online learning [5] meets these criteria by efficiently updating the estimate every time a new data point is obtained, instead of solving an objective function formed from the entire dataset.

Many online learning methods have regret guarantees, that is, the loss in performance relative to a batch method is provably small [5, 10]. In practice, however, the performance of an online learning method is strongly influenced by the learning rate which may need to vary over time in a dynamic environment [7, 21, 9].

Adaptive online learning methods attempt to address this problem by continuously updating the learning rate as new observations become available. For example, AdaGrad-style methods [21, 9] perform gradient descent steps with the step size adapted based on the magnitude of recent gradients. Follow the regularized leader (FTRL) type algorithms adapt the regularization to the observations [20]. Recently, a method called Strongly Adaptive Online Learning (SAOL) has been proposed, which maintains several learners with different learning rates and selects the best one based on recent performance [7]. Several of these adaptive methods have provable regret bounds [20, 15, 14]. These typically guarantee low total regret (i.e. regret from time 0 to time t) at every time [20]. SAOL, on the other hand, is guaranteed to have low regret on every subinterval, as well as low regret overall [7].

The remainder of this paper is structured as follows. In Section 2 we formalize the distance metric tracking problem, and section 3 reviews the existing COMID learning framework. Section 4 introduces our adaptive approaches to solving the distance metric tracking problem, and section 5 presents our Strongly Adaptive Online Metric Learning algorithm. Results on both synthetic data and a text review dataset are presented in Section 6 with discussion and future work presented in Section 7.

2 Problem Formulation

The goal of this work is to use analyst feedback to learn a metric on the data space that best matches the goals of the analyst. We formulate the problem as a cooperative dynamic game between the learner and the analyst. Both players' goal is for the learner to learn the internal metric $\mathbf{M}$ used by the analyst. The metric is changing over time, making the game dynamic.

The analyst selects pairs of data points $(\mathbf{x}_t, \mathbf{z}_t)$ and labels them as similar or dissimilar. The labels are assumed to arrive in a temporal sequence, hence the labels at the beginning may have arisen from a different metric than those at the end of the sequence.

In sum, the learning goals include tracking the analyst's internal metric in the presence of metric changes and noise, and (equivalently) finding an embedding which results in maximal separation of the clusters of interest to the analyst, enabling better interpretation and/or future feedback from the analyst. Potential extensions which we do not have the space to treat here include exploiting unlabeled data points [1], and/or choosing which pairs or groups of pairs to present to the analyst (i.e. active learning [22]).

2.1 Objective function

Metric learning seeks to learn a metric that encourages data points marked as similar to be close and data points marked as different to be far apart. The Mahalonobis distance is parameterized by $\mathbf{M}$ as

$$d^2_M(\mathbf{x},\mathbf{z}) = (\mathbf{x}-\mathbf{z})^T\mathbf{M}(\mathbf{x}-\mathbf{z}) \tag{1}$$

where $\mathbf{M} \in \mathbb{R}^{n\times n} \succeq 0$.

Suppose a set of similarity constraints are given, where each constraint is the triplet $(\mathbf{x}_t, \mathbf{z}_t, y_t)$, $\mathbf{x}_t$ and $\mathbf{z}_t$ are data points in $\mathbb{R}^n$, and the label $y_t = +1$ if the points $\mathbf{x}_t, \mathbf{z}_t$ are similar and $y_t = -1$ if they are dissimilar.

Following [18], we introduce the following margin based constraints:

$$\begin{aligned} d^2_M(\mathbf{x}_t,\mathbf{z}_t) &\leq \mu - 1, \quad \forall\{t|y_t = 1\} \\ d^2_M(\mathbf{x}_t,\mathbf{z}_t) &\geq \mu + 1, \quad \forall\{t|y_t = -1\} \end{aligned} \tag{2}$$

where μ is a threshold that controls the margin between similar and dissimilar points. A diagram illustrating these constraints and their effect is shown in Figure 1.

In typical fashion, these constraints are softened by penalizing violation of the constraints with a convex loss function ℓ_t. This gives the following objective:

$$\min_{\mathbf{M}\succeq 0, \mu \geq 1} \frac{1}{T}\sum_{t=1}^{T} \ell_t(\mathbf{M},\mu) + \rho r(\mathbf{M}) \tag{3}$$

$$\ell_t(\mathbf{M},\mu) = \ell(m_t),\ m_t = y_t(\mu - \mathbf{u}_t^T\mathbf{M}\mathbf{u}_t),\ \mathbf{u}_t = \mathbf{x}_t - \mathbf{z}_t$$

where r is the regularizer. Kunapuli and Shavlik propose using nuclear norm regularization ($r(\mathbf{M}) = \|\mathbf{M}\|_*$) to encourage projection of the data onto a low dimensional subspace (feature selection/dimensionality reduction).

3 Composite Objective Mirror Descent

One principled approach to online learning involves viewing the acquisition of new data points as stochastic realizations of the underlying distribution, suggesting the use

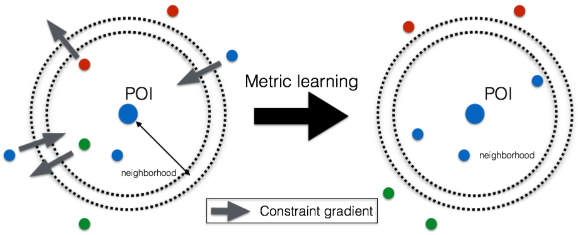

Figure 1: Visualization of the margin based constraints (2), with colors indicating class. The goal of the metric learning constraints is to move target neighbors towards the point of interest (POI), while moving points from other classes away from the target neighborhood.

of stochastic mirror descent techniques. The authors of [18] propose a composite objective mirror descent (COMID) approach to online metric learning that solves a regularized positive semidefinite learning problem.

