Mach Learn (2015) 99:169–187
DOI 10.1007/s10994-014-5446-z

Adaptive Euclidean maps for histograms: generalized Aitchison embeddings

Tam Le · Marco Cuturi

Received: 28 January 2014 / Accepted: 6 May 2014 / Published online: 13 August 2014

Abstract Learning distances that are specifically designed to compare histograms in the probability simplex has recently attracted the attention of the machine learning community. Learning such distances is important because most machine learning problems involve bags of features rather than simple vectors. Ample empirical evidence suggests that the Euclidean distance in general and Mahalanobis metric learning in particular may not be suitable to quantify distances between points in the simplex. We propose in this paper a new contribution to address this problem by generalizing a family of embeddings proposed by Aitchison (J R Stat Soc 44:139–177, 1982) to map the probability simplex onto a suitable Euclidean space. We provide algorithms to estimate the parameters of such maps by building on previous work on metric learning approaches. The criterion we study is not convex, and we consider alternating optimization schemes as well as accelerated gradient descent approaches. These algorithms lead to representations that outperform alternative approaches to compare histograms in a variety of contexts.

Keywords Metric learning for histograms · Aitchison geometry · Probability simplex · Embeddings

Editors: Cheng Soon Ong, Tu Bao Ho, Wray Buntine, Bob Williamson, and Masashi Sugiyama.

T. Le (✉) · M. Cuturi
Graduate School of Informatics, Kyoto University, Kyoto, Japan
e-mail: tam.le@iip.ist.i.kyoto-u.ac.jp
URL: http://www.iip.ist.i.kyoto-u.ac.jp/member/tamle/

M. Cuturi
e-mail: mcuturi@i.kyoto-u.ac.jp
URL: http://www.i.kyoto-u.ac.jp/∼mcuturi

1 Introduction

Defining a distance to compare objects of interest is an important problem in machine learning. Many metric learning algorithms were proposed to tackle this problem by considering labeled datasets, most of which exploit the simple and intuitive framework of Mahalanobis distances (Xing et al. 2002; Schultz and Joachims 2003; Kwok 2003; Goldberger et al. 2004; Shalev-Shwartz et al. 2004; Globerson and Roweis 2005). Within these contributions, two algorithms are particularly popular in applications: the Large Margin Nearest Neighbor (LMNN) approach described by Weinberger et al. (2006), Weinberger and Saul (2008, 2009), and the Information-Theoretic Metric Learning (ITML) approach proposed by Davis et al. (2007).

Among such objects of interest, histograms—the normalized representation for bags of features—play a fundamental role in many applications, from computer vision (Julesz 1981; Sivic and Zisserman 2003; Perronnin et al. 2010; Vedaldi and Zisserman 2012), natural language processing (Salton and McGill 1983; Salton 1989; Baeza-Yates and Ribeiro-Neto 1999; Joachims 2002; Blei et al. 2003; Blei and Lafferty 2006, 2009), speech processing (Doddington 2001; Campbell et al. 2003; Campbell and Richardson 2007) to bioinformatics (Erhan et al. 1980; Burge et al. 1992; Leslie et al. 2002). Mahalanobis distances can be used as such on histograms or bags-of-features, but fail however to incorporate the geometrical constraints of the probability simplex (non-negativity, normalization) in their definition. Given this issue, Cuturi and Avis (2011) and Kedem et al. (2012) have very recently proposed to learn the parameters of distances specifically designed for histograms, namely the transportation distance and a generalized variant of the χ^2 distance respectively.

We propose in this work a new approach to compare histograms that builds upon older work by Aitchison (1982). In a series of influential papers and monographs, Aitchison and Shen (1980), Aitchison and Lauder (1985), Aitchison (1982, 1986, 2003) proposed to study different maps from the probability simplex onto a Euclidean space of suitable dimension. These maps are constructed such that they preserve the geometric characteristics of the probability simplex, yet make subsequent analysis easier by relying only upon Euclidean tools, such as Euclidean distances, quadratic forms and ellipses. Our goal in this paper is to follow this line of work and propose suitable maps from the probability simplex to a Euclidean space of suitable dimension. However, rather than relying on a few mappings defined a priori such as those proposed in Aitchison (1982), we propose to *learn* such maps directly in a supervised fashion using Mahalanobis metric learning.

We build upon our earlier contribution (Le and Cuturi 2013) and provide new insights on the empirical behaviour of our method, notably in terms of convergence speed and parameter sensitivity. We also consider the adaptive restart heuristic (O'Donoghue and Candès 2013) and show that it can prove beneficial. Source code for our tools can be obtained in http://github.com/lttam/GenAitEmb.

This paper is organized as follows: after providing some background on Aitchison embeddings in Sect. 2, we propose a generalization of Aitchison embeddings in Sect. 3. In Sect. 4, we propose algorithms to learn the parameters of such embeddings using training data. We also review related work in Sect. 5, before providing experimental evidence in Sect. 6 that our approach improves upon other adaptive metrics on the probability simplex. Finally, we provide some observations on the empirical behavior of our algorithms in Sect. 7 before concluding this paper in Sect. 8.

2 Aitchison embeddings

We consider the probability simplex of d coordinates, $\mathbb{S}_d \stackrel{\text{def}}{=} \left\{ \mathbf{x} \in \mathbb{R}^d \;\middle|\; \sum_{i=1}^{d} \mathbf{x}_i = 1 \text{ and } \mathbf{x}_i \geq 0, \right.$ $\left. 1 \leq i \leq d \right\}$, throughout this paper. Aitchison (1982, 1986, 2003) claims that the information reflected in histograms lies in *the relative values of their coordinates* rather than on their absolute value. Therefore, Aitchison makes the point that comparing histograms directly with Euclidean distances is not appropriate, since the Euclidean distance can only measure the arithmetic difference between coordinates. Given two points $\mathbf{x}$ and $\mathbf{z}$ in the simplex, Aitchison proposes to focus explicitly on the log-ratio of $\mathbf{x}_i$ and $\mathbf{z}_i$ for each coordinate i, which can be expressed as the arithmetic difference of the logarithms of $\mathbf{x}_i$ and $\mathbf{z}_i$,

$$\log \frac{\mathbf{x}_i}{\mathbf{z}_i} = \log \mathbf{x}_i - \log \mathbf{z}_i .$$

2.1 Additive log-ratio embedding

The first embedding proposed by Aitchison (1982, p. 144, 2003, p. 29) is the additive log-ratio map (**alr**) which maps a vector $\mathbf{x}$ from the probability simplex $\mathbb{S}_d$ onto $\mathbb{R}^{d-1}$,

$$\mathbf{alr}\,(\mathbf{x}) \stackrel{\text{def}}{=} \left[\log \frac{\mathbf{x}_i + \varepsilon}{\mathbf{x}_d + \varepsilon} \right]_{1 \leq i \leq d-1} \in \mathbb{R}^{d-1},$$

where $\varepsilon > 0$ is small. The **alr** map for $\mathbf{x} \in \mathbb{S}_d$ can be reformulated as:

$$\mathbf{alr}\,(\mathbf{x}) = \mathbf{U} \log\left(\mathbf{x} + \varepsilon \mathbf{1}_d\right), \mathbf{U} = \begin{bmatrix} 1 & \cdots & 0 & -1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & -1 \end{bmatrix}, \tag{1}$$

where $\mathbf{U} \in \mathbb{R}^{(d-1)\times d}$, $\mathbf{1}_d \in \mathbb{R}^d$ is the vector of ones, and $\log \mathbf{x}$ is the element-wise logarithm of $\mathbf{x}$. The formula of the **alr** map is related to the definition of the logistic-normal distribution (Aitchison and Shen 1980; Blei and Lafferty 2006) on $\mathbb{S}_d$. The density of a logistic normal distribution at any point in the simplex is proportional to the density of the multivariate normal density on the image of that point under the **alr** map. The **alr** map is an isomorphism between $(\mathbb{S}_d, \oplus, \otimes)$ and $\left(\mathbb{R}^{d-1}, +, \times\right)$ where $\oplus$ and $\otimes$ are the perturbation (Aitchison 2003, p. 24) and power (Aitchison 2003, p. 26) operations in the probability simplex respectively, but not isometric since it does not preserve the distance between them.

