Stochastic Dykstra Algorithms for Metric Learning on Positive Semi-Definite Cone

Tomoki Matsuzawa[†], Raissa Relator[◇], Jun Sese[◇], Tsuyoshi Kato[†,‡,*]

[†] Faculty of Science and Engineering, Gunma University, Kiryu-shi, Gunma, 326–0338, Japan.
[‡] Center for Informational Biology, Ochanomizu University, Bunkyo-ku, Tokyo, 112–8610, Japan. n
[◇] BRD, AIST, Koto-ku, Tokyo, 135–0064, Japan.

Abstract

Recently, covariance descriptors have received much attention as powerful representations of set of points. In this research, we present a new metric learning algorithm for covariance descriptors based on the Dykstra algorithm, in which the current solution is projected onto a half-space at each iteration, and runs at $O(n^3)$ time. We empirically demonstrate that randomizing the order of half-spaces in our Dykstra-based algorithm significantly accelerates the convergence to the optimal solution. Furthermore, we show that our approach yields promising experimental results on pattern recognition tasks.

1 Introduction

Learning with example objects characterized by a set of several points, instead of a single point, in a feature space is an important task in the computer vision and pattern recognition community. In the case of visual categorization of still images, many local image descriptors such as SIFT [17] are extracted from an input image to form a single vector such as a Bag-of-Visual-Words vector or a Fisher Vector [18, 20]. For image set classification, a surge of methods have been developed in the last decade, and probabilistic models [25] or kernels [22] are introduced to describe the image set. Alternative descriptors are the covariance descriptors, which have received much attention as a powerful representation of a set of points.

The performance of categorizing covariance descriptors depends on the metric that is used to measure the distances between them. To compare covariance descriptors, a variety of distance measures such as affine invariant Riemannian metric [19], Stein metric [27], J-divergence [29], Frobenius distance [11], and Log-Frobenius distance [1], have been discussed in existing literature. Some of them are designed from their geometrical properties, but some are not. Many of these distance measures are expressed in the form

$$D_{\boldsymbol{\Phi}}(\boldsymbol{X}_1, \boldsymbol{X}_2) := \|\boldsymbol{\Phi}(\boldsymbol{X}_1) - \boldsymbol{\Phi}(\boldsymbol{X}_2)\|_{\mathrm{F}}^2 ,$$

with $\boldsymbol{\Phi} : \mathbb{S}_{++}^n \to \mathbb{R}^{n\times m}$ for some $m \in \mathbb{N}$. If $\boldsymbol{\Phi}(\boldsymbol{X}) := \mathrm{logm}(\boldsymbol{X})$, where $\mathrm{logm}(\boldsymbol{X})$ takes the principal matrix logarithm of a strictly positive definite matrix $\boldsymbol{X}$, the Log-Frobenius distance [1] is obtained. Setting $\boldsymbol{\Phi}(\boldsymbol{X}) := \boldsymbol{X}^p$ gives the Power-Frobenius distance [11], while $\boldsymbol{\Phi}(\boldsymbol{X}) := \mathrm{chol}(\boldsymbol{X})$, where $\mathrm{chol} : \mathbb{S}_{+}^n \to \mathbb{R}^{n\times n}$ produces the Cholesky decomposition of $\boldsymbol{X}$ such that $\boldsymbol{X} = \mathrm{chol}(\boldsymbol{X})\mathrm{chol}(\boldsymbol{X})^\top$, yields the Cholesky-Frobenius distance [8]. These metrics are pre-defined before the employment of machine learning algorithms, and are not adaptive to the data to be analyzed. Meanwhile, for categorization of vectorial data, supervised learning for fitting metrics to the task has been proven to significantly increase the performance of the distance-based classifier [6, 13, 23].

In this paper, we introduce a parametric distance measure between covariance descriptors and present novel metric learning algorithms to determine the parameters of the distance measure function. The learning problem is formulated as the Bregman projection onto the intersections of half-spaces. This kind of problem can be solved by the Dykstra algorithm [4, 9], which chooses a single half-space in a cyclic order and projects a current solution to the half-space. We developed an efficient technique for projection onto a single half-space. Furthermore, we empirically found that selecting the half-space stochastically, rather than in a cyclic order, dramatically increases the speed of converging to an optimal solution.

1.1 Related work

To the best of our knowledge, Vemulapalli et al. (2015) [28] were the first to introduce the supervised metric learning approach for covariance descriptors. They vectorized the matrix logarithms of the covariance descriptors to apply existing metric learning methods to the vectorizations of matrices. The dimensionality of the vectorizations is $n(n+1)/2$ when the size of the covariance matrices are $n \times n$. Thus, the size of the Mahalanobis matrix is $n(n+1)/2 \times n(n+1)/2$, which is computationally prohibitive when n is large.

Our approach is an extension of the distance measure of Huang et al. [10], which is based on the Log-Euclidean metric, with their loss function being a special case of our formulation. They also adopted the cyclic Dykstra algorithm for learning the Mahalanobis-like matrix. However, they misused the Woodbury matrix inversion formula when deriving the projection onto a single half-space, therefore, their algorithm has no theoretical guarantee of converging to the optimal solution. In this paper, their update rule is corrected by presenting a new technique that projects a current solution to a single half-space within $O(n^3)$ computational time.

Yger and Sugiyama [30] devised a different formulation of metric learning. They introduced the congruent transform [2] and measures distances between the transformations of covariance descriptors. An objective function based on the kernel target alignment [5] is employed to determine the transformation parameters. Compared to their algorithm, our algorithm has the capability to monitor the upper bound of the objective gap, i.e. the difference between the current objective and the minimum. This implies that the resultant solution is ensured to be ϵ-suboptimal if the algorithm's convergence criterion is set such that the objective gap upper bound is less than a very small number ϵ. Since Yger and Sugiyama [30] employed a gradient method for learning the congruent transform, there is no way to know the objective gap.

1.2 Contributions

Our contributions of this paper can be summarized as follows.

- For metric learning on positive semidefinite cone, we developed a new algorithm based on the Dykstra algorithm, in which the current solution is projected onto a half-space at each iterate, and runs at $O(n^3)$ time.
- We present an upper-bound for the objective gap which provides a stopping criterion and ensures the optimality of the solution.
- We empirically found that randomizing the order of half-spaces in our Dykstra-based algorithm significantly accelerates the convergence to the optimal solution.
- We show that our approach yields promising experimental results on pattern recognition tasks.