Using the COMID framework [10], for the objective (3) we have online learning updates that iterate through the constraints

$$\begin{aligned}\hat{\mathbf{M}}_{t+1} =& \arg\min_{\mathbf{M}\succeq 0} B_\psi(\mathbf{M}, \hat{\mathbf{M}}_t) \\ &+ \eta_t \langle \nabla_M \ell_t(\hat{\mathbf{M}}_t, \mu_t), \mathbf{M} - \hat{\mathbf{M}}_t \rangle + \eta_t \rho \|\mathbf{M}\|_* \\ \hat{\mu}_{t+1} =& \arg\min_{\mu \geq 1} B_\psi(\mu, \hat{\mu}_t) + \eta_t \nabla_\mu \ell_t(\hat{\mathbf{M}}_t, \hat{\mu}_t)'(\mu - \hat{\mu}_t),\end{aligned} \tag{4}$$

where B_ψ is any Bregman divergence and η_t is the learning rate parameter. $\hat{\mathbf{M}}_0, \hat{\mu}_0$ are initialized to some initial value. In [18] a closed-form algorithm for solving the minimization in (4) is developed for a variety of common losses and Bregman divergences, involving rank one updates and eigenvalue shrinkage. A kernel version of the algorithm is also available for the batch case.

By standard mirror descent analysis, this method has $O(\sqrt{T})$ regret for the static case when the learning rate is set as $\eta_t = \eta/\sqrt{t}$. For online learning of a static objective, the learning rate will decay to zero. However, in the case of a dynamic objective, the learning rate must not decay to zero so that the estimate of the parameters will be most strongly influenced by the recent constraint history. This was proposed in a generic online learning scenario in [12], where low regret guarantees were derived, which we extend to metric learning in the supplementary material. Critically, the optimal learning rate depends on how fast the objective is changing. We propose two methods for addressing this issue and for learning distance metrics that change over time in an arbitrary way.

4 Dynamic Metric Learning Algorithms

4.1 Windowed Batch Approach

Intuitively, if the underlying metric is changing smoothly, the most recent samples are the most relevant. Similarly to the covariance estimation method of [26], it is possible to apply batch methods to learn a changing metric. At any given time the importance of past samples are weighted by their recency, and a batch method is used to estimate the current metric. This is then repeated at various times, giving in effect a weighted sliding window of samples from which to learn. The resulting objective is

$$\min_{\mathbf{M}_t \succeq 0, \mu_t \geq 1} \sum_{k=1}^{K} a_k \ell_{k+t-K}(\mathbf{M}_t, \mu_t) + \rho r(\mathbf{M}_t) \tag{5}$$

where a_k is such that $\sum_{k=1}^{K} a_k = 1$. This function is convex. Nonrectangular windows can be more difficult computationally, and repeated batch processing is not efficient.

In our experiments, we use COMID to solve the objective (5) at each step. Since from t to $t+1$ the objective function only changes slightly if K is large enough and a is sufficiently smooth, computational complexity is reduced by initializing the current update with the previous estimate.

4.2 Adaptive Online Approach

In an online learning scenario where drift is occurring, as noted above, the choice of the learning rate η_t can be critical. Furthermore, if discrete shifts and/or changes in drift occur, the optimal η_t may change with time, and setting a drift rate dependent η_t using cross validation is not practical in a truly online setting. Hence, a method of adaptively choosing the learning rate in an online fashion is desirable.

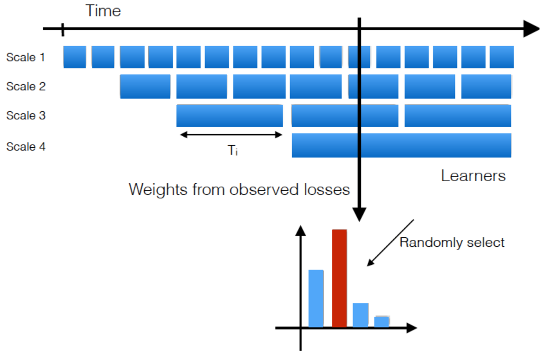

Figure 2: Strongly Adaptive Online Learning - Learners at multiple scales run in parallel. Observed losses for each are used to create weights that are used to select the current scale.

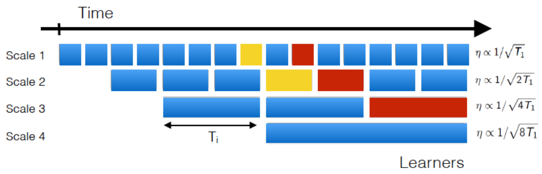
Time
Scale 1
Scale 2
Scale 3
Scale 4
$\eta \propto 1/\sqrt{T_1}$
$\eta \propto 1/\sqrt{2T_1}$
$\eta \propto 1/\sqrt{4T_1}$
$\eta \propto 1/\sqrt{8T_1}$
T_i
Learners

Figure 3: Stochastic mirror descent learners and initialization. Each yellow and red learner is initialized by the output of the previous learner of the same color, that is, the learner of the next shorter scale.

While any method of adaptively setting η_t may be used, in this work we chose the Strongly Adaptive Online Learning (SAOL) framework of [7] because of its ability to perform well on every time subinterval.

SOAL proposes running a bank of multiple online base learners in parallel, each having parameters optimized for learning on an interval of a different length, or alternatively in our case, for learning a metric that has its drift spread out over an interval of a different length. SAOL then uses the recent history of losses suffered by each learner to select the learner that is most accurate at the current time (Figure 2). SAOL has strong theoretical guarantees on the regret on every subinterval, as opposed to the traditional bounds on regret over the entire learning period. This guarantees that the estimate will be sufficiently responsive to make it accurate at all times.