2.2 Centered log-ratio embedding

The second embedding proposed by Aitchison (2003, p. 30) is the centered log-ratio embedding (**clr**), which considers the log-ratio of each coordinate of $\mathbf{x}$ with the geometric mean of all its coordinates,

$$\mathbf{clr}\,(\mathbf{x}) \stackrel{\text{def}}{=} \left[\log \frac{\mathbf{x}_i + \varepsilon}{\sqrt[d]{\prod_{j=1}^{d} (\mathbf{x}_j + \varepsilon)}} \right]_{1 \leq i \leq d} \in \mathbb{R}^d. \tag{2}$$

The **clr** map can be also expressed with simpler notations in matrix form:

$$\mathbf{clr}\,(\mathbf{x}) = \left(\mathbf{I} - \frac{\mathbf{1}_{d\times d}}{d} \right) \log\left(\mathbf{x} + \varepsilon \mathbf{1}_d\right).$$

Here, $\mathbf{I}$ and $\mathbf{1}_{d\times d}$ stand for the identity matrix and the matrix of ones in $\mathbb{R}^{d\times d}$ respectively. The **clr** map is not only an isomorphism, but also an isometry between the probability simplex $\mathbb{S}_d$ and $\mathbb{R}^d$. Note that the **clr** map spans the orthogonal of $\mathbf{1}_d$ in $\mathbb{R}^d$.

2.3 Isometric log-ratio embedding

Egozcue et al. (2003) proposed to project the images of the **clr** map onto $\mathbb{R}^{d-1}$, to define the *isometric log-ratio embedding* (**ilr**). The **ilr** map is defined as follows:

$$\mathbf{ilr}\,(\mathbf{x}) \stackrel{\text{def}}{=} \mathbf{V}\mathbf{clr}\,(\mathbf{x}) = \mathbf{V}\left(\mathbf{I} - \frac{\mathbf{1}_{d\times d}}{d}\right)\log\left(\mathbf{x} + \varepsilon\mathbf{1}_d\right), \tag{3}$$

where $\mathbf{V} \in \mathbb{R}^{(d-1)\times d}$ is a matrix whose row vectors describe a base of the null space of $\mathbf{1}_d^T$ in $\mathbb{R}^d$. The **ilr** map is also an isometric map between both spaces in Aitchison's sense.

Aitchison's original definitions do not consider explicitly the regularization coefficient ε (1982, 1986, 2003). In that literature, the histograms are either assumed to have strictly positive values or the problem is dismissed by stating that all values can be regularized by a very small constant (Aitchison and Lauder 1985, p. 132; 1986, §11.5). We consider explicitly this constant ε here because it forms the basis of the embeddings we propose in the next section.

3 Generalized Aitchison embeddings

Rather than settling for a particular weight matrix—such as those defined in Eqs. (1), (2) or (3)—and defining a regularization constant ε arbitrarily, we introduce in the definition below a family of mappings that leverage instead these parameters to define a flexible generalization of Aitchison's maps. In the following, $\mathcal{S}_d^+$ is the cone of symmetric positive semidefinite matrices of size $d \times d$.

Definition 1 Let $\mathbf{P}$ be a matrix in $\mathbb{R}^{m\times d}$ and $\mathbf{b}$ be a vector in the positive orthant $\mathbb{R}^d_+$. We define the generalized Aitchison embedding $\mathfrak{a}(\mathbf{x})$ of a point $\mathbf{x}$ in $\mathbb{S}_d$ parameterized by $\mathbf{P}$ and $\mathbf{b}$ as

$$\mathfrak{a}(\mathbf{x}) \stackrel{\text{def}}{=} \mathbf{P}\log\left(\mathbf{x} + \mathbf{b}\right) \in \mathbb{R}^m. \tag{4}$$

Vector $\mathbf{b}$ in Eq. (4), can be interpreted as a pseudo-count vector that weights the importance of each coordinate of $\mathbf{x}$. Figure 1 illustrates how larger pseudo-count values tend to smoothen the logarithm mapping. A large value for $\mathbf{b}_i$ directly implies that the map for the coordinate described in bin number i is nearly constant, thereby canceling the impact of that coordinate in subsequent analysis. Smaller values for $\mathbf{b}_i$ denote on the contrary influential coordinates.

We propose to *learn* $\mathbf{P}$ and $\mathbf{b}$ such that histograms mapped following $\mathfrak{a}$ can be efficiently discriminated using the Euclidean distance. The Euclidean distance between the images of two histograms $\mathbf{x}$ and $\mathbf{z}$ under the embedding $\mathfrak{a}$ is:

$$\begin{aligned} d_{\mathfrak{a}}(\mathbf{x}, \mathbf{z}) &\stackrel{\text{def}}{=} d\left(\mathfrak{a}\left(\mathbf{x}\right), \mathfrak{a}\left(\mathbf{z}\right)\right) \\ &= \left\|\mathbf{P}\log\left(\mathbf{x} + \mathbf{b}\right) - \mathbf{P}\log\left(\mathbf{z} + \mathbf{b}\right)\right\|_2 \\ &= \left\|\log\left(\frac{\mathbf{x} + \mathbf{b}}{\mathbf{z} + \mathbf{b}}\right)\right\|_{\mathbf{Q}}, \end{aligned} \tag{5}$$

where the division between two vectors is here considered element-wise and we have introduced the positive semidefinite matrix $\mathbf{Q} = \mathbf{P}^T\mathbf{P}$, along with the Mahalanobis norm

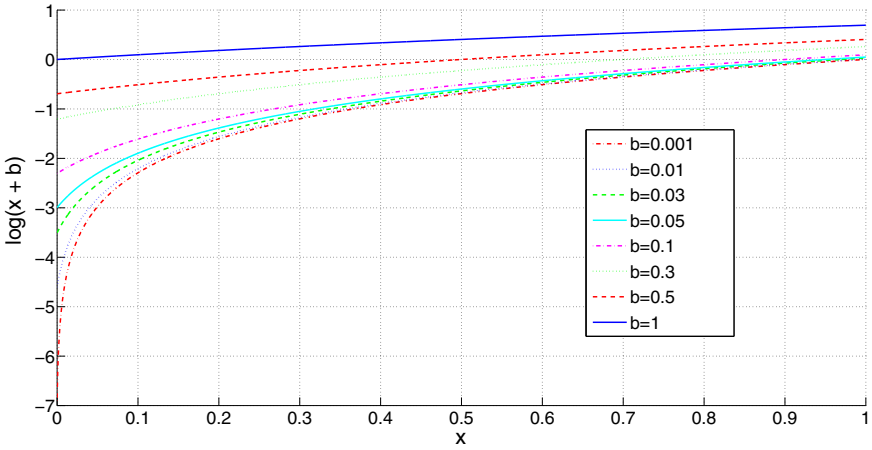

Fig. 1 Impact of variable pseudo-count values in the logarithm function

$\|\cdot\|_{\mathbf{Q}} \stackrel{\text{def}}{=} \sqrt{\cdot^T Q \cdot}$. Our goal is to learn both $\mathbf{Q} \in \mathcal{S}_d^+$ (we may also consider $\mathbf{P}$ directly) and the pseudo-count vector $\mathbf{b}$ to obtain an embedding that performs well with k-nearest neighbors.

4 Learning generalized Aitchison embeddings

4.1 Criterion

Let $D = \{(\mathbf{x_i}, y_i)_{1 \leq i \leq N}\}$ be a dataset of labeled points in the simplex, where each $\mathbf{x_i} \in \mathbb{S}_d$ and each $y_i \in \{1, \cdots, L\}$ is a label. We follow Weinberger's approach to define a criterion to optimize the parameters $(\mathbf{Q}, \mathbf{b})$ (2006, 2009). Weinberger et al. propose a large margin approach to nearest neighbor classification: given a training set D, their criterion considers for a single reference point $\mathbf{x_i}$ the cumulated distance of its closest neighbors that belong to the same class, corrected by a coefficient which takes into account whether points from a different class are in the immediate neighborhood of $\mathbf{x_i}$. Taken together over the entire dataset, these two factors promote metric parameters which ensure that each point's immediate neighborhood is mostly composed of points that share its label.

These ideas can be formulated using the following notations. Let κ be an integer. Given a pair of parameters $(\mathbf{Q}, \mathbf{b})$, consider the geometry induced by $d_{\mathfrak{a}}$. For each point $\mathbf{x_i}$ in the dataset, there exists κ neighbors of $\mathbf{x_i}$ which share its label. We single out these indices by introducing the binary relationship $j \rightsquigarrow i$ for two indices $1 \leq i \neq j \leq N$. The notation $j \rightsquigarrow i$ means that the j-th point is among those close neighbors with the same class (namely $y_i = y_j$). The set of indices j such that $j \rightsquigarrow i$ is called the set of *target neighbors* of the i-th point. Note that $j \rightsquigarrow i$ does not imply $i \rightsquigarrow j$.

