Least-Squares Log-Density Gradient Clustering for Riemannian Manifolds

Mina Ashizawa[1] **Hiroaki Sasaki**[2,3] **Tomoya Sakai**[1] **Masashi Sugiyama**[3,1]
[1]The University of Tokyo, Japan [2]Nara Institute of Science and Technology, Japan [3]RIKEN, Japan

Abstract

Mean shift is a mode-seeking clustering algorithm that has been successfully used in a wide range of applications such as image segmentation and object tracking. To further improve the clustering performance, mean shift has been extended to various directions, including generalization to handle data on Riemannian manifolds and extension to directly estimating the log-density gradient without density estimation. In this paper, we combine these ideas and propose a novel mode-seeking algorithm for Riemannian manifolds with direct log-density gradient estimation. Although the idea of combining the two extensions is rather straightforward, directly estimating the log-density gradient on Riemannian manifolds is mathematically challenging. We will provide a mathematically sound algorithm and demonstrate its usefulness through experiments.

1 Introduction

Clustering is one of the most important unsupervised learning tasks in machine learning and has been extensively studied for decades (Clarke et al., 2009; Murphy, 2012). Among various different types of clustering algorithms, *mode-seeking* is a well-studied and practically useful approach. *Mean shift* (Fukunaga and Hostetler, 1975; Cheng, 1995; Comaniciu and Meer, 2002; Carreira-Perpinán, 2015) is a seminal algorithm for mode-seeking clustering: Kernel density estimation is first performed on given data points and then the data points are updated along the gradient of the estimated density towards the modes. Finally, the data points which converged to the same mode are given the same cluster label. A notable advantage of mean shift is that it does not require to specify the number of clusters in advance. Thanks to this useful property, mean shift has been successfully employed in a wide range of real-world applications such as image segmentation (Wang et al., 2004; Tao et al., 2007) and object tracking (Comaniciu et al., 2000; Collins, 2003).

The original mean shift algorithm considers data points in the Euclidean space. However, in practice, data points sometimes lie on a structured space such as the *Lie group* and *Grassmann manifold*. For data on such a structured space, kernel density estimation and gradient ascent with the Euclidean metric do not necessarily perform appropriately. To cope with this problem, mean shift has been extended to Riemannian manifolds (Boothby, 2003) and demonstrated to work well in experiments (Tuzel et al., 2005; Subbarao and Meer, 2006, 2009; Cetingul and Vidal, 2009).

Another important extension of mean shift is to avoid density estimation. The original mean shift algorithm uses kernel density estimation, which tends to perform poorly when the data dimension is high. Furthermore, a good density estimator does not necessarily mean a good density gradient estimator, and thus the two-step approach of first estimating the density and then computing its gradient does not always perform well. To cope with this problem, a method to directly estimate the log-density gradient without density estimation has been developed (Cox, 1985; Sasaki et al., 2014), and a mode-seeking clustering algorithm based on the direct log-density gradient estimator was experimentally shown to work well (Sasaki et al., 2014).

The purpose of this paper is to combine these two extensions and propose a novel clustering algorithm based on direct log-density gradient estimation and mode-seeking on Riemannian manifolds. Although the idea of combining the two extensions is rather straightforward, directly estimating the density gradient on Riemannian manifolds is mathematically challenging. We will provide a mathematically sound algorithm and demonstrate its usefulness through experiments.

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

2 Problem Formulation

In this section, we formulate the clustering problem by mode-seeking and review existing algorithms.

Clustering by Mode-Seeking: Suppose that we are given independent and identically distributed samples of size n on $\mathbb{R}^d$, $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^n$, with unknown probability density $p(\boldsymbol{x})$. The goal of clustering is to split the set $\mathcal{X}$ into c disjoint subsets $\{\mathcal{X}_i\}_{i=1}^c$ so that samples in each subset share similar properties while samples in different subsets have different properties.