We use the COMID online learners of Section 3 (with learning rate η_t) as the base learners. Algorithm 1 shows the SAOL algorithm applied to the metric learning problem. The next section explains SAOL and its implementation.

5 SAOML

5.1 SAOL Framework

We first describe the SAOL framework of [7]. SAOL is based on dyadically partitioning the temporal axis into intervals and assigning a black box learner to each interval. Specifically, define a set $\mathcal{I}$ of intervals $I = [t_{I1}, t_{I2}]$ such that the lengths $|I|$ of the intervals are proportional to powers of two, i.e. $|I| = I_0 2^j$, with an arrangement that is a dyadic partition of the temporal axis. The first interval of length $|I|$ starts at $t = |I|$ (see Figure 2), and additional intervals of length $|I|$ exist such that the rest of time is covered.

Every interval I is associated with a base learner that operates on that interval. Hence, at a given time t, a set $\mathrm{ACTIVE}(t) \subseteq \mathcal{I}$ of $\mathrm{floor}(\log_2 t)$ intervals/learners are active, running in parallel. The base learner of given interval I is designed to have low total regret ($O(\sqrt{|I|})$) on that interval in the static case. Because the parameters being learned are changing with time, learners designed for low regret at different scales will have different performance (analogous to the classic bias/variance tradeoff). In other words, there is an optimal scale $|I|$ that can be selected from the base learning ensemble.

Algorithm 1 Strongly Adaptive Online Metric Learning

```
Initialize: $w_1(I)$
for $t = 1$ to $T$ do
    Initialize new learner if needed.
    Choose $\hat{I} \in \text{ACTIVE}(t)$ according to (10).
    Mirror Descent update (4) for all active learners.
    Set $\mathbf{M}_t \leftarrow \mathbf{M}_t(\hat{I})$, $\mu_t \leftarrow \mu_t(\hat{I})$
    Obtain constraint $(\mathbf{x}_t, \mathbf{z}_t, y_t)$, compute loss $\ell_{t,log}(\cdot)$.
    Update weights for all $t \in I$:
        $r_t(I) = \left( \sum_I \frac{w_t(I)}{W_t} \ell_{t,log}(\mathbf{M}_t(I), \mu_t(I)) \right) - \ell_{t,log}(\mathbf{M}_t(I), \mu_t(I))$
        $w_{t+1}(I) = w_t(I)(1 + \min\{1/2, 1/\sqrt{|T_i|}\} r_t(I))$
end for
Return $\{\mathbf{M}_t, \mu_t\}$.
```

It remains to select the output of one of the active learners and use it as the final estimate at any given time t. In [7], it is proposed to compute weights for each learner. These weights are updated based on the learner's recent estimated regret, which is estimated as described below, and are used to randomly select a learner. In our work, we update the weights according to

$$\begin{aligned} w_{t+1}(I) =& w_t(I)(1 + \eta_I r_t(I)), \quad \forall t \in I \\ r_t(I) =& \left(\sum_I \frac{w_t(I)}{W_t} \ell_t(\mathbf{M}_t(I), \mu_t(I)) \right) \\ & - \ell_t(\mathbf{M}_t(I), \mu_t(I)) \end{aligned} \tag{6}$$

for all $I \in \mathcal{I}$, where $\eta_I = \min\{1/2, 1/\sqrt{|I|}\}$, where $\mathbf{M}_t(I), \mu_t(I)$ are the outputs at time t of the learner on interval I, and $r_t(I)$ is called the estimated regret of the learner on interval I at time t. Essentially, this is highly weighting low loss learners and lowly weighting high loss learners.

For any given time t, the output of the learner of interval $I \in \text{ACTIVE}(t)$ is randomly selected as the output of the SAOL learner with probability

$$\begin{aligned} \Pr(\hat{M}_t = M_t(I), \hat{\mu}_t = \mu_t(I)) = \frac{w_t(I)}{\sum_{I \in \text{ACTIVE}(t)} w_t(I)}, \\ \forall I \in \text{ACTIVE}(t). \end{aligned} \tag{7}$$

In [7], SAOL assumes that the loss $\ell(\cdot)$ lies between 0 and 1. We propose a way to apply this to our unbounded loss in the next subsection.

5.2 Implementation

We note that [7] does not provide any further implementation details, and that selecting a learner at random can be problematic.

For stochastic mirror descent learners, we propose the following approach. Let each learner be a stochastic composite mirror descent learner (4) having a constant learning rate proportional to the inverse square of the length of the interval, i.e. $\eta_t(I) = \eta_0/\sqrt{|I|}$.

Each mirror descent learner (besides the coarsest) at level j ($|I| = I_0 2^j$) is initialized to the current estimate of the next coarsest learner (level $j-1$). Furthermore, the weight w_t is carried over from said coarser learner. This strategy is equivalent to "backdating" the interval learners so as to ensure appropriate convergence has occurred before the interval of interest is reached, and is effectively a "quantized square root decay" of the learning rate (Figure 3).