Next, we introduce the hinge loss of a real number t as $[t]_+ \stackrel{\text{def}}{=} \max(t, 0)$, to define the margin between three points: given a triplet (i, j, ℓ) of distinct indices, the margin $\mathcal{H}_{ij\ell}$ is derived as:

$$\mathcal{H}_{ij\ell} \stackrel{\text{def}}{=} \left[1 + d_{\mathfrak{a}}^2\left(\mathbf{x_i}, \mathbf{x_j}\right) - d_{\mathfrak{a}}^2\left(\mathbf{x_i}, \mathbf{x_\ell}\right)\right]_+ .$$

This margin is positive whenever the distance between the i-th and ℓ-th points is not larger than the distance between the i-th and j-th points plus an offset of 1. If, for instance, the i-th

Algorithm 1 Alternating optimization (AO) Approach for Problem (6)

Input: data $(\mathbf{x_i}, y_i)_{1\leq i\leq N}$, neighborhood size κ, intialization $\mathbf{b}_0$, $\mathbf{Q}_0$.
Set $t \leftarrow 0$.
repeat
Find κ target neighbors for each point $\mathbf{x_i}$ with $d_\mathfrak{a}$ as in Equation (5) at $(\mathbf{Q}_t, \mathbf{b}_t)$.
Compute $\mathbf{Q}_{t+1}$ using the LMNN algorithm initialized with $\mathbf{Q}_t$ and training data $\{(\log(\mathbf{x_i} + \mathbf{b}_t), y_i)_{1\leq i\leq N}\}$.
Update target neighbors for each vector $\mathbf{x_i}$ using parameters $(\mathbf{Q}_{t+1}, \mathbf{b}_t)$.
Compute $\mathbf{b}_{t+1}$ using Algorithm 2 initialized with $\mathbf{b}_t$, on $\{(\mathbf{x_i}, y_i)_{1\leq i\leq N}\}$ and $\mathbf{Q}_{t+1}$.
Update the objective $\mathcal{F}_{t+1} \leftarrow \mathcal{F}(\mathbf{Q}_{t+1}, \mathbf{b}_{t+1})$.
$t \leftarrow t + 1$.
until $t < t_{\max}$ or insufficient progress for $\mathcal{F}_t$.
Output: matrix $\mathbf{Q}_t$, pseudo-count vector $\mathbf{b}_t$.

and j-th points share the same class but the ℓ-th point comes from a different class, $\mathcal{H}_{ij\ell}$ will be positive whenever the ℓ-th point is not far enough from the i-th point relative to where the j-th point stands.

Using these definitions, we can define the following metric learning problem:

$$\begin{aligned} \min_{\mathbf{Q},\mathbf{b}} \quad & \mathcal{F} \stackrel{\text{def}}{=} \sum_{i,j\rightsquigarrow i} d^2_\mathfrak{a}\left(\mathbf{x_i}, \mathbf{x_j}\right) + \mu \sum_{i,j\rightsquigarrow i} \sum_{\ell} (1 - y_{i\ell})\mathcal{H}_{ij\ell} + \lambda\|\mathbf{b}\|_2^2 \\ \text{s.t.} \quad & \mathbf{Q} \succeq 0 \\ & \mathbf{b} > \mathbf{0}_d, \end{aligned} \tag{6}$$

where $y_{i\ell}$ is equal to 1 if $y_i = y_\ell$ and 0 otherwise, and $\mu > 0, \lambda > 0$ are two regularization parameters. The first term in the objective favors small distances between neighboring points of the same class, while the second term ensures no points with a different label are in the neighborhood of each point, complemented by a regularization term.

4.2 Alternating optimization

Unlike the original LMNN formulation, optimization problem (6) is not convex because of the introduction of a pseudo count vector $\mathbf{b}$. Although the objective is still convex with respect to $\mathbf{Q}$, it is non-convex with respect to $\mathbf{b}$. We consider first a naive approach which updates alternatively $\mathbf{Q}$ and $\mathbf{b}$. This approach is summarized in Algorithm 1 and detailed below.

When $\mathbf{b}$ is fixed, optimization problem (6) is equivalent to the Mahalanobis metric learning problem: indeed, once each training vector $\mathbf{x}$ is mapped to $\log(\mathbf{x} + \mathbf{b})$, problem (6) can be solved with a LMNN solver.

When $\mathbf{Q}$ is fixed, we can use a projected subgradient descent to learn the pseudo-count vector $\mathbf{b}$. Defining $g_{ij}(\mathbf{b}) \stackrel{\text{def}}{=} d^2_\mathfrak{a}\left(\mathbf{x_i}, \mathbf{x_j}\right)$, we can compute the gradient of g_{ij} as:

$$\nabla g_{ij}(\mathbf{b}) = 2\left(\mathbf{Q}\log\frac{\mathbf{x_i} + \mathbf{b}}{\mathbf{x_j} + \mathbf{b}}\right) \circ \left(\frac{1}{\mathbf{x_i} + \mathbf{b}} - \frac{1}{\mathbf{x_j} + \mathbf{b}}\right),$$

where $\circ$ is the Schur product between vectors or matrices. Since only terms such that $\mathcal{H}_{ij\ell}$ is positive contribute to the gradient, a subgradient γ for the objective function $\mathcal{F}$ at $\mathbf{b}_t$ can be expressed as

$$\gamma = \sum_{i,j\rightsquigarrow i}\left[\nabla g_{ij}(\mathbf{b}_t) + \mu \sum_{\ell|\mathcal{H}_{ij\ell}>0}\left[\nabla g_{ij}(\mathbf{b}_t) - \nabla g_{i\ell}(\mathbf{b}_t)\right]\right] + 2\lambda\mathbf{b}_t. \tag{7}$$

Algorithm 2 Subgradient Descent Update of $\mathbf{b}$ when $\mathbf{Q}$ is fixed.

Input: data $(\mathbf{x_i}, y_i)_{1\leq i\leq N}$, a matrix $\mathbf{Q}$, a subgradient step size t_0, an initial vector $\mathbf{b}_0$.
Set $t \leftarrow 0$.
Set $\mathbf{b}_t \leftarrow \mathbf{b}_0$.
repeat
  Compute a subgradient γ at $\mathbf{b}_t$ following Equation (7).
  Compute $\mathbf{b}_{t+1} \leftarrow \Pi\left(\mathbf{b}_t - \frac{t_0}{\sqrt{t}}\gamma\right)$.
  Update the objective $\mathcal{F}_{t+1} \leftarrow \mathcal{F}(\mathbf{Q}, \mathbf{b}_{t+1})$.
  Set $t \leftarrow t+1$.
until $t < t_{\max}$ or insufficient progress for $\mathcal{F}_t$.
Output: a pseudo-count vector $\mathbf{b}_t$.

This formula results in the following update for $\mathbf{b}_t$ using a preset step size $\frac{t_0}{\sqrt{t}}$:

$$\mathbf{b}_{t+1} = \Pi\left(\mathbf{b}_t - \frac{t_0}{\sqrt{t}}\gamma\right),$$

where $\Pi(\mathbf{x})$ is the projection of $\mathbf{x}$ on the positive orthant offset by a small minimum threshold $\varepsilon = 10^{-20}$, namely the set of all vectors whose coordinates are larger or equal to 10^{-20}. A pseudo-code of this approach is summarized in Algorithm 2. We can set the initial point $\mathbf{Q}_0$ to be equal to $\mathbf{P}^T\mathbf{P}$ where $\mathbf{P}$ can be selected among the linear embeddings originally considered by Aitchison presented in Sect. 2. We initialize the pseudo-count vector to the uniform smoothing term $\mathbf{1}_d/d$.

4.3 Projected subgradient descent with Nesterov acceleration

We propose in this section a more straightforward approach to the problem of minimizing Problem (6) which bypasses the cost associated with running many iterations of the LMNN solver. We consider a projected subgradient descent using Nesterov acceleration scheme (Nesterov 1983, 2004) to optimize the parameters $(\mathbf{Q}, \mathbf{b})$ in Problem (6) directly. Our experiments show that this approach is considerably faster and equally efficient in terms of classification accuracy.