1.3 Notation

We denote vectors by bold-faced lower-case letters and matrices by bold-faced upper-case letters. Entries of vectors and matrices are not bold-faced. The transposition of a matrix $\boldsymbol{A}$ is denoted by $\boldsymbol{A}^\top$, and the inverse of $\boldsymbol{A}$ is by $\boldsymbol{A}^{-1}$. The $n \times n$ identity matrix is denoted by $\boldsymbol{I}_n$. The subscript is often omitted. The $m \times n$ zero matrix is denoted by $\boldsymbol{O}_{m \times n}$. The subscript is often omitted. The n-dimensional vector all of whose entries are one is denoted by $\mathbf{1}_n$. We use $\mathbb{R}$ and $\mathbb{N}$ to denote the set of real and natural numbers, $\mathbb{R}^n$ and $\mathbb{N}^n$ to denote the set of n-dimensional real and natural vectors, and $\mathbb{R}^{m \times n}$ to denote the set of $m \times n$ real matrices. For any $n \in \mathbb{N}$, we use $\mathbb{N}_n$ to denote the set of natural numbers less than or equal to n. Let us define $\mathbb{R}_+ := \{x \in \mathbb{R} \,|\, x \geq 0\}$, $\mathbb{R}_{++} := \{x \in \mathbb{R} \,|\, x > 0\}$, $\mathbb{R}^n_+ := \{x \in \mathbb{R}^n \,|\, \boldsymbol{x} \geq \mathbf{0}_p\}$, and $\mathbb{R}^n_{++} := \{x \in \mathbb{R}^n \,|\, \boldsymbol{x} > \mathbf{0}_p\}$. The relational operator $\succ$ denotes the generalized inequality associated with the strictly positive definite cone. We use $\mathbb{S}^n$ to denote the set of symmetric $n \times n$ matrices. $\mathbb{S}^n_+$ to denote the set of symmetric positive semi-definite $n \times n$ matrices, and $\mathbb{S}^n_{++}$ to denote the set of symmetric strictly positive definite $n \times n$ matrices. For any $\boldsymbol{x} = [x_1, \ldots, x_n]^\top \in \mathbb{R}^n$, $\mathrm{diag}(\boldsymbol{x})$ is defined as an $n \times n$ diagonal matrix whose diagonal entries are $x_1, \ldots, x_n$. For any $n \times n$ square matrix $\boldsymbol{X}$, its trace is denoted by $\mathrm{tr}(\boldsymbol{X})$. For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, define $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \sum_{i=1}^n x_i y_i$ where x_i and y_i is the i-th entry of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. For any $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{m \times n}$, define $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle := \sum_{i=1}^m \sum_{j=1}^n X_{i,j} Y_{i,j}$ where $X_{i,j}$ and $Y_{i,j}$ is the (i,j)-th entry of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. $\mathbb{O}_n$ is used to denote the set of $n \times n$ orthonormal matrices, i.e. $\mathbb{O}_n := \{\boldsymbol{A} \in \mathbb{R}^{n \times n} \,|\, \boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{I}_n\}$.

2 Our Metric Learning Problem

2.1 Parametric distance measure on $\mathbb{S}^n_+$

We introduce the following distance measure for covariance descriptors $\boldsymbol{X}_1, \boldsymbol{X}_2 \in \mathbb{S}^n_+$:

$$D_{\boldsymbol{\Phi}}(\boldsymbol{X}_1, \boldsymbol{X}_2; \boldsymbol{W}) := \left\langle \boldsymbol{W}, (\boldsymbol{\Phi}(\boldsymbol{X}_1) - \boldsymbol{\Phi}(\boldsymbol{X}_2)) (\boldsymbol{\Phi}(\boldsymbol{X}_1) - \boldsymbol{\Phi}(\boldsymbol{X}_2))^\top \right\rangle,$$

where $\boldsymbol{W} \in \mathbb{S}^n_+$ is the parameter of this distance measure function. If $\boldsymbol{W}$ is strictly positive definite and $\boldsymbol{\Phi}$ is bijective, then this distance measure $D_{\boldsymbol{\Phi}}(\cdot, \cdot; \boldsymbol{W}) : \mathbb{S}^n_+ \times \mathbb{S}^n_+ \to \mathbb{R}$ is a metric because all of the following conditions are satisfied:(i) non-negativity: $D_{\boldsymbol{\Phi}}(\boldsymbol{X}_1, \boldsymbol{X}_2; \boldsymbol{W}) \geq 0$; (ii) identity of indiscernibles: $D_{\boldsymbol{\Phi}}(\boldsymbol{X}_1, \boldsymbol{X}_2; \boldsymbol{W}) = 0$ iff $\boldsymbol{X}_1 = \boldsymbol{X}_2$; (iii) symmetry: $D_{\boldsymbol{\Phi}}(\boldsymbol{X}_1, \boldsymbol{X}_2; \boldsymbol{W}) = D_{\boldsymbol{\Phi}}(\boldsymbol{X}_2, \boldsymbol{X}_1; \boldsymbol{W})$; (iv) triangle inequality: $D_{\boldsymbol{\Phi}}(\boldsymbol{X}_1, \boldsymbol{X}_3; \boldsymbol{W}) \leq D_{\boldsymbol{\Phi}}(\boldsymbol{X}_1, \boldsymbol{X}_2; \boldsymbol{W}) + D_{\boldsymbol{\Phi}}(\boldsymbol{X}_2, \boldsymbol{X}_3; \boldsymbol{W})$.

If the parameter matrix $\boldsymbol{W}$ is singular, $D_{\boldsymbol{\Phi}}(\cdot,\cdot;\boldsymbol{W})$ is a pseudometric, and the identity of indiscernibles is changed to the following property: For any $\boldsymbol{X}_1 \in \mathbb{S}^n_{++}$, $D_{\boldsymbol{\Phi}}(\boldsymbol{X}_1, \boldsymbol{X}_1; \boldsymbol{W}) = 0$ holds, while $D_{\boldsymbol{\Phi}}(\boldsymbol{X}_1, \boldsymbol{X}_2; \boldsymbol{W}) = 0$ occurs for some non-identical positive semi-definite matrices $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$.