In the SAOL framework, the loss must lie between 0 and 1. For convexity reasons, the loss function we use in (3) is unbounded. However, this is a relaxation of the underlying 0-1 loss. Hence for purposes of updating the weights, we use the logistic loss

$$\ell_{t,log}(x_t|M_t,\mu_t) = \text{logistic}\left(\frac{cm_t}{\mu_t}\right) \tag{8}$$

where the argument is scaled by μ_t because only the relative scale of μ_t and $\mathbf{M}_t$ is relevant to the similar/dissimilar boundary. The constant c scales the "buffer region" created by the loss. We set $c = 2$ in all our experiments. Incorporating the logistic loss into (6),

$$\begin{aligned} r_t(I) =& \left(\sum_I \frac{w_t(I)}{W_t}\ell_{t,log}(\mathbf{M}_t(I),\mu_t(I))\right) \\ &- \ell_{t,log}(\mathbf{M}_t(I),\mu_t(I)) \\ w_{t+1}(I) =& w_t(I)(1+\eta_I r_t(I)), \quad \forall t \in I. \end{aligned} \tag{9}$$

In the original SAOL framework, the current estimates are selected randomly. While this gives useful bounds on the expected regret, it means that a known poor estimate is chosen with nonzero probability. We instead propose the following: Choose the I that minimizes the expected total Bregman divergence.

$$\begin{aligned} &\hat{I}(t) = \\ &\operatorname*{arg\,min}_{J \in ACTIVE(t)} \sum_{I \in ACTIVE(t)} B_\psi(\theta_t(I),\theta_t(J))\frac{w_t(I)}{\sum_I w_t(I)}. \end{aligned} \tag{10}$$

If B_ψ is the Frobenius norm, then this is equivalent to choosing the estimate closest to the expectation.

5.3 Performance Guarantees

In the game theory literature, learning rates for stochastic mirror descent techniques have been developed to be able to play dynamic games, i.e., to solve optimization problems that are changing over time [4, 6, 12, 16].

In this section, we parameterize our convex loss as $f_t(\theta_t) = \ell_t(\theta_t) + r(\theta_t)$, where $\theta = [\mathbf{M}, \mu]$. Since the optimal parameter value is changing in a dynamic environment, defining the static regret of an algorithm $\mathcal{B}$ on an interval I as

$$R_{\mathcal{B}}(I) = \sum_{t\in I} f_t(\hat{\theta}_t) - \min_{\theta\in\Theta} \sum_{t\in I} f_t(\theta) \tag{11}$$

is not useful.

A more useful generalization of the standard static regret is as follows. Let $\mathcal{W}$ be a possible set of actions, in this case the set of possible sequences $w = \{\theta_t\}_{t\in I}$ satisfying some criterion. This allows for a dynamically changing estimate. Then, the dynamic regret of an algorithm $\mathcal{B}$ is defined as

$$R_{\mathcal{B}}(I) = \sum_{t\in I} f_t(\hat{\theta}_t) - \min_{w\in\mathcal{W}} \sum_{t\in I} f_t(\theta_t). \tag{12}$$

In [12] the authors define *dynamic regret* by setting $\mathcal{W} = \{w | \sum_{t\in I} \|\theta_{t+1} - \theta_t\| \leq \gamma\}$, i.e. bounding the total amount of variation in the estimated parameter. Without temporal regularization, minimizing the loss would cause θ_t to grossly overfit, hence the constraint on how fast θ_t can change.

We now use this notion of dynamic regret and extend it to the stronger notion of strongly adaptive regret. Following [7], we define strongly adaptive regret of an algorithm $\mathcal{A}$ as

$$\text{SA-Regret}^T_{\mathcal{A}}(\tau) = \max_{I=[q,q+\tau-1]\subset[0,T]} E[R_{\mathcal{A}}(I)] \tag{13}$$

where the expectation is with respect to the possibly random output of the algorithm. We call an algorithm*strongly adaptive* if $\text{SA-Regret}^T_{\mathcal{A}}(\tau) = O(\text{poly}(\log T) R_{\mathcal{P}}(\tau))$, where $R_{\mathcal{P}}(\tau)$ is the regret of the learning problem, i.e. the best possible regret bound.

Low strongly adaptive regret implies that the dynamic regret is low on every subinterval, instead of only low in the aggregate. As a result, a strongly adaptive algorithm must quickly adapt to changes, otherwise the subintervals immediately following the change will not have low regret, even if the total regret over all time is low.

In the supplementary material, we prove the following:

Theorem 1 (SAOML). *Let $\mathcal{W} = \{w | \sum_t \|\theta_{t+1} - \theta_t\| \leq \gamma\}$ and $\mathcal{B}$ be the COMID algorithm of* (4) *with $\eta_t(I) = \eta_0/\sqrt{|I|}$ and fixed μ. Then the strongly adaptive online learner $SAOL^{\mathcal{B}}$ using $\mathcal{B}$ as the black box learners satisfies*

$$R_{SAOL}(I) \leq \frac{4}{2^{1/2}-1} C(1+\gamma)|I|^{1/2} + 40\log(s+1)|I|^{1/2} \tag{14}$$

for some constant C and every interval $I = [q, s]$. In particular, $SAOL^{\mathcal{B}}$ is strongly adaptive.

6 Results

6.1 Synthetic Data

We run our metric learning algorithms on synthetic datasets undergoing different types of simulated metric drift. The first dataset we consider has three classes, with a 50-20-30% split of the prior probability. Each class is associated with a Gaussian blob in 3-dimensional space, with each class having a different mean and covariance. For each of 2000 data points, we select a class at random and generate a 3-dimensional point from that classes' Gaussian distribution. We then embed the 3-dimensional dataset in a random subspace of a 25-dimensional space. The remaining 22-dimensional subspace is filled with iid Gaussian noise.

We generate a series of T constraints from random pairs of points in the dataset, incorporating simulated drift (described below), running each experiment with 1000 random trials. For each experiment conducted in this section, we evaluate performance using three metrics. First the data points in the first two dimensions (as determined by the SVD of $\hat{\mathbf{M}}_T$) of the final learned embedding, color coded according to their true classes are shown. We plot the K-nearest neighbor error rate, using the learned embedding at each time point, averaging over all trials. We quantify the clustering performance by plotting the empirical probability that the normalized mutual information (NMI) of the K-means clustering of the unlabeled data points in the learned embedding at each time point exceeds 0.85 (out of a possible 1). We believe clustering NMI, rather than k-NN performance, is a more realistic indicator of metric learning performance, at least in the case where finding a relevant embedding is the primary goal.