Analogously to the previous section, we consider the distance $d^2_{\mathfrak{a}}\left(\mathbf{x_i}, \mathbf{x_j}\right)$ as a function $h_{ij}(\mathbf{Q}, \mathbf{b})$ of $\mathbf{Q}$ and $\mathbf{b}$. Gradients of h_{ij} with respect to $\mathbf{Q}$ and $\mathbf{b}$ are, introducing the notation $\mathbf{u}_{ij}$ below:

$$\nabla h_{ij}(\mathbf{Q}, \mathbf{b})|_{\mathbf{Q}} = \left(\log\frac{\mathbf{x_i}+\mathbf{b}}{\mathbf{x_j}+\mathbf{b}}\right)\left(\log\frac{\mathbf{x_i}+\mathbf{b}}{\mathbf{x_j}+\mathbf{b}}\right)^T = \mathbf{u}_{ij}\mathbf{u}_{ij}^T,$$
$$\nabla h_{ij}(\mathbf{Q}, \mathbf{b})|_{\mathbf{b}} = \nabla g_{ij}(\mathbf{b}).$$

At iteration $t+1$, a subgradient of the objective $\mathcal{F}$ with respect to $\mathbf{b}$ was given in Eq. (7). We derive similarly a subgradient Γ with respect to $\mathbf{Q}$:

$$\Gamma = \sum_{i,\, j \rightsquigarrow i}\left[\mathbf{u}_{ij}\mathbf{u}_{ij}^T + \mu\sum_{\ell|\mathcal{H}_{ij\ell}>0}\left[\mathbf{u}_{ij}\mathbf{u}_{ij}^T - \mathbf{u}_{i\ell}\mathbf{u}_{i\ell}^T\right]\right].$$

Nesterov acceleration scheme builds gradient updates using a momentum that involves two iterations. $\mathbf{b}_t$ and $\mathbf{Q}_t$ can be updated analogously as follows:

$$\begin{aligned}
\mathbf{b}_{t-1}^{nes} &= \mathbf{b}_{t-1} + \frac{t-2}{t+1}\left(\mathbf{b}_{t-1} - \mathbf{b}_{t-2}\right), \\
\mathbf{b}_t &= \Pi\left(\mathbf{b}_{t-1}^{nes} - \frac{t_0}{\sqrt{t}}\frac{\partial \mathcal{F}}{\partial \mathbf{b}}\left(\mathbf{Q}_{t-1}, \mathbf{b}_{t-1}^{nes}\right)\right). \\
\mathbf{Q}_{t-1}^{nes} &= \mathbf{Q}_{t-1} + \frac{t-2}{t+1}\left(\mathbf{Q}_{t-1} - \mathbf{Q}_{t-2}\right), \\
\mathbf{Q}_t &= \pi_{\mathcal{S}_d^+}\left(\mathbf{Q}_{t-1}^{nes} - \frac{t_0}{\sqrt{t}}\frac{\partial \mathcal{F}}{\partial \mathbf{Q}}\left(\mathbf{Q}_{t-1}^{nes}, \mathbf{b}_{t-1}\right)\right).
\end{aligned}$$

The projection $\pi_{\mathcal{S}_d^+}$ of a matrix onto the cone of positive semidefinite matrices is carried out by thresholding its negative eigenvalues.

4.4 Low-rank approaches

Torresani and Lee (2006) have proposed to learn low-rank embeddings for LMNN. We include this variation here, which is beneficial in terms of computational speed, since it only involves storing a low-rank Cholesky factor $\mathbf{P} \in \mathbb{R}^{m \times d}$ of $\mathbf{Q}$ where $m < d$ is a predetermined parameter. This gain comes at the cost of losing convexity when the problem is parameterized by $\mathbf{Q}$. The subgradient of $\mathcal{F}$ with respect to $\mathbf{P}$ is:

$$\frac{\partial \mathcal{F}}{\partial \mathbf{P}} = 2\mathbf{P}\frac{\partial \mathcal{F}}{\partial \mathbf{Q}} \in \mathbb{R}^{m \times d}.$$

When using a descent expressed in terms of $\mathbf{P}$, we obtain the updates

$$\begin{aligned}
\mathbf{P}_{t-1}^{nes} &= \mathbf{P}_{t-1} + \frac{t-2}{t+1}\left(\mathbf{P}_{t-1} - \mathbf{P}_{t-2}\right), \\
\mathbf{P}_t &= \mathbf{P}_{t-1}^{nes} - \frac{t_0}{\sqrt{t}}\frac{\partial \mathcal{F}}{\partial \mathbf{P}}\left(\mathbf{P}_{t-1}^{nes}\right).
\end{aligned}$$

Since no constraints hold on $\mathbf{P}$, we do not need a projection step.

4.5 Adaptive restart

The projected subgradient descent with Nesterov acceleration presented in Sect. 4.3 does not guarantee a monotone decrease of the objective value. Indeed, it has been observed that Nesterov acceleration scheme may create ripples in the objective value curve when plotted against iteration count. This phenomenon happens when the momentum built from Nesterov acceleration scheme becomes higher than a critical value (the optimal momentum value described by Nesterov (1983, 2004)), and thus damage convergence speed. To overcome this, we adopt the heuristic of O'Donoghue and Candès (2013), which sets the momentum back to zero whenever an increase in the objective is detected. Whenever $\mathcal{F}_t > \mathcal{F}_{t-1}$ at some point in time t, the idea of this heuristic is to erase the memory of previous iterations, reset the algorithm counter to 0 and use the current iteration as a warm start.

Table 1 Properties of datasets and their corresponding experimental parameters

Dataset	#Train	#Test	#Class	Feature	Rep	#Dim	#Run
MIT Scene	800	800	8	SIFT	BoF	800	5
UIUC Scene	1,500	1,500	15	SIFT	BoF	800	5
DSLR	409	89	31	SURF	BoF	800	5
WEBCAM	646	149	31	SURF	BoF	800	5
AMAZON	2,262	551	31	SURF	BoF	800	5
OXFORD Flower	680	680	17	SIFT	BoF	400	5
CALTECH-101	3,060	2,995	102	SIFT	BoF	400	3
Pascal Voc 2007	5,011	4,952	20	Dense Hue	BoF	100	1
MirFlickr	12,500	12,500	38	Dense Hue	BoF	100	1
MNIST	5,000	5,000	10	Normalized intensity		784	5
20 News Group	600	19,397	20	BoW	LDA	200	5
Reuters	500	9,926	10	BoW	LDA	200	5

5 Related work

Notwithstanding Aitchison's work, the logarithm mapping has been consistently applied in information retrieval to correct for the *burstiness* of feature counts (Salton 1989; Baeza-Yates and Ribeiro-Neto 1999; Rennie et al. 2003; Lewis et al. 2004; Madsen et al. 2005), using the mapping

$$\mathbf{x} \mapsto \log(\mathbf{x} + \alpha \mathbf{1}_d), \tag{8}$$

for an unnormalized histogram of feature counts $\mathbf{x}$, where $\alpha > 0$ is a constant in $\mathbb{R}_+$ typically set to $\alpha = 1$. This embedding can be directly applied to the original histograms or used on term-frequency inverse-document-frequency (TFIDF) and its variants (Aizawa 2003; Madsen et al. 2005). These logarithmic maps can be interpreted as particular cases of the embeddings we propose here.

In addition to the logarithm, Hellinger's embedding, which considers the element-wise square-root vector of a histogram ($\mathbf{x} \mapsto \sqrt{\mathbf{x}}$) is particularly popular in computer vision (Perronnin et al. 2010; Vedaldi and Zisserman 2012). This embedding was also considered as an adequate representation to learn Mahanlanobis metrics in the probability simplex as argued by Cuturi and Avis, §6.2.1. Some other explicit feature maps such as χ^2, intersection and Jensen-Shannon are also benchmarked in Vedaldi and Zisserman (2012).

6 Experiments

6.1 Experimental setting and implementation notes

Datasets We evaluate our algorithms on 12 benchmark datasets of various sizes. Table 1 displays their properties and relevant parameters. These datasets include problems such as scene classification, image classification with a single label or multi labels, handwritten digit and text classification. We follow recommended configurations for these datasets. If they are not provided, we randomly generate fivefolds to evaluate in each run. Additionally, we also

repeat the experiments at least 3 times to obtain averaged results, except for PASCAL VOC 2007 and MirFlickr datasets where we use a predefined train and test set.