2.2 Formulations of the learning problems

To determine the value of the parameter matrix $\boldsymbol{W}$, we pose a constrained optimization problem based on the idea of ITML [6]. We now consider a multi-class categorization problem. Let n_c be the number of classes, and the class labels are represented by natural numbers in $\mathbb{N}_{n_c}$. Suppose weare given

$$(\boldsymbol{X}_1, \omega_1), \ldots, (\boldsymbol{X}_\ell, \omega_\ell) \in \mathbb{S}^n_+ \times \mathbb{N}_{n_c}$$

as a training dataset, where $\boldsymbol{X}_i$ is the covariance descriptor of the i-th example, and ω_i is its class label. From the ℓ examples, K pairs $(i_1, j_1), \ldots, (i_K, j_K) \in \mathbb{N}_\ell \times \mathbb{N}_\ell$ are picked to give, to each pair, the following constraint:

$$D_{\boldsymbol{\Phi}}(\boldsymbol{X}_{i_k}, \boldsymbol{X}_{j_k}; \boldsymbol{W}) \begin{cases} \leq b_{\mathrm{ub}} \xi_k, & \text{if } \omega_{i_k} = \omega_{j_k}, \\ \geq b_{\mathrm{lb}} \xi_k, & \text{if } \omega_{i_k} \neq \omega_{j_k}, \end{cases} \tag{1}$$

where, when $\xi_k = 1$, the two constants b_{ub} and b_{lb}, respectively, are the upper-bound of the distances between any two examples in the same class and the lower-bound of the distances between any two examples in different classes. Now let us define for $k \in \mathbb{N}_K$,

$$y_k := \begin{cases} +1, & \text{if } \omega_{i_k} = \omega_{j_k}, \\ -1, & \text{if } \omega_{i_k} \neq \omega_{j_k}, \end{cases} \qquad \text{and} \qquad b_k := \begin{cases} b_{\mathrm{ub}}, & \text{if } \omega_{i_k} = \omega_{j_k}, \\ b_{\mathrm{lb}}, & \text{if } \omega_{i_k} \neq \omega_{j_k}. \end{cases}$$

Under the constraint (1), we wish to find $\boldsymbol{W}$ and ξ_k such that $\boldsymbol{W}$ is not much deviated from the identity matrix and ξ_k is close to one. From this motivation, we pose the following problem:

$$\begin{aligned} \min \quad & \mathrm{BD}_\varphi((\boldsymbol{W}, \boldsymbol{\xi}), (\boldsymbol{I}, \boldsymbol{1})), \qquad \text{wrt} \quad \boldsymbol{W} \in \mathbb{S}^n_{++}, \quad \boldsymbol{\xi} = [\xi_1, \ldots, \xi_K]^\top \in \mathbb{R}^K_{++}, \\ \text{subject to} \quad & \forall k \in \mathbb{N}_K, \quad y_k D_{\boldsymbol{\Phi}}(\boldsymbol{X}_{i_k}, \boldsymbol{X}_{j_k}; \boldsymbol{W}) \leq y_k b_k \xi_k, \end{aligned} \tag{2}$$

where $\mathrm{BD}_\varphi(\cdot,\cdot) : (\mathbb{S}^n_{++} \times \mathbb{R}^K_{++}) \times (\mathbb{S}^n_{++} \times \mathbb{R}^K_{++}) \to \mathbb{R}_+$ is the *Bregman divergence* [14]. Only if $(\boldsymbol{W}, \boldsymbol{\xi}) = (\boldsymbol{I}, \boldsymbol{1})$ will the divergence $\mathrm{BD}_\varphi((\boldsymbol{W}, \boldsymbol{\xi}), (\boldsymbol{I}, \boldsymbol{1}))$ become zero, and the value of divergence becomes larger if $(\boldsymbol{W}, \boldsymbol{\xi})$ is more deviated from $(\boldsymbol{I}, \boldsymbol{1})$. The definition of the Bregman divergence contains a seed function $\varphi : \mathbb{S}^n_{++} \times \mathbb{R}^K_{++} \to \mathbb{R}$ which is assumed to be continuously differentiable and strictly convex. For some φ, the Bregman divergence is defined as

$$\mathrm{BD}_\varphi(\boldsymbol{\Theta}, \boldsymbol{\Theta}_0) = \varphi(\boldsymbol{\Theta}) - \varphi(\boldsymbol{\Theta}_0) - \langle \nabla\varphi(\boldsymbol{\Theta}_0), \boldsymbol{\Theta} - \boldsymbol{\Theta}_0 \rangle,$$

for $\boldsymbol{\Theta}, \boldsymbol{\Theta}_0 \in \mathbb{S}^n_{++} \times \mathbb{R}^K_{++}$. This implies that the quantities of the deviations of the solution $(\boldsymbol{W}, \boldsymbol{\xi})$ from $(\boldsymbol{I}, \boldsymbol{1})$ depend on the definition of the seed function. In this study, the seed function is assumed to be the sum of two terms:

$$\varphi(\boldsymbol{W}, \boldsymbol{\xi}) := \varphi_{\mathrm{r}}(\boldsymbol{W}) + \sum_{k=1}^K c_k \varphi_{\mathrm{l}}(\xi_k),$$

where c_k is a positive constant. The first term $\varphi_{\mathrm{r}} : \mathbb{S}^n_{++} \to \mathbb{R}$ in the definition of the seed function is defined by $\varphi_{\mathrm{r}}(\boldsymbol{W}) := -\mathrm{logdet}(\boldsymbol{W})$.As for the definition of the second term$\varphi_{\mathrm{l}} : \mathbb{R}_{++} \to \mathbb{R}$, we considered the following three functions:

$$\varphi_{\mathrm{is}}(\xi_k) := -\log(\xi_k), \qquad \varphi_{\mathrm{l2}}(\xi_k) := \frac{1}{2}\xi_k^2, \qquad \varphi_{\mathrm{e}}(\xi_k) := (\log \xi_k - 1)\xi_k.$$

The Bregman divergences generated from three seed functions φ_{is}, φ_{l2}, and φ_{e}, respectively, are referred to as *Itakura-Saito Bregman Divergence* (ISBD), *L2 Bregman Divergence* (L2BD), and *Relative Entropy Bregman Divergence* (REBD), where ISBD is equal to the objective function employed by Huang et al. [10].