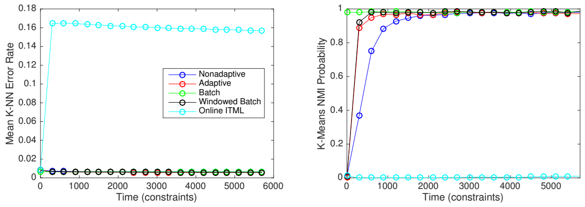

Figure 4: Dataset 1. The dataset remains fixed throughout, no drift occurs as the constraints are observed. Shown as a function of time is the mean k-NN error rate and the probability the k-means NMI > 0.85. Note the failure of ITML and similar performance of the remaining online and batch methods.

Figure 4 shows the static drift-free results for nonadaptive COMID, SAOML, LMNN (batch), our weighted batch method, and online ITML. All parameters were set via cross validation and remain constant through all experiments on the dataset. Online ITML fails due to its bias agains low-rank solutions [8], and the other methods perform comparably as there is no drift. Discrete drift where at time $T/2$ the 25 dimensions are randomly permuted is shown in Figure 5, and continuous drift with a changing rate is shown in Figure 6. To simulate continuous drift, at each time step we perform a small random rotation of the dataset, and at time $T/2$ the rate of rotation is increased by a factor of 6. It can be seen that the weighted batch and especially SAOML respond quickly to drift, performing significantly better than the nonadaptive COMID.

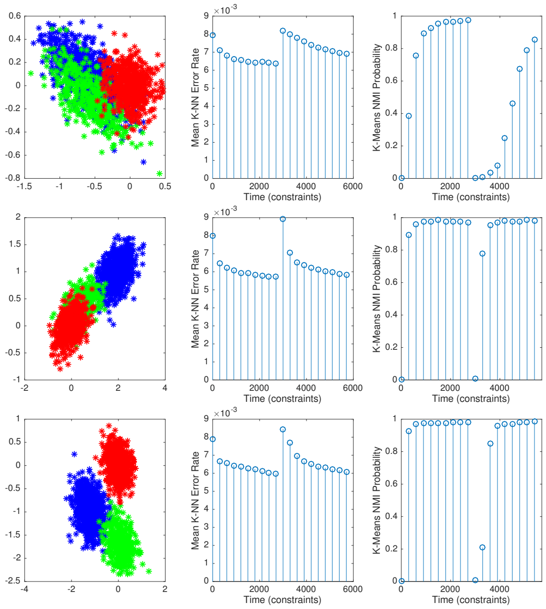

Figure 5: Dataset 1. Random permutation of the data dimensions occurs at the halfway point. Top to bottom: Nonadaptive COMID, adaptive SAOML, and the windowed batch method. Note the slow recovery of the nonadaptive method after the change.

A second dataset identical to the one described above, except with an alternative generative cluster model, was also used. For each of data points, we assign two classes (corresponding to different possible partitions A and B of the data), both selected at random, and for both generate a 3-dimensional point from that classes' Gaussian distribution. The two points are then concatenated into a single 6-dimensional point. We then embed the entire 6-dimensional dataset in a random subspace, with the remaining dimensions filled with iid Gaussian noise as before.

The results for no drift are shown in Figure 7, similar to those found with the first dataset. We also consider drift between partitions (Figure 8): At first, partition A is used, and at time $T/2$, the labeling is changed to partition B. By way of interpretation, the goal of metric learning is to identify the 3-dimensional subspace corresponding to the labeling of interest, and project away the noisy subspaces, thus improving the performance of secondary algorithms. The nonadaptive method fails to quickly catch up to the shift, whereas SAOML effectively increases the learning rate parameter to quickly learn the new paradigm.

6.2 Clustering Product Reviews

As an example real data task, we consider clustering Amazon text reviews, using the Multi-Domain Sentiment Dataset [3]. We use the 11402 reviews from the Electronics

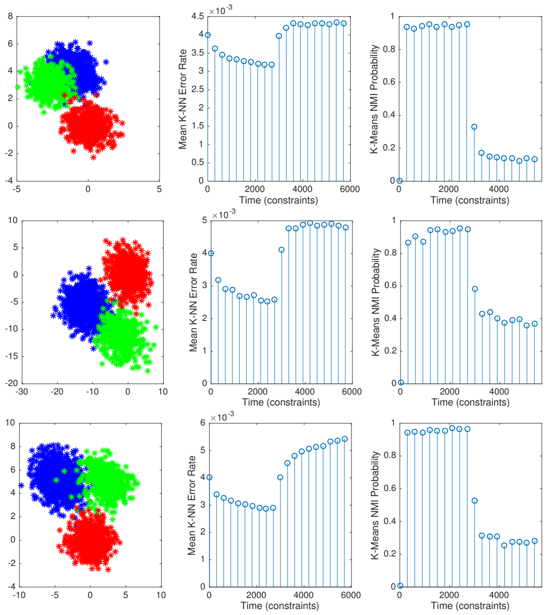

Figure 6: Dataset 1. Continuous slow rotational drift of the dataset occurs, followed by more rapid drift. From top to bottom: Nonadaptive COMID, SAOML, and the weighted batch method. The nonadaptive method with its fixed learning rate performs poorly during rapid drift relative to the adaptive methods.

and Books categories, and preprocess the data by computing word counts for each review and 2369 commonly occurring words. Two possible clusterings of the reviews are considered: product category (books or electronics) and sentiment (positive: star rating 4/5 or greater, or negative: 2/5 or less).

Figures 9 and 10 show the first two dimensions of the embeddings learned by static COMID for the category and sentiment clusterings respectively. Also shown are the 2-dimensional standard PCA embeddings, and the k-NN classification performance both before embedding and in each embeddings. As expected, metric learning is able to find embeddings with improved class separability. We emphasize that while improvements in k-NN classification are observed, we use k-NN merely as a way to quantify the separability of the classes in the learned embeddings. In these experiments, we set the regularizer $r(\cdot)$ to the L1 norm.