Parameters of the proposed algorithms We set the target neighborhood size $\kappa = 3$ as a default parameter setting of the LMNN solver.[1] We note that the number of target neighbor κ is not necessary to be equal to parameter k in k-nearest neighbor classification. In our experiments, κ is a fixed number while k varies. We also set the regularization $\mu = 1$ as in LMNN (Weinberger and Saul 2009) while the regularization λ is set to κN (recall that N is the size of the training set), guided by preliminary experiments. For the step size t_0 in the subgradient descent update, we choose from the set $\frac{1}{\kappa N}\{0.01, 0.05, 0.1, 0.5\}$ via cross validation. For the alternating optimization (Algorithm 1), we set $t_{\max} = 20$ iterations (in our experiments, we observe that this number is generous, since usually 6–10 iterations suffice for most datasets, as shown in Figs. 6, 7). For the projected subgradient descent with Nesterov acceleration (PSGD-NES), the algorithm takes less than 500 iterations for converge (usually about 300 iterations, illustrated in Figs. 10, 11). So, we set $t_{\max} = 500$ for the PSGD-NES algorithm.

Dense SIFT features for images Dense SIFT features are computed by operating a SIFT descriptor of 16×16 patches computed over each pixel of an image as in (Le et al. 2011) instead of key points (Lowe 2004) or a grid of points (Lazebnik et al. 2006). Additionally, before computing the dense SIFT, we convert images into gray scale ones to improve robustness. We obtained dense SIFT features by using the LabelMe toolbox[2] (Russell et al. 2008).

6.2 Metrics and metric learning methods

We consider LMNN metric learning for histograms using: their original representation; the **ilr** representation (Sect. 2, Eq. (3)); their Hellinger map. We also include the simple Euclidean distance in our benchmarks. To illustrate the fact that learning the pseudo-count vector **b** results in significant performance improvements, we also conduct experiments with an algorithm that learns **Q** through LMNN but only considers a uniform pseudo-count vector of α chosen in $\{0.0001, 0.001, 0.01, 0.1, 1\}$ by cross validation on the training fold. We call this approach Log-LMNN.

6.3 Scene classification

We conduct experiments on the MIT Scene[3] and UIUC Scene[4] datasets. In these datasets, we select randomly 100 train and 100 test points from each class. Histograms are obtained by using dense SIFT features with bag-of-feature representation (BoF) where the number of visual words is set to 800. We repeat experiments 5 times on each dataset and split randomly onto train and test sets.

The two leftmost graphs in Fig. 2 shows averaged results with error bars on these datasets. The performance of the proposed embedding improves upon that of LMNN on the original histograms by more than 15 % and is slightly better than LMNN combined with the Hellinger

[1] http://www.cse.wustl.edu/~kilian/code/lmnn/lmnn.html.

[2] http://new-labelme.csail.mit.edu/Release3.0/browserTools/php/matlab_toolbox.php.

[3] http://people.csail.mit.edu/torralba/code/spatialenve-lope/.

[4] http://www.cs.illinois.edu/homes/slazebni/research/.

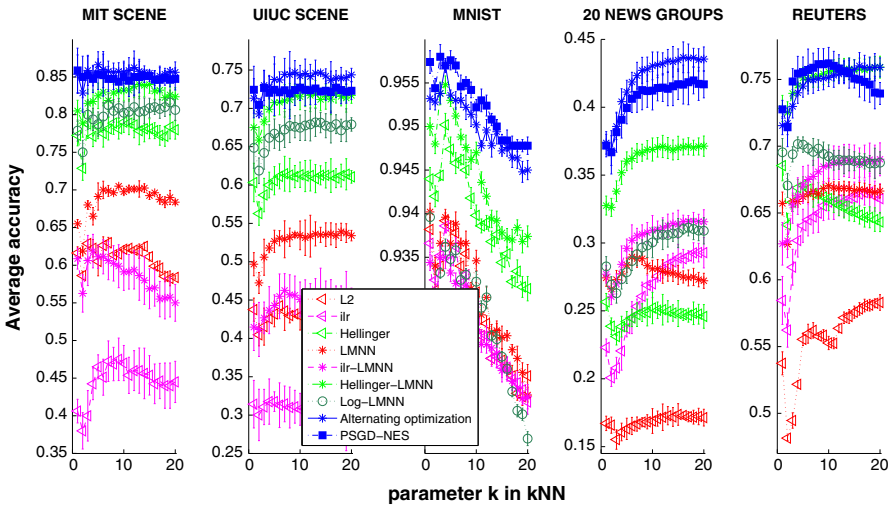

Fig. 2 Classification on scene (MIT Scene & UIUC Scene), handwritten digit (MNIST) and text (20 News Group & Reuters)

map. These graphs also illustrates that Hellinger is an efficient embedding for histograms. The performance of k-NN seeded with the Hellinger distance is even better than that of LMNN in these datasets. The performances of all alternative embeddings with LMNN are better than those with Euclidean distance respectively.

6.4 Handwritten digits classification

We also perform experiments for handwritten digits classification on the MNIST[5] dataset. A feature vector for each point is constructed from a normalized intensity level of each pixel. We randomly choose 500 points disjointly from each class for train and test sets, repeat 5 times for averaged results. The middle graph in Fig. 2 illustrates that the generalized Aitchison embedding also outperforms other alternative embeddings.

6.5 Text classification

We also carry out experiments for text classification on 20 News Groups[6] and Reuters[7] (the 10 largest classes) datasets. In these datasets, we calculate bag of words (BoW) for each document, and then we use topic modelling (LDA) to reduce the dimension of histograms using the *gensim* toolbox.[8] We obtain a histogram of topics for each document (Blei et al. 2003; Blei and Lafferty 2009). We randomly choose 30 points and 50 points from each class in 20 News Groups and Reuters datasets for training, and use the remaining points for testing respectively. We randomly generate 5 different train and test sets for each dataset and average results.

[5] http://yann.lecun.com/exdb/mnist/.

[6] http://qwone.com/~jason/20Newsgroups/.

[7] http://archive.ics.uci.edu/ml/datasets/Reuters--21578+Text+Categorization+Collection.

[8] http://radimrehurek.com/gensim/.

Table 2 Averaged percentage of zero-elements in a histogram (sparseness) of single-label datasets

Dataset	Sparseness (%)
MIT Scene	20.04
UIUC Scene	20.33
DSLR	39.58
WEBCAM	64.44
AMAZON	83.20
OXFORD Flower	1.12
CALTECH-101	13.15
MNIST	80.68
20 News Group	98.01
Reuters	98.00

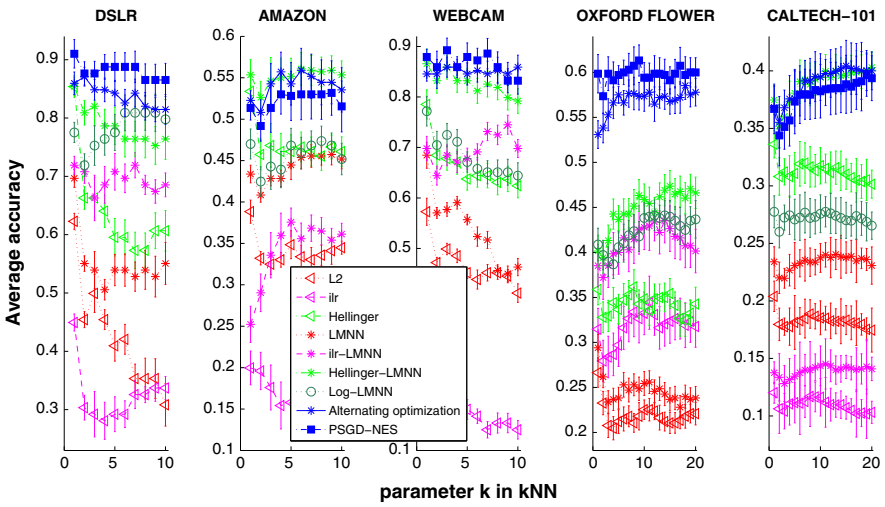

Fig. 3 Single-label object classification on DSLR, AMAZON, WEBCAM, OXFORD FLOWER and CALTECH-101

The two rightmost graphs in Fig. 2 show that the proposed embedding improves the performance of LMNN by more than 10 % on each dataset. It also outperforms the **ilr** and Hellinger representations on these datasets, except for the Reuters dataset where their performances are comparable. Moreover, as in Table 2 which illustrates averaged percentages of zero-elements in a histogram (sparseness), these datasets are very sparse. There are averaged more than 98 % zero-elements in a histogram in these datasets. Therefore, the proposed algorithm may have advantages for very sparse datasets.