Various types of clustering algorithms have been explored so far (Clarke et al., 2009; Murphy, 2012). Among them, *mode-seeking* is one of the popular and well-studied approaches (Fukunaga and Hostetler, 1975; Cheng, 1995; Comaniciu and Meer, 2002; Carreira-Perpinán, 2015). In mode-seeking clustering, data samples $\{\boldsymbol{x}_i\}_{i=1}^n$ are first gathered to the modes of data density $p(\boldsymbol{x})$ by, e.g., gradient ascent $\boldsymbol{x} \leftarrow \boldsymbol{x} + \varepsilon \nabla p(\boldsymbol{x})$, where $\varepsilon > 0$ is the step size and $\nabla p(\boldsymbol{x})$ is the gradient of the density function $p(\boldsymbol{x})$ with respect to $\boldsymbol{x} = (x^{(1)}, \ldots, x^{(d)})^\top$. Then, data samples which converged to the same mode are given the same cluster label.

Below, we review representative mode-seeking clustering algorithms.

Mean Shift Clustering: *Mean shift* is a seminal algorithm of mode-seeking clustering (Fukunaga and Hostetler, 1975; Cheng, 1995; Comaniciu and Meer, 2002). In the mean shift algorithm, the probability density $p(\boldsymbol{x})$ is first learned by kernel density estimation:

$$\widehat{p}(\boldsymbol{x}) = \frac{c_{k,\sigma}}{n} \sum_{i=1}^{n} k\Big(\Big\|\frac{\boldsymbol{x} - \boldsymbol{x}_i}{\sigma}\Big\|^2\Big), \tag{1}$$

where $k(t)$ is a non-negative function, $\sigma > 0$ is the bandwidth, and $c_{k,\sigma}$ is the normalization constant such that the integration of $\widehat{p}(\boldsymbol{x})$ is equal to 1. As function $k(t)$, the exponential decaying function $k(t) = \exp(-t/2)$ is often used in practice, which yields the Gaussian kernel:

$$k\Big(\Big\|\frac{\boldsymbol{x} - \boldsymbol{x}_i}{\sigma}\Big\|^2\Big) = \exp\Big(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}{2\sigma^2}\Big).$$

The bandwidth σ can be systematically chosen by cross-validation with respect to the log-likelihood or squared error criteria.

Next, the gradient of the kernel density estimator $\widehat{p}(\boldsymbol{x})$ is computed:

$$\nabla \widehat{p}(\boldsymbol{x}) = \frac{c_{k',\sigma}}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{x}) k'\Big(\Big\|\frac{\boldsymbol{x} - \boldsymbol{x}_i}{\sigma}\Big\|^2\Big) = \epsilon(\boldsymbol{x}) \boldsymbol{m}(\boldsymbol{x}),$$

where $c_{k',\sigma} = -2c_{k,\sigma}/\sigma^2$, k' is the derivative of k,

$$\epsilon(\boldsymbol{x}) := \frac{c_{k',\sigma}}{n} \sum_{i=1}^{n} k'\Big(\Big\|\frac{\boldsymbol{x} - \boldsymbol{x}_i}{\sigma}\Big\|^2\Big) > 0,$$

and $\boldsymbol{m}(\boldsymbol{x})$ is called the *mean shift vector* (Comaniciu and Meer, 2002):

$$\boldsymbol{m}(\boldsymbol{x}) := \frac{\sum_{i=1}^n \boldsymbol{x}_i k'(\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{\sigma}\|^2)}{\sum_{i'=1}^n k'(\|\frac{\boldsymbol{x}-\boldsymbol{x}_{i'}}{\sigma}\|^2)} - \boldsymbol{x}.$$

To obtain the modes of data density, the mean shift algorithm uses the *fixed-point iteration* for mode-seeking. More specifically, a necessary condition for local maximum $\nabla \widehat{p}(\boldsymbol{x}) = \boldsymbol{0}$ implies $\boldsymbol{m}(\boldsymbol{x}) = \boldsymbol{0}$, which yields $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{m}(\boldsymbol{x})$. Since this fixed-point update rule can be rewritten as $\boldsymbol{x} \leftarrow \boldsymbol{x} + \frac{1}{\epsilon(\boldsymbol{x})} \nabla \widehat{p}(\boldsymbol{x})$, it can be interpreted as gradient ascent with adaptive step size $1/\epsilon(\boldsymbol{x}) > 0$.