We then conducted drift experiments where the clustering changes. The change happens after the metric learner for the original clustering has converged, hence the nonadaptive learning rate is effectively zero. For each change, we show the k-NN error rate in the learned SAOML embedding as it adapts to the new clustering. Emphasizing the visualization and computational advantages of a low-dimensional embedding, we computed the k-NN error after projecting the data into the first 5 dimensions of the embedding. Also shown are the results for a learner where an oracle allows reinitialization of the metric to the identity at time zero, and the nonadaptive learner for which

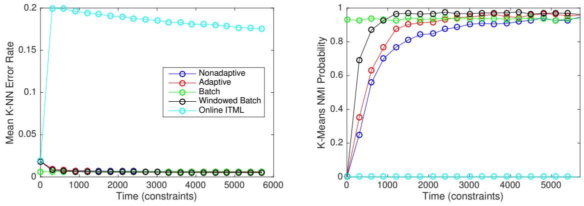

Figure 7: Dataset 2. The dataset and labeling remains fixed throughout, no drift occurs. Shown is the average performance for each method.

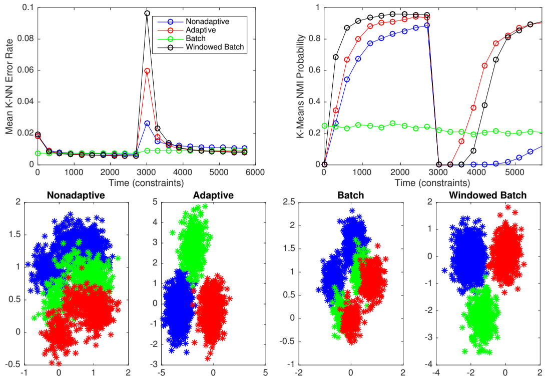

Figure 8: Dataset 2: two possible clusterings of the data exist. For the first half, the first clustering is used to generate the labels, and in the second half a switch is made to the second possible clustering. Top: Average performance; Bottom: an example final embedding for each method. Note the failure of the batch method (LMNN), and the poor performance of the nonadaptive method.

the learning rate is not increased. Figure 11 (left) shows the results when the clustering changes from the four class sentiment + type partition to the two class product type only partition, and Figure 11 (right) shows the results when the partition changes from sentiment to product type. In the first case, the similar clustering allows SAOML to significantly outperform even the reinitialized method, and in the second remain competitive where the clusterings are unrelated.

7 Conclusion and Future Work

We introduced the problem of metric learning in a changing environment, and presented an efficient, strongly adaptive online algorithm having strong theoretical performance guarantees. Performance of our algorithms was evaluated both on synthetic and real datasets, demonstrating the ability of SAOML to learn and adapt quickly in the presence of changes both in the clustering of interest and in the underlying data distribution.

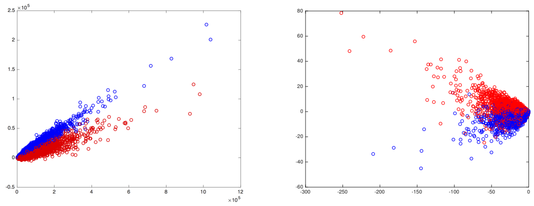

Figure 9: Metric learning for product type clustering. Book reviews blue, electronics reviews red. Original LOO k-NN error rate 15.3%. Left: First two dimensions of learned SAOML embedding (LOO k-NN error rate 11.3%). Right: embedding from standard PCA (k-NN error 20.4%).

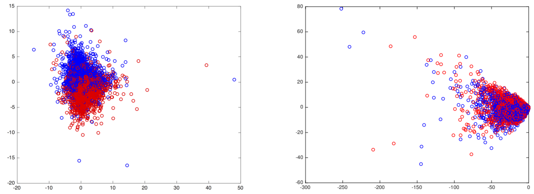

Figure 10: Metric learning for sentiment clustering. Positive reviews blue, negative red. Original LOO k-NN error rate 35.7%. Left: First two dimensions of learned SAOML embedding (LOO k-NN error rate 23.5%). Right: embedding from standard PCA (k-NN error 41.9%).

Potential directions for future work include the learning of more expressive metrics beyond the Mahalanobis metric, the incorporation of unlabeled data points in a semi-supervised learning framework, and the incorporation of an active learning framework to select which pairs of data points to obtain labels for at any given time.

8 Acknowledgments

The Lincoln Laboratory portion of this work was sponsored by the Assistant Secretary of Defense for Research and Engineering under Air Force Contract #FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

References

[1] Mikhail Bilenko, Sugato Basu, and Raymond J Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first International Conference on Machine learning*, page 11. ACM, 2004.

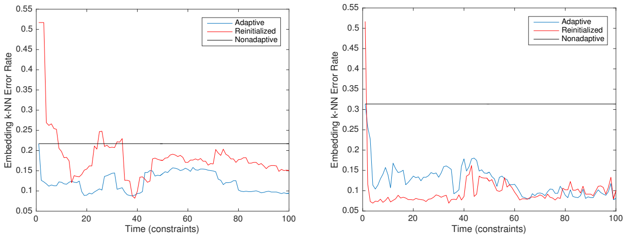

Figure 11: Metric drift in Amazon review data. Left: Change from product type + sentiment clustering to simply product type; Right: Change from sentiment to product type clustering. SAOML performance shown as it adapts to the new clustering, as well as the results for a reinitialized COMID learner and the nonadaptive learner for which the learning rate is not increased.

[2] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] John Blitzer, Mark Dredze, Fernando Pereira, et al. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447, 2007.