6.6 Single-label object classification

DSLR, AMAZON and WEBCAM These datasets[9] are split into fivefolds. Each point is a histogram of visual words obtained by BoF representation on SURF features (Bay et al.

[9] http://www1.icsi.berkeley.edu/~saenko/projects.html.

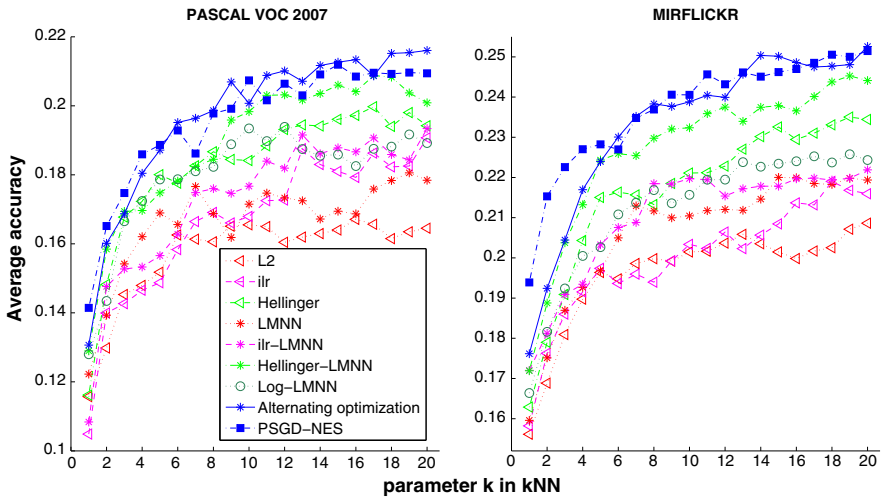

Fig. 4 Multi-label object classification on PASCAL VOC 2007 & MirFlickr

2006) where the code-book size is set to 800. We repeat experiments 5 times on each dataset with different random splits and average results.

The three leftmost graphs in Fig. 3 illustrate that the performance of the proposed embedding outperforms that of `LMNN` on these datasets and even improves about 30, 25 and 10 % on DSLR, WEBCAM and AMAZON dataset respectively. Our proposed algorithm also improves the performances of `Log-LMNN` about 7 %.

OXFORD FLOWER [10] We randomly choose 40 flower images of each class for training and use the rest for testing. We construct histograms using a BoF representation with 400 visual words on a dense SIFT feature and repeat experiments 5 times on different random splits to obtain averaged results. The fourth graph in Fig. 3 shows that the proposed embedding outperforms that of histograms more than 30 %, and also improves about 15 % comparing to the **ilr** embedding as well as the Hellinger representation with `LMNN`. As showed in Table 2, this dataset is highly dense since there are only about 1 % zero-elements in a histogram. This suggests that our approaches might work better with dense datasets.

CALTECH-101 We randomly choose 30 images for training and up to 50 other images for testing. We use BoF representation with 400 visual words on a dense SIFT feature to construct histograms for each image. The rightmost graph in Fig. 3 shows averaged results on 3 different random splits of the CALTECH-101[11] dataset, illustrating again the interest of our approach.

6.7 Multi-label object classification

We evaluate the proposed method on multi-label image categorization using the PASCAL VOC 2007[12] and MirFlickr[13] datasets. We follow the guidelines to define the train and test

[10] http://www.robots.ox.ac.uk/~vgg/data/flowers/17/.

[11] http://www.vision.caltech.edu/Image_Datasets/Cal-tech101/.

[12] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/.

[13] http://press.liacs.nl/mirflickr/.

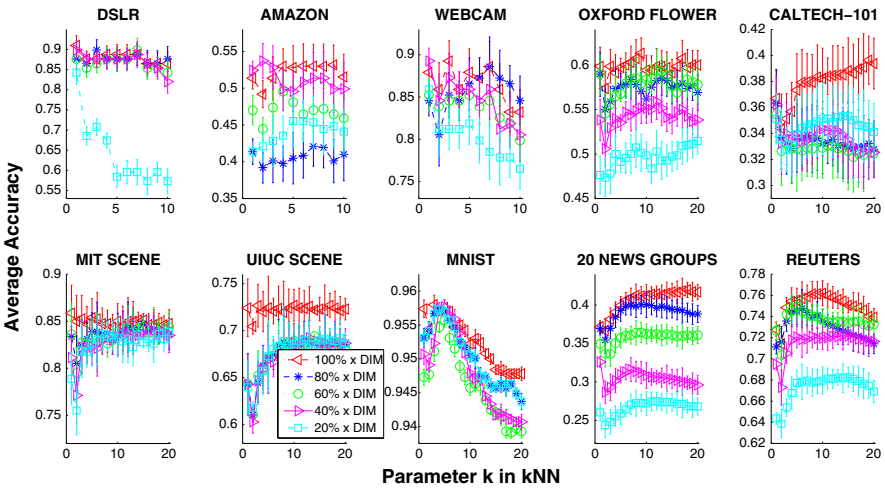

Fig. 5 Classification results for low-rank generalized Aitchison embedding

sets. Histograms for each image are built in these datasets based on BoF representation with 100 visual words on a dense hue feature. Then, we employ a one-versus-all strategy for k-NN classification and calculate averaged precisions for each dataset. Figure 4 illustrates that the proposed embedding outperforms original, **ilr**, and Hellinger representation with LMNN again. Additionally, the performance of Hellinger distance is better than that of LMNN and comparative with that of Log-LMNN in these datasets.

6.8 Low-rank embeddings

We conduct experiments for the low-rank version of our algorithm, where the dimension is set to {80, 60, 40, 20 %} of the original dimension of the single-label datasets. Figure 5 indicates that reducing rank can be carried out to accelerate computations, but this speed up can come, depending on the dataset, at the expense of a degradation in performance.

7 Experimental behavior of the algorithm

7.1 Convergence speed

Figures 6 and 7 illustrate the convergence of the objective with respect to computational time on a log-log scale. We consider the naive alternating optimization approach (Sect. 4.2), a standard projected subgradient descent, projected subgradient descent with Nesterov acceleration (Sect. 4.3), and a version with adaptive restart (PSGD-NES-AR in Sect. 4.5). We use the LMNN solver directly and measure raw time using a single core. Gaps in that curve indicate the value of the objective before and after running the LMNN solver.

The naive alternating optimization has computational cost that is about one order of magnitude larger than that of a direct application of LMNN. This factor appears because we run the LMNN solver multiple times. The burden of optimizing the pseudo-count vector is small due to the fact that the gradient has a closed-form solution for each pair in the

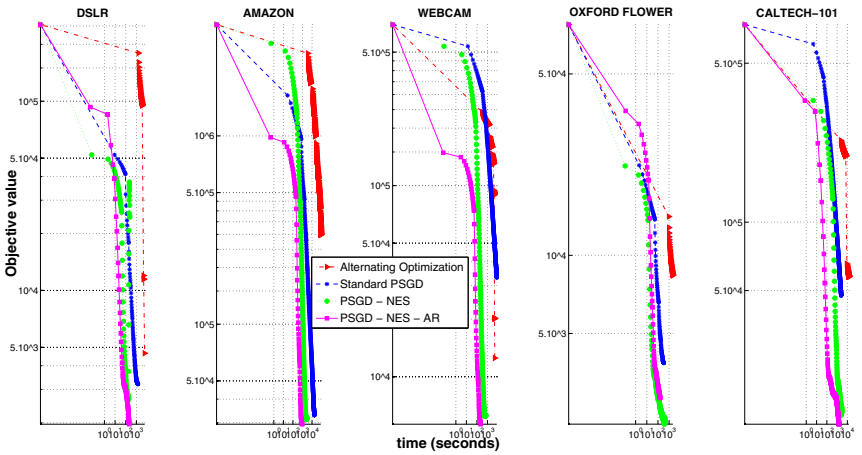

Fig. 6 Log–log plot illustration for the relation between behavior of the objective function and computational time in the proposed algorithms on DSLR, AMAZON, WEBCAM, OXFORD FLOWER and CALTECH-101