After repeating the fixed-point update for all samples $\{\boldsymbol{x}_i\}_{i=1}^n$ until convergence, samples which converge to the same mode are given the same cluster label.

The mean shift algorithm has been successfully employed in various real-world applications such as image segmentation (Wang et al., 2004; Tao et al., 2007) and object tracking (Comaniciu et al., 2000; Collins, 2003).

Mean Shift Clustering on Riemannian Manifolds: The original mean shift algorithm considers data points in the Euclidean space. However, in practice, data points sometimes lie on a structured space such as the Lie group and Grassmann manifold. For data on such a structured space, kernel density estimation and gradient ascent with the Euclidean metric do not necessarily perform appropriately. To cope with this problem, the mean shift algorithm has been extended to Riemannian manifolds (Tuzel et al., 2005; Subbarao and Meer, 2006, 2009; Cetingul and Vidal, 2009). Here we briefly review such an extention. Let us consider data points $\{\boldsymbol{X}_i\}_{i=1}^n$ on Riemannian manifold $\mathcal{M}^m$ of dimension $m \leq d$ embedded in $\mathbb{R}^d$.

As shown in Eq.(1), kernel density estimation in the original mean shift algorithm uses the squared Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{x}_i$, i.e., $\|\boldsymbol{x} - \boldsymbol{x}_i\|^2$. For data points on a Riemannian manifold, this may be replaced with the squared *geodesic distance* between $\boldsymbol{X}$ and $\boldsymbol{X}_i$:

$$\widehat{p}(\boldsymbol{X}) = \frac{c_{k,\sigma,\delta}}{n} \sum_{i=1}^{n} k\left(\frac{\delta(\boldsymbol{X}, \boldsymbol{X}_i)^2}{\sigma^2}\right), \tag{2}$$

where $\delta(\boldsymbol{X}, \boldsymbol{X}')$ denotes the geodesic distance between $\boldsymbol{X}$ and $\boldsymbol{X}'$, and $c_{k,\sigma,\delta}$ is a positive constant.[1]

[1] Strictly speaking, Eq.(2) is inappropriate as a density

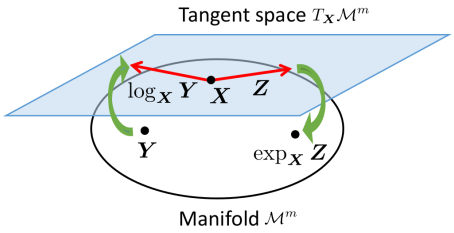


Figure 1: The exponential map and the logarithm map. The exponential map at the point $\boldsymbol{X}$ transforms points on the tangent space at $\boldsymbol{X}$ to the manifold, while the logarithm map at $\boldsymbol{X}$ transforms points on the manifold to the tangent space at $\boldsymbol{X}$.

Then, the gradient of the estimated density is given by

$$\nabla\widehat{p}(\boldsymbol{X}) = \frac{-c_{k',\sigma,\delta}}{2n}\sum_{i=1}^{n}\left[\nabla\delta(\boldsymbol{X},\boldsymbol{X}_i)^2\right]k'\left(\frac{\delta(\boldsymbol{X},\boldsymbol{X}_i)^2}{\sigma^2}\right)$$
$$= \frac{c_{k',\sigma,\delta}}{n}\sum_{i=1}^{n}\left[\log_{\boldsymbol{X}}\boldsymbol{X}_i\right]k'\left(\frac{\delta(\boldsymbol{X},\boldsymbol{X}_i)^2}{\sigma^2}\right), \quad (3)$$