3 Stochastic Variants of Dykstra Algorithm

We introduce the Dykstra algorithm [4, 9] to solve the optimization problem (2). The original Dykstra algorithm [9] was developed as a computational method that finds the Euclidean projection from a point onto the intersection of

convex sets. Censor & Reich [4] extended the algorithm to finding the Bregman projection from a point $\boldsymbol{x}_0$ to a set $\mathcal{C}$, defined by

$$\operatorname*{argmin}_{\boldsymbol{x}\in\mathcal{C}} \mathrm{BD}_\varphi(\boldsymbol{x}, \boldsymbol{x}_0).$$

In available literature related to stochastic gradient descent methods and the variants [3, 12, 24, 26] that minimize the regularized loss averaged over a set of examples, it is empirically shown that, rather than picking an example in a cyclic order, example selection in a stochastic order dramatically speeds up the convergence to the optimal solution. Alternatively, some literature reported that at the beginning of every epoch in the gradient method, random permutation of the order of examples also accelerates the convergence [7].

Motivated by these facts, this study proposes the use of stochastic orders for selection of convex set components in the Dykstra algorithm. We term the stochastic version of the Dykstra algorithm as the *stochastic Dykstra algorithm*. In our case, every convex set component is one of K half-spaces, as will be described in a later discussion. There are, then, three ways to select half-spaces:

- **Cyclic**: Pick a half-space in a cyclic order at each iteration.
- **Rand**: Pick a half-space randomly at each iteration.
- **Perm**: Permute the order of K half-spaces randomly at the beginning of each epoch.

Hereinafter, we assume to employ the "Rand" option, although replacing this option with one of the remaining two is straightforward.

If every convex set component is a half-space, and the k-th convex set component $\mathcal{C}_k$ is expressed as

$$\mathcal{C}_k := \left\{\boldsymbol{x} \,|\, \langle \boldsymbol{a}_k, \boldsymbol{x}\rangle \leq b_k\right\},$$

then computing the Bregman projection from a point $\boldsymbol{x}_0$ to its boundary $\mathrm{bd}(\mathcal{C}_k)$ is equivalent to solving the following saddle point problem:

$$\max_{\delta}\min_{\boldsymbol{x}} \mathrm{BD}_\varphi(\boldsymbol{x}, \boldsymbol{x}_0) + \delta(\langle \boldsymbol{a}_k, \boldsymbol{x}\rangle - b_k).$$

This fact enables us to rewrite the Dykstra algorithm with Rand option for finding the Bregman projection from a point $\boldsymbol{x}_0$ to the intersection of $\mathcal{C}_1, \ldots, \mathcal{C}_K$, as described in Algorithm 1, where φ^* is the convex conjugate of the seed function φ.

Algorithm 1 Stochastic Dykstra Algorithm.

begin
$\forall k \in \mathbb{N}_K :\ \alpha_k := 0$;
for $t = 1, 2, \ldots$ **do**
  Pick k randomly from $\{1, \ldots, K\}$;
  Solve the following saddle point problem and let $\delta_{t-1/2}$ be the solution of δ:

$$\max_{\delta}\min_{\boldsymbol{x}} \mathrm{BD}_\varphi(\boldsymbol{x}, \boldsymbol{x}_{t-1}) + \delta_t(\langle \boldsymbol{a}_k, \boldsymbol{x}\rangle - b_k); \tag{3}$$

  $\delta_t := \max(\delta_{t-1/2}, -\alpha_k)$; $\alpha_k := \alpha_k + \delta_t$;
  $\boldsymbol{x}_t = \nabla\varphi^*(\nabla\varphi(\boldsymbol{x}_{t-1}) - \delta_t \boldsymbol{a}_k)$;
end for
end.

4 Efficient Projection Technique

We now show that solving the optimization problem (2) is equivalent to finding a Bregman projection from a point $(\boldsymbol{I}, \boldsymbol{1}) \in \mathbb{S}^n_{++} \times \mathbb{R}^K_{++}$ onto the intersection of multiple half-spaces.

Let $\boldsymbol{A}_k$ be a positive semidefinite matrix expressed as

$$\boldsymbol{A}_k := \left(\boldsymbol{\Phi}(\boldsymbol{X}_{i_k}) - \boldsymbol{\Phi}(\boldsymbol{X}_{j_k})\right)\left(\boldsymbol{\Phi}(\boldsymbol{X}_{i_k}) - \boldsymbol{\Phi}(\boldsymbol{X}_{j_k})\right)^\top$$

for $k \in \mathbb{N}_K$, to define a half-space

$$\mathcal{C}_k := \left\{(\boldsymbol{W}, \boldsymbol{\xi}) \in \mathbb{S}^n_{++} \times \mathbb{R}^K_{++} \,|\, y_k \langle \boldsymbol{A}_k, \boldsymbol{W}\rangle - y_k b_k \xi_k \leq 0\right\}.$$

Then, it can be seen that the intersection of K half-spaces

$$\bigcap_{k=1}^{K} \mathcal{C}_k$$

is the feasible region of the optimization problem (2). This implies that the Dykstra algorithm can be applied to solve problem (2).

Next we present an efficient technique that projects $(\boldsymbol{W}_{t-1}, \boldsymbol{\xi}_{t-1}) \in \mathbb{S}^n_{++} \times \mathbb{R}^K_{++}$ onto the k-th half-space $\mathcal{C}_k$, where $(\boldsymbol{W}_{t-1}, \boldsymbol{\xi}_{t-1}) \in \mathbb{S}^n_{++} \times \mathbb{R}^K_{++}$ is the model parameter after the $(t-1)$-th iteration. Let $\xi_{k,t-1}$ be the k-th entry in the vector $\boldsymbol{\xi}_{t-1}$. The value of the function $J_t : \mathbb{R} \to \mathbb{R}$ defined by

$$J_t(\delta) := \left\langle \boldsymbol{A}_k, (\boldsymbol{W}_{t-1}^{-1} + \delta y_k \boldsymbol{A}_k)^{-1} \right\rangle - b_k \nabla \varphi_1^*(\nabla \varphi_1(\xi_{t-1}) + \delta y_k b_k / c_k),$$

is zero at the solution $\delta_{t-1/2}$ of the saddle point problem (3). The solution δ must satisfy the strictly positive definiteness:

$$(\boldsymbol{Y}(\delta))^{-1} := \boldsymbol{W}_{t-1}^{-1} + \delta y_k \boldsymbol{A}_k \succ \boldsymbol{O}, \tag{4}$$

and the feasibility of the slack variables:

$$\exists\, \xi_{k,t-1/2} \quad \text{s.t.} \quad \nabla \varphi_1(\xi_{k,t-1/2}) = \nabla \varphi_1(\xi_{k,t-1}) - \delta y_k b_k / c_k. \tag{5}$$