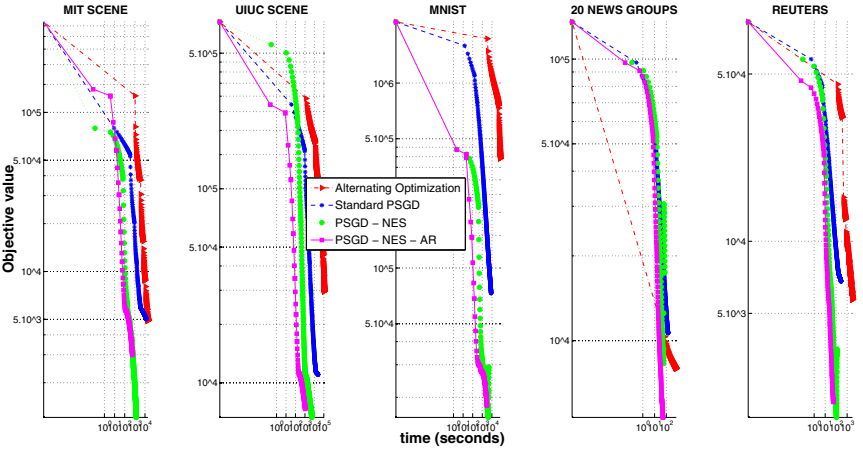

Fig. 7 Log–log plot illustration for the relation between behavior of the objective function and computational time in the proposed algorithms on scene (MIT Scene & UIUC Scene), handwritten digit (MNIST) and text (20 News Group & Reuters)

objective function. We only need to run a few iterations of the LMNN algorithm using a warm start when alternating. Our experiments show that we only need to run 6–10 alternating iterations for these datasets, but each iteration is costly. These results show the interest of using Nesterov acceleration scheme here, and even suggest adopting the adaptive restart heuristic of O'Donoghue and Candès (2013).

7.2 Sensitivity to parameters

Target neighbors Figures 8 and 9 illustrate the effect of the number of target neighbors κ on the results of our algorithms. We evaluate for $\kappa = \{1, 3, 5, 7, 9\}$ for single-label datasets, except $\kappa = \{7, 9\}$ for DSLR and $\kappa = 9$ for WEBCAM due to the size of the smallest

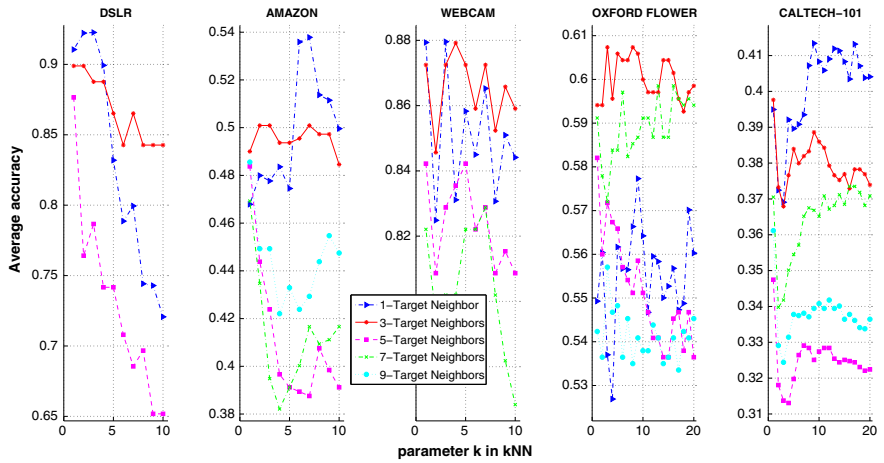

Fig. 8 Illustration for the effect of target neighbors in the PSGD-NES on DSLR, AMAZON, WEBCAM, OXFORD FLOWER and CALTECH-101

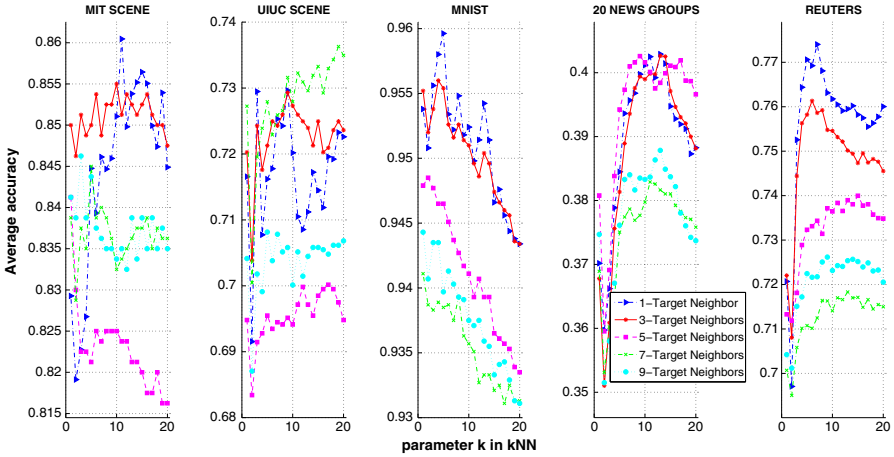

Fig. 9 Illustration for the effect of target neighbors in the PSGD-NES scene (MIT Scene & UIUC Scene), handwritten digit (MNIST) and text (20 News Group & Reuters)

class in these datasets. These results suggest that the number of target neighbors has a large impact and should remain low, both from a computational viewpoint and performances of the algorithm. Figures 8 and 9 also show that 3-target-neighbor setup is an appropriate choice for those evaluated datasets.

Average test accuracy over iteration count Figures 10 and 11 show the average test accuracy over iteration count. The curves of average test accuracy value seem to increase monotonically with the iteration count, therefore suggesting that our algorithms do not overfit training data in these evaluations.

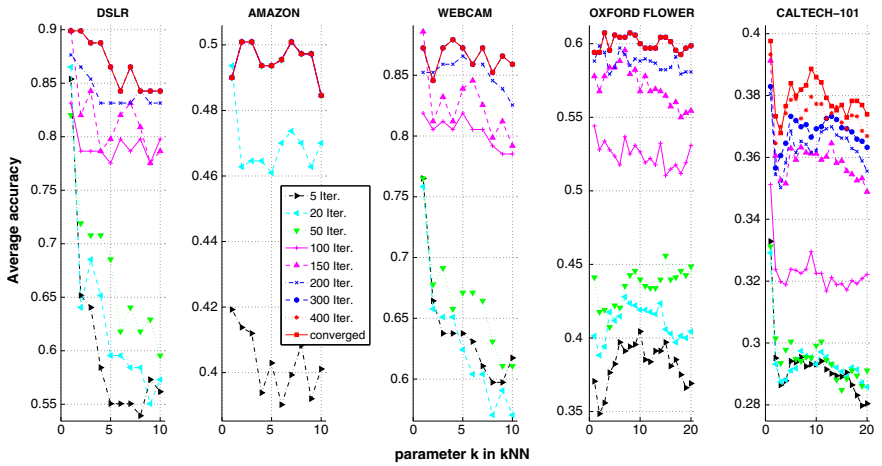

Fig. 10 Illustration for the average test accuracy over iteration count of the PSGD-NES to indicate that the algorithm do not overfit to training data on DSLR, AMAZON, WEBCAM, OXFORD FLOWER and CALTECH-101

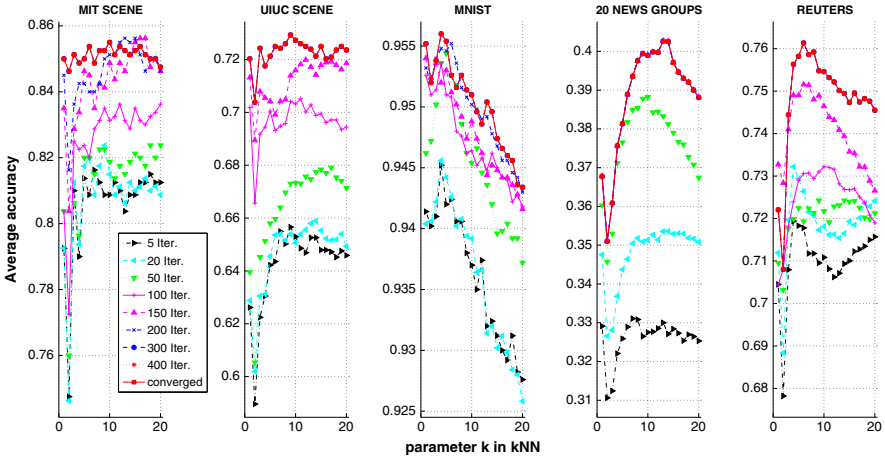

Fig. 11 Illustration for the average test accuracy over iteration count of the PSGD-NES to indicate that the algorithm do not overfit to training data scene (MIT Scene & UIUC Scene), handwritten digit (MNIST) and text (20 News Group & Reuters)

8 Conclusion

We have shown that a generalized family of embeddings for histograms coupled with different procedures to estimate its parameters can be effective to represent histograms in Euclidean spaces. Our variations outperform other common approaches such as the Hellinger map or Aitchison's original embeddings. Rather than using an alternative optimization scheme and use `LMNN` solvers, our results indicated that a simple accelerated subgradient method provides the best results both in performance and computational time. Other variations, such as learning a low-rank embedding or using adaptive restart heuristic for PSGD-NES, can also prove beneficial, depending on the datasets.