where $c_{k',\sigma,\delta} = -2c_{k,\sigma,\delta}/\sigma^2$. $\log_{\boldsymbol{X}}\boldsymbol{X}_i$ denotes the *logarithm map* of $\boldsymbol{X}_i$ at $\boldsymbol{X}$, which satisfies the following relation (Boothby, 2003):

$$\log_{\boldsymbol{X}}\boldsymbol{X}_i = -\frac{1}{2}\nabla\delta(\boldsymbol{X},\boldsymbol{X}_i)^2. \quad (4)$$

In the same way as the original mean shift algorithm, Eq.(3) can be expressed as $\nabla\widehat{p}(\boldsymbol{X}) = \epsilon(\boldsymbol{X})\boldsymbol{M}(\boldsymbol{X})$, where $\epsilon(\boldsymbol{X}) := \frac{c_{k',\sigma,\delta}}{n}\sum_{i=1}^{n}k'\left(\frac{\delta(\boldsymbol{X},\boldsymbol{X}_i)^2}{\sigma^2}\right) > 0$ and $\boldsymbol{M}(\boldsymbol{X})$ is the mean shift vector defined as

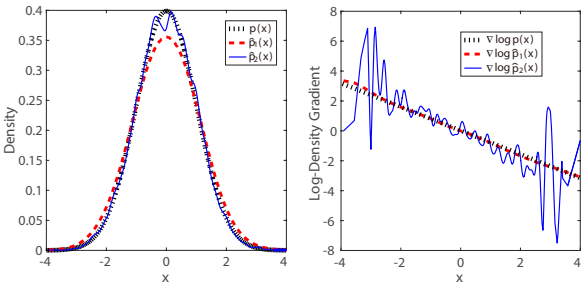
$$\boldsymbol{M}(\boldsymbol{X}) := \frac{\sum_{i=1}^{n}\left[\log_{\boldsymbol{X}}\boldsymbol{X}_i\right]k'\left(\frac{\delta(\boldsymbol{X},\boldsymbol{X}_i)^2}{\sigma^2}\right)}{\sum_{i'=1}^{n}k'\left(\frac{\delta(\boldsymbol{X},\boldsymbol{X}_{i'})^2}{\sigma^2}\right)}.$$

This shows that mean shift vector $\boldsymbol{M}(\boldsymbol{X})$ lies on the *tangent space* at $\boldsymbol{X}$, which is denoted by $T_{\boldsymbol{X}}\mathcal{M}^m$ (see Figure 1).

Thus, when a point $\boldsymbol{X}$ on the manifold is updated along with this vector, it no longer lies on the manifold. To cope with this problem, the updated point is projected back to the manifold by the *exponential map* $\boldsymbol{X} \leftarrow \exp_{\boldsymbol{X}}\boldsymbol{M}(\boldsymbol{X})$ (see Figure 1 again) (Boothby, 2003).

After repeating this update for all samples $\{\boldsymbol{X}_i\}_{i=1}^n$ until convergence, samples which converge to the same mode are given the same cluster label.

estimator on Riemannian manifolds: To ensure that the integration of $\widehat{p}(\boldsymbol{X})$ is equal to 1, the normalizing constant of each $k\left(\frac{\delta(\boldsymbol{X},\boldsymbol{X}_i)^2}{\sigma^2}\right)$ has to depend on $\boldsymbol{X}_i$ (Pelletier, 2005; Arvanitidis et al., 2016). Nonetheless, Eq.(2) has been employed in the mean shift algorithm because of its computational efficiency and simple form (Subbarao and Meer, 2009).