There is no closed-form solution found for this projection problem. Hence, some numerical method such as the Newton-Raphson method is necessary for solving the nonlinear system $J_t(\delta) = 0$. If one tries to compute the value of $J_t(\delta)$ naïvely, it will require an $O(n^3)$ computational cost because $J_t(\cdot)$ involves computation of the inverse of an $n \times n$ matrix. If we suppose the numerical method assesses the value of the scalar-valued function $J_t(\cdot)$ L times, the naïve approach will take $O(Ln^3)$ computational time to find the solution of the nonlinear system $J_t(\delta) = 0$. Furthermore, the positive definiteness condition in (4) and the feasibility condition in (5) must be checked.

We will show the following two claims:

- The solution of the system $J_t(\delta) = 0$ satisfying (4) and (5) can be computed within $O(n^3 + Ln)$ time, where L is the number of times a numerical method assesses the value of $J_t(\cdot)$.

- The solution exists and it is unique.

Hereinafter, we assume $\boldsymbol{A}_k$ is strictly positive definite. By setting $\boldsymbol{A}_k \leftarrow \boldsymbol{A}_k + \epsilon \boldsymbol{I}$, with ϵ as a small positive constant, it is easy to satisfy this assumption. Since $L \in O(n^2)$ in a typical setting, we can say that each update can be done in $O(n^3)$ computation.

We define $\boldsymbol{A}_k^{1/2}$, $\boldsymbol{U}$, $\boldsymbol{D}$, and $\boldsymbol{d}$ as follows. Let $\boldsymbol{A}_k^{1/2} \in \mathbb{S}^n_{++}$ such that $\boldsymbol{A}_k^{1/2} \boldsymbol{A}_k^{1/2} = \boldsymbol{A}_k$, and denote by $\boldsymbol{A}_k^{-1/2} \in \mathbb{S}^n_{++}$ the inverse of $\boldsymbol{A}_k^{1/2}$. Introduce an orthonormal matrix $\boldsymbol{U} \in \mathbb{O}_n$ and a diagonal matrix $\boldsymbol{D} = \mathrm{diag}(\{d_1, \ldots, d_n\})$ that represent a spectral decomposition $\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\top = \boldsymbol{A}^{-1/2} \boldsymbol{W}_{t-1}^{-1} \boldsymbol{A}^{-1/2}$, with $d_1 \geq \cdots \geq d_n$. Then, we have

$$\boldsymbol{Y}(\delta) = \boldsymbol{A}_k^{-1/2} \boldsymbol{U} (\boldsymbol{D} + y_k \delta \boldsymbol{I})^{-1} \boldsymbol{U}^\top \boldsymbol{A}_k^{-1/2}, \tag{6}$$

which allows us to rewrite the first term of $J_t(\delta)$ as

$$\left\langle \boldsymbol{A}_k, (\boldsymbol{W}_{t-1}^{-1} + \delta y_k \boldsymbol{A}_k)^{-1} \right\rangle = \sum_{i=1}^{n} \frac{1}{d_i + y_k \delta}. \tag{7}$$

Assessment of $J_t(\delta)$ can be done within $O(n)$ computational cost after $d_1, \ldots, d_n$ are obtained. To get the n scalars $d_1, \ldots, d_n$, we need to find $\boldsymbol{A}_k^{-1/2}$ and the spectral decomposition of $\boldsymbol{A}_k^{-1/2} \boldsymbol{W}_{t-1}^{-1} \boldsymbol{A}_k^{-1/2}$, each of which requires $O(n^3)$ computation. The $n \times n$ matrix $\boldsymbol{A}_k^{-1/2}$ can be computed in the pre-process of the Dykstra algorithm, while the spectral decomposition of $\boldsymbol{A}_k^{-1/2} \boldsymbol{W}_{t-1}^{-1} \boldsymbol{A}_k^{-1/2}$ is done once before invoking some numerical method to solve the nonlinear system $J_t(\delta) = 0$. These support the first claim.

Equation (6) suggests that the set of δ satisfying (4) is given by

$$I_{\mathrm{r},t} := \begin{cases} (-d_n, +\infty), & \text{for } y_k = +1, \\ (-\infty, d_n), & \text{for } y_k = -1. \end{cases} \tag{8}$$

The set of δ satisfying (5) is given as follows. In the case of using ISBD, δ ensuring (5) is in the interval

$$I_{\mathrm{is},t} := \begin{cases} (-\infty, \delta_{\mathrm{b}}), & \text{for } y_k = +1, \\ (-\delta_{\mathrm{b}}, +\infty), & \text{for } y_k = -1, \end{cases}$$

where

$$\delta_{\mathrm{b}} := \frac{c_k}{b_k \xi_{k,t-1}}.$$

In the case of using L2BD and REBD, there exists $\xi_{t-1/2}$ even if $\nabla\varphi_{\mathrm{l}}(\xi_{t-1}) - \delta y_k b_k / c_k$ takes any value.

Hence, if ISBD is employed, the solution $\delta_{t-1/2}$ can be searched from the interval

$$I_{\mathrm{r},t} \cap I_{\mathrm{is},t} = \begin{cases} (-d_n, \delta_{\mathrm{b}}), & \text{for } y_k = +1, \\ (-\delta_{\mathrm{b}}, +d_n), & \text{for } y_k = -1. \end{cases}$$

If L2BD or REBD is employed, the solution $\delta_{t-1/2}$ can be searched from $I_{\mathrm{r},t}$. In the reminder of this section, we shall use the notation I_t to denote the interval for δ satisfying (4) and (5) simultaneously.