[4] Nicolò Cesa-Bianchi, Pierre Gaillard, Gábor Lugosi, and Gilles Stoltz. Mirror descent meets fixed share (and feels no regret). In *Advances in Neural Information Processing Systems*, pages 980–988, 2012.

[5] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[6] Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

[7] Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. Strongly adaptive online learning. *ICML*, 2015.

[8] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

[9] John C Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *COLT*, 2010.

[10] John C Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. In *COLT*, pages 14–26. Citeseer, 2010.

[11] Jacob Goldberger, Geoffrey E Hinton, Sam T Roweis, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *Advances in neural information processing systems*, pages 513–520, 2004.

[12] E.C. Hall and R.M. Willett. Online convex optimization in dynamic environments. *Selected Topics in Signal Processing, IEEE Journal of*, 9(4):647–662, June 2015.

[13] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[14] Elad Hazan and C Seshadhri. Adaptive algorithms for online decision problems. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 14, 2007.

[15] Mark Herbster and Manfred K Warmuth. Tracking the best expert. *Machine Learning*, 32(2):151–178, 1998.

[16] Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization: Competing with dynamic comparators. *arXiv preprint arXiv:1501.06225*, 2015.

[17] Brian Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2012.

[18] Gautam Kunapuli and Jude Shavlik. Mirror descent for metric learning: a unified approach. In *Machine Learning and Knowledge Discovery in Databases*, pages 859–874. Springer, 2012.

[19] John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

[20] H Brendan McMahan. Analysis techniques for adaptive online learning. *arXiv preprint arXiv:1403.3465*, 2014.

[21] H Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. In *COLT*, 2010.

[22] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[23] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing System*, pages 1473–1480, 2005.

[24] Kilian Q Weinberger and Lawrence K Saul. Fast solvers and efficient implementations for distance metric learning. In *International Conference on Machine Learning*, pages 1160–1167. ACM, 2008.

[25] Eric P Xing, Michael I Jordan, Stuart Russell, and Andrew Y Ng. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, pages 505–512, 2002.

[26] Shuheng Zhou, John Lafferty, and Larry Wasserman. Time varying undirected graphs. *Machine Learning*, 80(2-3):295–319, 2010.

Distance Metric Tracking: Supplementary Material

1. Online DML Dynamic Regret

In this section, we derive the dynamic regret of our COMID metric learning algorithm. Recall that the COMID algorithm is given by

$$\begin{aligned}\hat{\mathbf{M}}_{t+1} =& \arg\min_{\mathbf{M}\succeq 0} B_\psi(\mathbf{M},\hat{\mathbf{M}}_t) \\ &+ \eta_t\langle \nabla_M \ell_t(\hat{\mathbf{M}}_t,\mu_t), \mathbf{M}-\hat{\mathbf{M}}_t\rangle + \eta_t\rho\|\mathbf{M}\|_* \\ \hat{\mu}_{t+1} =& \arg\min_{\mu\geq 1} B_\psi(\mu,\hat{\mu}_t) + \eta_t\nabla_\mu \ell_t(\hat{\mathbf{M}}_t,\hat{\mu}_t)'(\mu-\hat{\mu}_t),\end{aligned} \tag{1}$$

where B_ψ is any Bregman divergence and η_t is the learning rate parameter. From (Hall & Willett, 2015) we have:

Theorem 1.

$$\begin{aligned} G_\ell &= \max_{\theta\in\Theta,\ell\in\mathcal{L}} \|\nabla f(\theta)\| \\ \phi_{max} &= \frac{1}{2}\max_{\theta\in\Theta}\|\nabla\psi(\theta)\| \\ D_{max} &= \max_{\theta,\theta'\in\Theta} B_\psi(\theta'\|\theta) \end{aligned}$$

Let the sequence $\hat{\theta}_t = [\hat{\mathbf{M}}_t, \hat{\mu}_t]$, $t = 1,\cdots,T$ be generated via the COMID algorithm, and let w be an arbitrary sequence in $\mathcal{W}\{w|\sum_{t\in I}\|\theta_{t+1}-\theta_t\|\leq\gamma\}$. Then using $\eta_{t+1}\leq\eta_t$ gives

$$R_T(\Theta_T) \leq \frac{D_{max}}{\eta_{T+1}} + \frac{4\phi_{max}}{\eta_T}\gamma + \frac{G_\ell^2}{2\sigma}\sum_{t=1}^T \eta_t \tag{2}$$

Using a decaying learning rate η_t, we can then prove a bound on the dynamic regret for a quite general set of stochastic optimization problems.

Applying this to our problem, we obtain the following. Assume a fixed μ. Then for the estimation of $\mathbf{M}_t$ we have

$$\begin{aligned} G_\ell &= \max_{\|\mathbf{M}\|\leq c,t,\mu}\|\nabla(\ell_t(\mathbf{M},\mu)+\rho\|\mathbf{M}\|_*)\|_2 \\ \phi_{max} &= \frac{1}{2}\max_{\|\mathbf{M}\|\leq c}\|\nabla\psi(\mathbf{M})\|_2 \\ D_{max} &= \max_{\|\mathbf{M}\|,\|\mathbf{M}'\|\leq c} B_\psi(\mathbf{M}'\|\mathbf{M}) \end{aligned}$$

For $\ell_t(\cdot)$ being the hinge loss and $\psi = \|\cdot\|_F^2$,

$$\begin{aligned} G_\ell &\leq \sqrt{(\max_t d^2(\mathbf{x}_t,\mathbf{z}_t)+\rho)^2} \\ \phi_{max} &= c\sqrt{n} \\ D_{max} &= 2c\sqrt{n} \end{aligned}$$

where $d(\mathbf{x},\mathbf{z}) = \|\mathbf{x}-\mathbf{z}\|_2$ is the standard Euclidean distance. The other two quantities are guaranteed to exist and depend on the choice of Bregman divergence and c. Thus,