(a) Data densities (b) Log-density gradients

Figure 2: Density estimation and log-density gradient estimation. $\widehat{p}_2(x)$ is a better estimate of true density $p(x)$ than $\widehat{p}_1(x)$, but $\nabla\log\widehat{p}_2(x)$ is a worse estimate of true log-density gradient $\nabla\log p(x)$ than $\nabla\log\widehat{p}_1(x)$. Thus, a good density estimator is not necessarily a good log-density gradient estimator.

The mean shift algorithm on Riemannian manifolds has been shown to work well experimentally (Tuzel et al., 2005; Subbarao and Meer, 2006, 2009; Cetingul and Vidal, 2009).

Least-Squares Log-Density Gradient Clustering: Another important extension of the original mean shift algorithm is to avoid density estimation (Sasaki et al., 2014). Kernel density estimation used in the original mean shift algorithm tends to perform poorly when the data dimension d is high. Furthermore, a good density estimator does not necessarily mean a good density gradient estimator (see Figure 2), and thus the two-step approach of first estimating the density and then computing its gradient does not always perform well. To cope with these problems, a method to directly estimate the density gradient without density estimation has been developed (Cox, 1985; Sasaki et al., 2014). Here, we briefly review the direct log-density gradient estimator called the *least-squares log-density gradient* (LSLDG) and the mode-seeking clustering algorithm based on it called *LSLDG clustering* (Sasaki et al., 2014).

Let $\boldsymbol{g}(\boldsymbol{x}) := (g^{(1)}(\boldsymbol{x}),\ldots,g^{(d)}(\boldsymbol{x}))^\top$ be the gradient of the log-density function, where $g^{(j)}(\boldsymbol{x}) := \partial_j\log p(\boldsymbol{x})$ and $\partial_j = \frac{\partial}{\partial x^{(j)}}$. The key idea of LSLDG is to directly fit a model $\widetilde{g}^{(j)}(\boldsymbol{x})$ to the true log-density gradient $g^{(j)}(\boldsymbol{x})$ under the squared loss:

$$J^{(j)}(\widetilde{g}^{(j)}(\boldsymbol{x})) := \int\left(\widetilde{g}^{(j)}(\boldsymbol{x}) - g^{(j)}(\boldsymbol{x})\right)^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - C$$
$$= \int\widetilde{g}^{(j)}(\boldsymbol{x})^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - 2\int\widetilde{g}^{(j)}(\boldsymbol{x})\partial_j p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$
$$= \int\widetilde{g}^{(j)}(\boldsymbol{x})^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} + 2\int\partial_j\widetilde{g}^{(j)}(\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x},$$

where $C = \int g^{(j)}(\boldsymbol{x})^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$ and the last equality follows from *integration by parts* under

$\lim_{x^{(j)}\to\pm\infty} \widetilde{g}^{(j)}(\boldsymbol{x})p(\boldsymbol{x}) = 0$. Then the empirical approximation of $J^{(j)}$ is given as

$$J^{(j)}(\widetilde{g}^{(j)}(\boldsymbol{x})) \approx \frac{1}{n}\sum_{i=1}^{n} \widetilde{g}^{(j)}(\boldsymbol{x}_i)^2 + \frac{2}{n}\sum_{i=1}^{n} \partial_j \widetilde{g}^{(j)}(\boldsymbol{x}_i). \quad (5)$$

As the log-density gradient model $\widetilde{g}^{(j)}(\boldsymbol{x})$, a linear-in-parameter model is used:

$$\widetilde{g}^{(j)}(\boldsymbol{x}) = \boldsymbol{\theta}^{(j)\top}\boldsymbol{\psi}^{(j)}(\boldsymbol{x}) = \sum_{l=1}^{b} \theta_l^{(j)} \psi_l^{(j)}(\boldsymbol{x}),$$

where b denotes the number of parameters, $\boldsymbol{\theta}^{(j)} \in \mathbb{R}^b$ is the parameter vector, and $\boldsymbol{\psi}^{(j)}(\boldsymbol{x}) \in \mathbb{R}^b$ is the vector of basis functions. By plugging this linear model into Eq.(5) and adding the ℓ_2-regularizer to avoid overfitting, the following optimization problem is obtained:

$$\widehat{\boldsymbol{\theta}}^{(j)} = \operatorname*{argmin}_{\boldsymbol{\theta}^{(j)}\in\mathbb{R}^b} \left[\boldsymbol{\theta}^{(j)\top}\widehat{\boldsymbol{G}}^{(j)}\boldsymbol{\theta}^{(j)} + 2\boldsymbol{\theta}^{(j)\top}\widehat{\boldsymbol{h}}^{(j)} + \lambda^{(j)}\boldsymbol{\theta}^{(j)\top}\boldsymbol{\theta}^{(j)}\right],$$

where $\lambda^{(j)} \geq 0$ is the regularization parameter, and $\widehat{\boldsymbol{G}}^{(j)}$ and $\widehat{\boldsymbol{h}}^{(j)}$ are defined as

$$\widehat{\boldsymbol{G}}^{(j)} := \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\psi}^{(j)}(\boldsymbol{x}_i)\boldsymbol{\psi}^{(j)}(\boldsymbol{x}_i)^\top, \quad \widehat{\boldsymbol{h}}^{(j)} := \frac{1}{n}\sum_{i=1}^{n} \partial_j \boldsymbol{\psi}^{(j)}(\boldsymbol{x}_i)$$

The optimal solution $\widehat{\boldsymbol{\theta}}^{(j)}$ is obtained analytically by

$$\widehat{\boldsymbol{\theta}}^{(j)} = -(\widehat{\boldsymbol{G}}^{(j)} + \lambda^{(j)}\boldsymbol{I}_b)^{-1}\widehat{\boldsymbol{h}}^{(j)},$$

where $\boldsymbol{I}_b$ denotes the $b \times b$ identity matrix. All hyper-parameters such as the regularization parameter $\lambda^{(j)}$ and basis parameters in $\boldsymbol{\psi}^{(j)}(\boldsymbol{x})$ can be systematically chosen by cross-validation with respect to the squared error criterion $J^{(j)}$. This direct log-density gradient estimator is called LSLDG.

To derive a mean shift like fixed-point algorithm from LSLDG, for the Gaussian kernel, $\phi_l^{(j)}(\boldsymbol{x}) := \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}_l\|^2}{2\sigma^{(j)2}}\right)$, where the centers $\{\boldsymbol{c}_l\}_{l=1}^b$ are chosen randomly from $\{\boldsymbol{x}_i\}_{i=1}^n$ without overlap, its partial derivative was proposed to be used as basis functions (Sasaki et al., 2014):

$$\psi_l^{(j)}(\boldsymbol{x}) := \partial_j \phi_l^{(j)}(\boldsymbol{x}) = \frac{1}{\sigma^{(j)2}}(c_l^{(j)} - x^{(j)}) \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}_l\|^2}{2\sigma^{(j)2}}\right).$$

Suppose $\sum_{l=1}^b \widehat{\theta}_l^{(j)}\phi_l^{(j)}(\boldsymbol{x}) \neq 0$. Then the LSLDG solution can be expressed as

$$\widehat{g}^{(j)}(\boldsymbol{x}) = \sum_{l=1}^{b} \widehat{\theta}_l^{(j)}\psi_l^{(j)}(\boldsymbol{x})$$
$$= \frac{1}{\sigma^{(j)2}}\sum_{l=1}^{b} \widehat{\theta}_l^{(j)}(c_l^{(j)} - x^{(j)})\phi_l^{(j)}(\boldsymbol{x}) = \widehat{\epsilon}^{(j)}(\boldsymbol{x})\widehat{m}^{(j)}(\boldsymbol{x}),$$