We now show the uniqueness of the solution. The gradient of $J_t : \mathbb{R} \to \mathbb{R}$ is expressed as

$$\nabla J_t(\delta) = -\sum_{i=1}^{n} \frac{y_k}{(d_i + y_k \delta)^2} - \frac{y_k b_k^2}{c_k} \nabla^2 \varphi_{\mathrm{l}}^*(\nabla\varphi_{\mathrm{l}}(\xi_{t-1}) - \delta y_k b_k / c_k),$$

for $\delta \in I_t$. We first consider the case that $y_k = +1$. Clearly, the first term is negative. The second term is non-positive because any convex conjugate function is convex. Therefore, we have $\nabla J_t(\delta) < 0$. In the case of $y_k = -1$, we get $\nabla J_t(\delta) > 0$ from a similar derivation. These observations imply that the solution is unique if a solution exists. The existence of the solution can be established by showing that the curve $J_t(\delta)$ crosses the horizontal axis.

We consider the cases of using ISBD and using either L2BD or REBD separately. For the ISBD case, we have

$$\lim_{\delta \searrow -d_n} J_t(\delta) = +\infty, \qquad \lim_{\delta \nearrow \delta_{\mathrm{b}}} J_t(\delta) = -\infty,$$

if $y_k = +1$, and

$$\lim_{\delta \searrow -\delta_{\mathrm{b}}} J_t(\delta) = -\infty, \qquad \lim_{\delta \nearrow d_n} J_t(\delta) = +\infty,$$

if $y_k = -1$. On the other hand, when using either L2BD or REBD with $y_k = +1$ we get

$$\lim_{\delta \searrow -d_n} J_t(\delta) = +\infty, \qquad \lim_{\delta \to +\infty} J_t(\delta) = -\infty,$$

while we obtain

$$\lim_{\delta \to -\infty} J_t(\delta) = -\infty, \qquad \lim_{\delta \nearrow d_n} J_t(\delta) = +\infty,$$

when $y_k = -1$. Hence, we conclude that

$$\exists! \, \delta \in I_t \qquad \text{s.t.} \quad J_t(\delta) = 0.$$

4.1 Stopping Criterion

Here we discuss how to determine if the solution is already optimal and when to terminate the algorithm. While running the algorithm, $(\boldsymbol{W}_t, \boldsymbol{\xi}_t)$ may be infeasible to the primal problem. Denote the index set of the violated constraints by $\mathcal{I}_{\mathrm{vio}} := \{k \in \mathbb{N}_K \mid (\boldsymbol{W}_t, \boldsymbol{\xi}_t) \notin \mathcal{C}_k\}$ and let us define $\bar{\boldsymbol{\xi}}_t \in \mathbb{R}^K_{++}$ so that the k-th entry is given by $\bar{\xi}_{h,t} := \frac{1}{b_h} \langle \boldsymbol{W}_t, \boldsymbol{A}_h \rangle$ for $h \in \mathcal{I}_{\mathrm{vio}}$ and $\bar{\xi}_{h,t} := \xi_{h,t}$ for $h \notin \mathcal{I}_{\mathrm{vio}}$. Note that $(\boldsymbol{W}_t, \bar{\boldsymbol{\xi}}_t)$ is a feasible solution, and $\bar{\boldsymbol{\xi}}_t = \boldsymbol{\xi}_t$ when $(\boldsymbol{W}_t, \boldsymbol{\xi}_t)$ is feasible. The objective gap after iteration t is bounded as follows:

$$\mathrm{BD}_\varphi((\boldsymbol{W}_t, \bar{\boldsymbol{\xi}}_t), (\boldsymbol{I}, \boldsymbol{1})) - \mathrm{BD}_\star \leq \sum_{h \in \mathcal{I}_{\mathrm{vio}}} c_h \left(\varphi_{\mathrm{l}}(\bar{\xi}_{h,t}) - \varphi_{\mathrm{l}}(\xi_{h,t}) - \nabla\varphi_{\mathrm{l}}(1)(\bar{\xi}_{h,t} - \xi_{h,t}) \right) - \sum_{h=1}^{K} \alpha_h y_h \left(\langle \boldsymbol{A}_h, \boldsymbol{W}_t \rangle - b_h \xi_{h,t} \right),$$

where we have defined

$$\mathrm{BD}_\star := \min_{(\boldsymbol{W}, \boldsymbol{\xi}) \in \bigcap_h \mathcal{C}_h} \mathrm{BD}_\varphi((\boldsymbol{W}, \boldsymbol{\xi}), (\boldsymbol{I}, \boldsymbol{1})).$$

Then this upper-bound of the objective gap can be used for the stopping criterion of the Dykstra algorithm.