Corollary 1 (Dynamic Regret: ML COMID)**.** *Let the sequence $\hat{\mathbf{M}}_t, \hat{\mu}_t$ be generated by (1), and let $\{\mathbf{M}_t\}_{t=1}^T$ be an arbitrary sequence with $\|\mathbf{M}_t\|\leq c$ and $\sum_{t=1}^T\|\mathbf{M}_{t+1}-\mathbf{M}_t\|_F\leq\gamma$. Then using $\eta_{t+1}\leq\eta_t$ gives*

$$R_T(\{\mathbf{M}_t\}) \leq \frac{D_{max}}{\eta_{T+1}} + \frac{4\phi_{max}}{\eta_T}\gamma + \frac{G_\ell^2}{2\sigma}\sum_{t=1}^T\eta_t \tag{3}$$

and setting $\eta_t = \eta_0/\sqrt{T}$,

$$\begin{aligned} R_T(\{\mathbf{M}_t\}) \leq& \sqrt{T}\left(\frac{D_{max}+4\phi_{max}V(\{\mathbf{M}_t\})}{\eta_0} + \frac{\eta_0 G_\ell^2}{2\sigma}\right) \\ =& O\left(\sqrt{T}[1+\sum_t\|\mathbf{M}_{t+1}-\mathbf{M}_t\|_F]\right) \end{aligned} \tag{4}$$

for any sequence $\{\theta_t\}$.

Corollary 1 is a bound on the regret relative to the batch estimate of $\mathbf{M}_t$ that minimizes the total batch loss subject to a bounded variation $\sum_t\|\mathbf{M}_{t+1}-\mathbf{M}_t\|_F$. Furthermore, $\eta_t = \eta_0/\sqrt{t}$ gives the same bound as (4).

In other words, we pay a linear penalty on the total amount of variation in the underlying parameter sequence. From (4), it can be seen that the bound-minimizing η_0 increases with increasing $\sum_t\|\mathbf{M}_{t+1}-\mathbf{M}_t\|_F$, indicating the need for an adaptive learning rate.

For comparison, if the metric is in fact static then by standard stochastic mirror descent results (Hall & Willett, 2015)

Theorem 2 (Static Regret)**.** *If $\hat{\mathbf{M}}_1 = 0$ and $\eta_t = (2\sigma D_{max})^{1/2}/(G_f\sqrt{T})$, then*

$$R_T(\{\mathbf{M}_t\}) \leq G_f(2TD_{max}/\sigma)^{1/2}. \tag{5}$$

2. Strongly Adaptive Regret

The following theorem is from (Daniely et al., 2015), slightly modified to accomodate our different definition of strongly adaptive regret.

Theorem 3. *Fix a set $\mathcal{W}$ and choose an algorithm $\mathcal{B}$ such that*

$$R_{\mathcal{B}}(T) \leq CT^\alpha \tag{6}$$

for all $T > 0$ and some constants $\alpha \in (0, 1)$, $C > 0$. Then the strongly adaptive online learner $SAOL^{\mathcal{B}}$ using $\mathcal{B}$ as the black box learners satisfies

$$R^{\mathcal{B}}_{SAOL}(I) \leq \frac{4}{2^{\alpha} - 1} C|I|^{\alpha} + 40 \log(s + 1)|I|^{1/2} \quad (7)$$

for every interval $I = [q, s]$. In particular, $SAOL^{\mathcal{B}}$ will be strongly adaptive if $\alpha \geq \frac{1}{2}$ and $\mathcal{B}$ has low regret.

Apply to low dynamic regret of mirror descent. (4) Corollary 1.

From Corollary 1, COMID with $\eta_t = \eta_0/\sqrt{T}$ satisfies the black-box learner condition (6) with $\alpha = 1/2$. Hence, to apply Theorem 3 to SAOML, it remains to normalize the loss function to between 0 and 1.

As noted in Corollary 1, it is reasonable to assume that $\|\mathbf{M}\| \leq c$. Hence the loss function is bounded by $\ell_t(\mathbf{M}_t, \mu_t) \leq k = \ell(c \max_t \|\mathbf{x}_t - \mathbf{z}_t\|_2^2)$ and can be normalized to the appropriate range. We thus have

Theorem 4 (SAOML)**.** *Let $\mathcal{W} = \{w | \sum_t \|\theta_{t+1} - \theta_t\| \leq \gamma\}$ and $\mathcal{B}$ be the COMID algorithm of (1) with $\eta_t(I) = \eta_0/\sqrt{|I|}$ and fixed μ. Then the strongly adaptive online learner $SAOL^{\mathcal{B}}$ using $\mathcal{B}$ as the black box learners satisfies*

$$R_{SAOL}(I) \leq \frac{4}{2^{1/2} - 1} C(1 + \gamma)|I|^{1/2} + 40 \log(s + 1)|I|^{1/2} \quad (8)$$

for some constant C and every interval $I = [q, s]$. In particular, $SAOL^{\mathcal{B}}$ is strongly adaptive.

We note that this bound is stronger than those considered in (Daniely et al., 2015) as it incorporates dynamic regret in the definition of strongly adaptive regret.

References

Daniely, Amit, Gonen, Alon, and Shalev-Shwartz, Shai. Strongly adaptive online learning. *ICML*, 2015.

Hall, E.C. and Willett, R.M. Online convex optimization in dynamic environments. *Selected Topics in Signal Processing, IEEE Journal of*, 9(4):647–662, June 2015.

This material is based upon work supported by the Assistant Secretary of Defense for Research and Engineering under Air Force Contract No. FA8721-05-C-0002 and/or FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Assistant Secretary of Defense for Research and Engineering.