where $\widehat{\epsilon}^{(j)}(\boldsymbol{x}) = \frac{1}{\sigma^{(j)2}}\sum_{l=1}^b \widehat{\theta}_l^{(j)}\phi_l^{(j)}(\boldsymbol{x})$ and $\widehat{m}^{(j)}(\boldsymbol{x})$ is the j-th element of the mean shift vector defined as

$$\widehat{m}^{(j)}(\boldsymbol{x}) := \frac{\sum_{l=1}^b \widehat{\theta}_l^{(j)} c_l^{(j)} \phi_l^{(j)}(\boldsymbol{x})}{\sum_{l'=1}^b \widehat{\theta}_{l'}^{(j)}\phi_{l'}^{(j)}(\boldsymbol{x})} - x^{(j)}.$$

Then, a necessary condition for local maximum $\partial_j \widehat{p}(\boldsymbol{x}) = 0$ implies $\widehat{m}^{(j)}(\boldsymbol{x}) = 0$, which yields $x^{(j)} \leftarrow x^{(j)} + \widehat{m}^{(j)}(\boldsymbol{x})$. This update formula can be regarded as a weighted variant of the original mean shift algorithm (Cheng, 1995), and it is reduced to the original mean shift algorithm when $b = n$ and $\theta_l^{(j)} = 1/n$.

The LSLDG clustering algorithm was demonstrated to work well in experiments (Sasaki et al., 2014).

3 Proposed Method

In this section, we propose to combine the Riemannian extension of the mean shift algorithm and the LSLDG algorithm, and develop a novel clustering algorithm for Riemannian manifolds. Below, we consider data points $\{\boldsymbol{X}_i\}_{i=1}^n$ on a Riemannian manifold $(\mathcal{M}^m, H)$ of dimension $m(\leq d)$ embedded in $\mathbb{R}^d$ with Riemannian metric H.

Direct Log-Density-Gradient Estimation on Riemannian Manifolds: If LSLDG is naively applied to data on Riemannian manifolds, the estimated gradient vector does not necessarily lie on the tangent space. To prevent this problem, we propose to use the *common* parameter vector for all dimensions with basis functions confined in the tangent space. More specifically, let the true log-density gradient vector be $\boldsymbol{g}(\boldsymbol{X}) := \nabla \log p(\boldsymbol{X}) \in T_{\boldsymbol{X}}\mathcal{M}^m$, where ∇ denotes the Riemannian gradient. We model $\boldsymbol{g}(\boldsymbol{X})$ by $\widetilde{\boldsymbol{g}}(\boldsymbol{X}) = \sum_{l=1}^b \theta_l \boldsymbol{\psi}_l(\boldsymbol{X}) \in T_{\boldsymbol{X}}\mathcal{M}^m$, where θ_l is the common parameter, b is the number of the parameters, and $\boldsymbol{\psi}_l(\boldsymbol{X})$ is the vector of basis functions which we assume to be on the tangent space $T_{\boldsymbol{X}}\mathcal{M}^m$.

This common-parameter model is fitted to the true log-density gradient $\boldsymbol{g}(\boldsymbol{X})$ under the squared loss *on a manifold*:

$$J(\widetilde{g}(\boldsymbol{X})) := \int_{\mathcal{M}^m} \|\widetilde{\boldsymbol{g}}(\boldsymbol{X}) - \boldsymbol{g}(\boldsymbol{X})\|_H^2 p(\boldsymbol{X}) \mathrm{d}\mathbf{vol}_{\boldsymbol{X}} - \bar{C}$$
$$= \int_{\mathcal{M}^m} \|\widetilde{\boldsymbol{g}}(\boldsymbol{X})\|_H^2 p(\boldsymbol{X}) \mathrm{d}\mathbf{vol}_{\boldsymbol{X}}$$
$$- 2\int_{\mathcal{M}^m} \langle \widetilde{\boldsymbol{g}}(\boldsymbol{X}), \boldsymbol{g}(\boldsymbol{X})\rangle_H p(\boldsymbol{X}) \mathrm{d}\mathbf{vol}_{\boldsymbol{X}}$$
$$= \int_{\mathcal{M}^m} \|\widetilde{\boldsymbol{g}}(\boldsymbol{X})\|_H^2 p(\boldsymbol{X}) \mathrm{d}\mathbf{vol}_{\boldsymbol{X}}$$
$$- 2\int_{\mathcal{M}^m} \langle \widetilde{\boldsymbol{g}}(\boldsymbol{X}), \nabla p(\boldsymbol{X})\rangle_H \mathrm{d}\mathbf{vol}_{\boldsymbol{X}}, \quad (6)$$