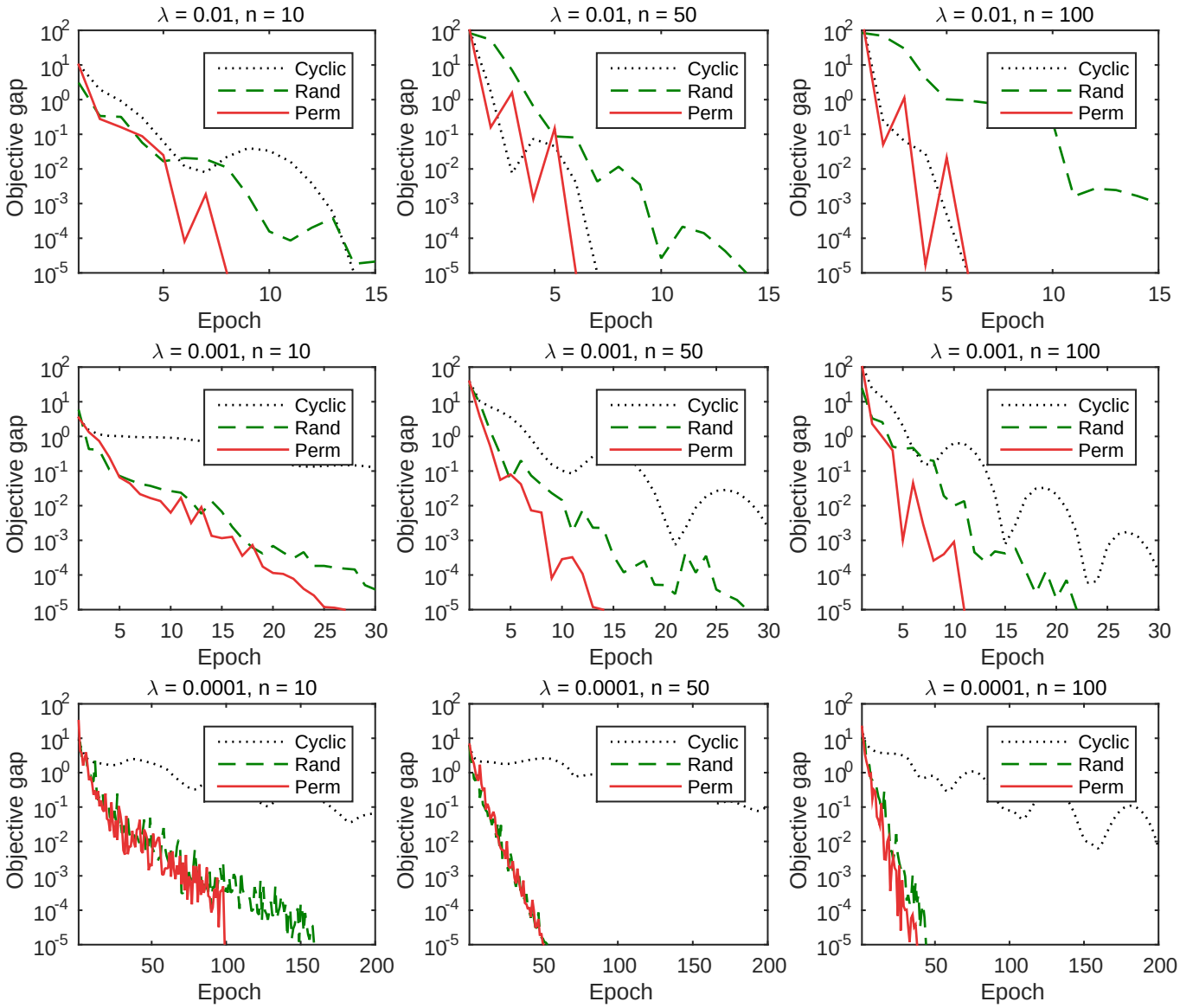

Figure 1: Convergence behavior of the algorithms using different settings.

5 Numerical Experiments

We conducted experiments to assess the convergence speed of our optimization algorithms and the generalization performance for pattern recognition.

5.1 Convergence behavior of optimization algorithms

We examined our algorithms for assessment of convergence speed. We generated datasets artificially as follows. $K = 50$ matrices $\boldsymbol{F}_k \in \mathbb{R}^{n \times n}$ are generated in which each entry is drawn from the uniform distribution in the interval $[-0.5, 0.5]$. Then, we set $\boldsymbol{A}_k := \boldsymbol{F}_k \boldsymbol{F}_k^\top$. The values of the variables y_k are randomly chosen from $\{\pm 1\}$ with same probabilities. We set $\boldsymbol{b} = \boldsymbol{1}$ and $\boldsymbol{c} = \boldsymbol{1}/(\lambda K)$. We exhaustively tested Cyclic, Perm, and Rand with the settings of $\lambda = 10^{-2}, 10^{-3}, 10^{-4}$ and $n = 10, 50, 100$.

Figure 1 demonstrates the convergence behavior of the cyclic Dykstra algorithm and the two stochastic Dykstra algorithm with various λ and n. Here, one epoch is called K times projection onto a single half-space. ISBD is employed as the objective function for learning the metric $\boldsymbol{W}$. The objective gap is defined as the difference between the current objective value and the minimum. In most of the settings, the two stochastic Dykstra algorithms converged faster than the cyclic algorithm. Especially when $\lambda = 10^{-4}$, the cyclic algorithm was too slow to use it in practice.

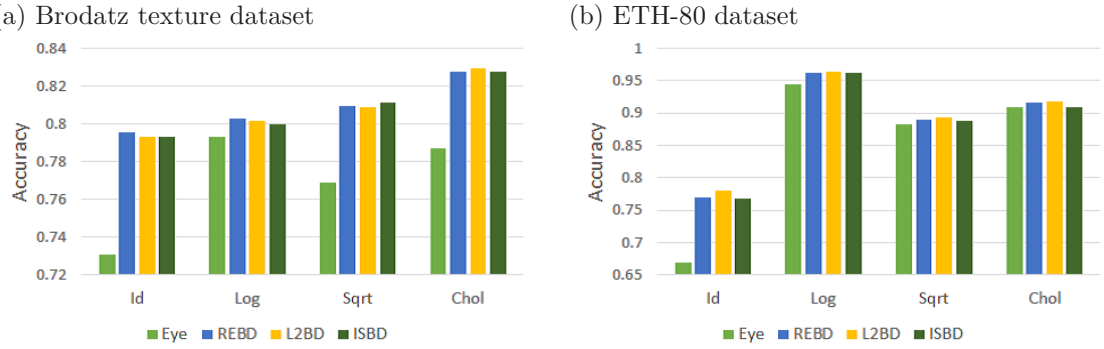

Figure 2: Generalized performances for pattern recognition.

5.2 Generalization performance for pattern recognition

We used the Brodatz texture dataset [21] containing 111 different texture images to examine the generalization performance for texture classification. Each image has a size of 640×640 and gray-scaled. Images were individually divided into four sub-images of equal size. One of the four sub-images was picked randomly and used for testing, and the rest of the images were used for training.

For each training image and each testing image, covariance descriptors of randomly chosen 50 were extracted from 128×128 patches. The covariance matrices are of five-dimensional feature vectors $[I, |I_x|, |I_y|, |I_{xx}|, |I_{yy}|]^\top$. Then, $11,100 (= 111 \times 2 \times 50)$ covariance descriptors are obtained for training and testing, respectively. For evaluation of generalized performance, k-nearest neighbor classifier is used, where the number of the nearest neighbors is set to three. We set $K = 100 \times n_c$, $b_{\mathrm{ub}} = 0.05$, and $b_{\mathrm{lb}} = 0.95$.

We also examined the generalization performance for generic visual categorization using the ETH-80 dataset [16] containing $n_c = 8$ classes. Each class has 10 objects, each of which includes 41 colored images. For every object, 20 images are randomly chosen and used for training, and the rest of images are used for testing.

One covariance matrix is obtained from each image. Eight features $[x, y, R, G, B, |I_x|, |I_y|, |I_{xx}|, |I_{yy}|]^\top$ are obtained from each pixel in an image.