Acknowledgments TL acknowledges the support of the MEXT scholarship 123353. MC acknowledges the support of the Japanese Society for the Promotion of Science Grant 25540100.

References

Aitchison, J. (1982). The statistical analysis of compositional data. *Journal of the Royal Statistical Society*, *44*, 139–177.

Aitchison, J. (1986). *The statistical analysis of compositional data*. London: Chapman and Hall Ltd.

Aitchison, J. (2003). A concise guide to compositional data analysis. In *CDA workshop*.

Aitchison, J., & Lauder, I. J. (1985). Kernel density estimation for compositional data. *Applied statistics*, *34*, 129–137.

Aitchison, J., & Shen, S. M. (1980). Logistic-normal distributions: Some properties and uses. *Biometrika*, *67*, 261–272.

Aizawa, A. (2003). An information-theoretic perspective of tf-idf measures. *Information Processing and Management*, *39*(1), 45–65.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval* (Vol. 463). New York: ACM press.

Bay, H., Tuytelaars, T.,& Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision* (pp. 404–417).

Blei, D., & Lafferty, J. (2006). Correlated topic models. In B. Schökopf, J. C. Platt & T. Hoffman (Eds.), *Advances in Neural Information Processing Systems* (pp. 147–154). Vancouver, Canada: MIT Press.

Blei, D., & Lafferty, J. (2009). Topic models. In A. Srivastava & M. Sahami (Eds.), *Text mining: Classification, clustering, and applications*. Boca Raton, FL: Chapman & Hall, CRC Press.

Blei, D., Ng, A., & Jordan, M. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Burge, C., Campbell, A. M., & Karlin, S. (1992). Over-and under-representation of short oligonucleotides in DNA sequences. *National Academy of Sciences*, *89*(4), 1358–1362.

Campbell, W. M., & Richardson, F. S. (2007). Discriminative keyword selection using support vector machines. In J. C. Platt, D. Koller, Y. Singer & S. T. Roweis (Eds.), *Advances in NeuralInformation Processing Systems*. Vancouver, Canada: Curran Associates, Inc.

Campbell, W. M., Campbell, J. P., Reynolds, D. A., Jones, D. A., & Leek,T. R. (2003). Phonetic speaker recognition with support vector machines. In S. Thrun, L. K. Saul & B. Schökopf (Eds.), *Advances in Neural Information Processing Systems*. Vancouver, Canada: MIT Press.

Cuturi, M., & Avis, D. (2014). Ground metric learning. *Journal of Machine Learning Research*, *15*(1), 533–564.

Cuturi, M., & Avis, D. (2011). Ground metric learning. arXiv preprint arXiv:1110.2306.

Davis, J. V., Kulis, B., Jain, P., Sra, S., & Dhillon, I. S. (2007). Information-theoretic metric learning. In *International conference on machine learning*, pp. 209–216.

Doddington, G. (2001). Speaker recognition based on idiolectal differences between speakers. In *P. Dalsgaard*, B. Lindberg, H. Benner, & Z.-H. Tan (Eds.), *Eurospeech* (pp. 2521–2524). Aalborg, Denmark: Center for Personkommunikation, Aalborg University.

Egozcue, J. J., Pawlowsky-Glahn, V., Mateu-Figueras, G., & Barcel-Vidal, C. (2003). Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, *35*(3), 279–300.

Erhan, S., Marzolf, T., & Cohen, L. (1980). Amino-acid neighborhood relationships in proteins. Breakdown of amino-acid sequences into overlapping doublets, triplets and quadruplets. *International Journal of Bio-Medical Computing*, *11*(1), 67–75.

Globerson, A., & Roweis, S. T. (2005). Metric learning by collapsing classes. In Y. Weiss, B. *Schökopf* & J. C. Platt (Eds.), *Advances in Neural Information ProcessingSystems* (pp. 451–458). Vancouver, Canada: MIT Press.

Goldberger, J., Roweis, S. T., Hinton, G. E., & Salakhutdinov, R. (2004). Neighbourhood components analysis. In L. K. Saul, Y. Weiss & L. Bottou (Eds.), *Advances in Neural Information Processing Systems*. Vancouver, Canada: MIT Press.

Joachims, T. (2002). *Learning to classify text using support vector machines: Methods: Theory and algorithms*. Berlin: Springer.

Julesz, B. (1981). Textons, the elements of texture perception, and their interactions. *Nature*, *290*(5802), 91–97.

Kedem, D., Tyree, S., Weinberger, K. Q., Sha, F., & Lanckriet, G. (2012). Nonlinear metric learning. In F. Pereira, C . J. C. Burges, L. Bottou & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (pp. 2582–2590). Nevada: Curran Associates, Inc.

Kwok, J. T., & Tsang, I. W. (2003). Learning with idealized kernels. In *International conference on machine learning* (pp. 400–407).
Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Computer Vision and Pattern Recognition*, *2*, 2169–2178.
Le, T.,& Cuturi, M. (2013). Generalized aitchison embeddings for histograms. In *Asian conference on machine learning* (pp. 293–308).
Le, T., Kang, Y., Sugimoto, A., Tran, S., & Nguyen, T. (2011). Hierarchical spatial matching kernel for image categorization. In *International conference on image analysis and recognition* (pp. 141–151).
Leslie, C. S., Eskin, E., & Noble, W. S. (2002). The spectrum kernel: A string kernel for SVM protein classification. In *Pacific symposium on biocomputing* (Vol. 7, pp. 566–575).
Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, *5*, 361–397.
Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.
Madsen, R. E., Kauchak, D., & Elkan, C. (2005). Modeling word burstiness using the dirichlet distribution. In *International conference on machine learning* (pp. 545–552).
Nesterov, Y. (1983). A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady* (Vol. 27, pp. 372–376).
Nesterov, Y. (2004). *Introductory lectures on convex optimization: A basic course*. Berlin: Springer.
O'Donoghue, B., & Candès, E. (2013). Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, *13*, 1–18.
Perronnin, F., Sánchez, J., & Liu, Y. (2010). Large-scale image categorization with explicit data embedding. In *Computer vision and pattern recogition* (pp. 2297–2304). San Francisco, CA: Curran Associates, Inc.
Rennie, J. D., Shih, L., Teevan, J., & Karger, D. (2003). Tackling the poor assumptions of naive bayes text classifiers. In *International conference on machine learning* (Vol. 3, pp. 616–623).
Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, *77*(1–3), 157–173.
Salton, G. (1989). *Automatic text processing: The transformation, analysis, and retrieval of*. Reading: Addison-Wesley.
Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. New York: McGraw-Hill.
Schultz, M., & Joachims, T. (2003). Learning a distance metric from relative comparisons. In S. Thrun , L. K. Saul & B. *Schökopf* (Eds.), *Advances in Neural Information Processing Systems* (Vol. 16, p. 41). Vancouver, Canada: MIT Press.
Shalev-Shwartz, S., Singer, Y., & Ng, A. Y. (2004). Online and batch learning of pseudo-metrics. In *International conference on machine learning* (p. 94).
Sivic, J., & Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In *International conference on computer vision*.
Torresani, L., & Lee, K. (2006). Large margin component analysis. In *Advances in Neural Information Processing Systems* (pp. 1385–1392).
Vedaldi, A., & Zisserman, A. (2012). Efficient additive kernels via explicit feature maps. *IEEE Pattern Analysis and Machine Intelligence*, *34*(3), 480–492.
Weinberger, K. Q., & Saul, L. K. (2008). Fast solvers and efficient implementations for distance metric learning. In *International conference on machine learning* (pp. 1160–1167).
Weinberger, K. Q., & Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, *10*, 207–244.
Weinberger, K. Q., Blitzer, J., & Saul, L. (2006). Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems* (pp. 1473–1480).
Xing, E. P., Ng, A. Y., Jordan, M. I., & Russell, S. J. (2002). Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems* (pp. 1473–1480).