We tried four types of $\boldsymbol{\Phi}$: **Id**: $\boldsymbol{\Phi}(\boldsymbol{X}) = \boldsymbol{X}$, **Log**: $\boldsymbol{\Phi}(\boldsymbol{X}) = \mathrm{logm}(\boldsymbol{X})$, **Sqrt**: $\boldsymbol{\Phi}(\boldsymbol{X}) = \boldsymbol{X}^{1/2}$, **Chol**: $\boldsymbol{\Phi}(\boldsymbol{X}) = \mathrm{chol}(\boldsymbol{X})$. The parameter $\boldsymbol{W}$ is determined by the metric learning algorithms with ISBD, L2BD, and REBD, to be compared with $\boldsymbol{W} = \boldsymbol{I}$ we denote as **Eye**. Note that $D_{\boldsymbol{\Phi}}(\cdot, \cdot; \boldsymbol{I}) = D_{\boldsymbol{\Phi}}(\cdot, \cdot)$. Figure 2 gives the accuracy bar plots for the two multi-class classification problems. Whichever $\boldsymbol{\Phi}$ is used, supervised metric learning improved the generalization performances both for texture classification and for generic visual categorization. For texture classification, the Cholesky decomposition-based mapping $\mathrm{chol}(\cdot)$ achieved the best accuracy, while the matrix logarithm-based mapping $\mathrm{logm}(\cdot)$ obtained the highest accuracy for generic image categorization.

6 Conclusions

In this paper, we have devised several objective functions for metric learning on positive semidefinite cone, all of which can be minimized by the Dykstra algorithm. We have introduced a new technique that performs each update efficiently when the Dykstra algorithm is applied to the metric learning problems. We have empirically demonstrated that the stochastic versions of the Dykstra algorithm are much faster than the original algorithm.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 26249075, 40401236. The last author would like to thank Dr. Zhiwu Huang for fruitful discussions.

References

[1] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic resonance in medicine*, 56(2):411–421, 2006.

[2] Rajendra Bhatia. *Positive Definite Matrices*. Princeton University Press, 2009.

[3] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010)*, pages 177–187, Paris, France, August 2010. Springer.

[4] Yair Censor and Simeon Reich. The dykstra algorithm with bregman projections. *Communications in Applied Analysis*, 2:407–419, 1998.

[5] Nello Cristianini, John Shawe-Taylor, André Elisseeff, and Jaz S. Kandola. On kernel-target alignment. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *NIPS*, pages 367–373. MIT Press, 2001.

[6] 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.

[7] Aaron J Defazio, Tibério S Caetano, and Justin Domke. Finito: A faster, permutable incremental gradient method for big data problems. *arXiv preprint arXiv:1407.2710*, 2014.

[8] Ian L Dryden, Alexey Koloydenko, and Diwei Zhou. Non-euclidean statistics for covariance matrices, with applications to diffusion tensor imaging. *The Annals of Applied Statistics*, pages 1102–1123, 2009.

[9] Richard L. Dykstra. An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, 78(384):837–842, December 1983.

[10] Zhiwu Huang, Ruiping Wang, Shiguang Shan, Xianqiu Li, and Xilin Chen. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, pages 720–729, 2015.

[11] Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong Li, and Mehrtash Tafazzoli Harandi. Kernel methods on the riemannian manifold of symmetric positive definite matrices. In *CVPR*, pages 73–80. IEEE, 2013.

[12] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26: Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 315–323, 2013.

[13] Tsuyoshi Kato and Nozomi Nagano. Metric learning for enzyme active-site search. *Bioinformatics*, 26(21):2698–2704, November 2010.

[14] Tsuyoshi Kato, Wataru Takei, and Shinichiro Omachi. A discriminative metric learning algorithm for face recognition. *IPSJ Transactions on Computer Vision and Applications*, 5:85–89, 2013.

[15] Tsuyoshi Kato, Koji Tsuda, and Kiyoshi Asai. Selective integration of multiple biological data for supervised network inference. *Bioinformatics*, 21:2488–2495, May 2005.

[16] Bastian Leibe and Bernt Schiele. Analyzing appearance and contour based methods for object categorization. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–409. IEEE, 2003.

[17] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[18] Tomoki Matsuzawa, Raissa Relator, Wataru Takei, Shinichiro Omachi, and Tsuyoshi Kato. Mahalanobis encodings for visual categorization. *IPSJ Transactions on Computer Vision and Applications*, 7:69–73, July 2015. doi: 10.2197/ipsjtcva.7.1.

[19] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.

[20] Florent Perronnin, Jorge Sanchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Computer Vision–ECCV 2010*, pages 143–156. Springer, 2010.

[21] Trygve Randen and John Hakon Husoy. Filtering for texture classification: A comparative study. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(4):291–310, 1999.

[22] Raissa Relator, Yoshihiro Hirohashi, Eisuke Ito, and Tsuyoshi Kato. Mean polynomial kernel and its application to vector sequence recognition. *IEICE Transactions on Information and Systems*, E97-D(7):1855–1863, July 2014.

[23] Raissa Relator, Nozomi Nagano, and Tsuyoshi Kato. Using bregmann divergence regularized machine for comparison of molecular local structures. *IEICE Transactions on Information & Systems*, E99-D(1):–, Jan 2016.

[24] Nicolas L. Roux, Mark Schmidt, and Francis R. Bach. A stochastic gradient method with an exponential convergence _rate for finite training sets. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2663–2671. Curran Associates, Inc., 2012.

[25] Gregory Shakhnarovich, John W Fisher, and Trevor Darrell. Face recognition from long-term observations. In *ECCV 2002*, pages 851–865. Springer Berlin Heidelberg, 2002.

[26] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: primal estimated sub-gradient solver for SVM. *Math. Program.*, 127(1):3–30, 2011.

[27] Suvrit Sra. A new metric on the manifold of kernel matrices with application to matrix geometric means. In *Advances in Neural Information Processing Systems*, pages 144–152, 2012.

[28] Raviteja Vemulapalli and David W Jacobs. Riemannian metric learning for symmetric positive definite matrices. *arXiv preprint arXiv:1501.02393*, 2015.

[29] Zhizhou Wang and Baba C Vemuri. An affine invariant tensor dissimilarity measure and its applications to tensor-valued image segmentation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–228. IEEE, 2004.

[30] Florian Yger and Masashi Sugiyama. Supervised logeuclidean metric learning for symmetric positive definite matrices. *arXiv preprint arXiv:1502.03505*, 2015.