where $\bar{C} = \int_{\mathcal{M}^m} \|\boldsymbol{g}(\boldsymbol{X})\|_H^2 p(\boldsymbol{X}) \mathrm{d}\mathbf{vol}_{\boldsymbol{X}}$, $\|\cdot\|_H^2 = \langle \cdot, \cdot \rangle_H$ denotes the inner product operator, $\mathrm{d}\mathbf{vol}_{\boldsymbol{X}}$ denotes a *volume element* of a Riemannian manifold (Petersen, 2006), and the last equality follows from the relation $\boldsymbol{g}(\boldsymbol{X}) = \nabla \log p(\boldsymbol{X}) = \nabla p(\boldsymbol{X}) / p(\boldsymbol{X})$. Applying the "*integration by parts*" formula for manifolds without boundary (Lee, 2012) to the second term in Eq.(6), we obtain

$$\int_{\mathcal{M}^m} \langle \widetilde{\boldsymbol{g}}(\boldsymbol{X}), \nabla p(\boldsymbol{X}) \rangle_H \mathrm{d}\mathbf{vol}_{\boldsymbol{X}} = -\int_{\mathcal{M}^m} p(\boldsymbol{X}) \mathrm{div} \widetilde{\boldsymbol{g}}(\boldsymbol{X}) \mathrm{d}\mathbf{vol}_{\boldsymbol{X}},$$

where "div" denotes the *divergence* (Petersen, 2006). Approximating the expectation over $p(\boldsymbol{X})$ by the average of samples $\{\boldsymbol{X}_i\}_{i=1}^n$ and adding the ℓ_2-regularization term, the following optimization problem is obtained:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^b}{\mathrm{argmin}} \left[\boldsymbol{\theta}^\top \widehat{\boldsymbol{G}} \boldsymbol{\theta} + 2 \boldsymbol{\theta}^\top \widehat{\boldsymbol{h}} + \lambda \boldsymbol{\theta}^\top \boldsymbol{\theta} \right],$$

where, for $l, l' = 1, \ldots, b$,

$$\widehat{G}_{l,l'} = \frac{1}{n} \sum_{i=1}^n \langle \boldsymbol{\psi}_l(\boldsymbol{X}_i), \boldsymbol{\psi}_{l'}(\boldsymbol{X}_i) \rangle_H, \tag{7}$$

$$\widehat{h}_l = \frac{1}{n} \sum_{i=1}^n \mathrm{div}(\boldsymbol{\psi}_l(\boldsymbol{X}_i)), \tag{8}$$

and $\lambda \geq 0$ is the regularization parameter. The optimal solution $\widehat{\boldsymbol{\theta}}$ can be obtained analytically as

$$\widehat{\boldsymbol{\theta}} = -(\widehat{\boldsymbol{G}} + \lambda \boldsymbol{I}_b)^{-1} \widehat{\boldsymbol{h}}.$$

All hyper-parameters such as the regularization parameter λ and basis parameters in $\boldsymbol{\psi}_l(\boldsymbol{X})$ can be systematically chosen by cross-validation with respect to the squared error criterion J. Finally, the log-density gradient estimator is given by $\widehat{\boldsymbol{g}}(\boldsymbol{X}) = \sum_{l=1}^b \widehat{\theta}_l \boldsymbol{\psi}_l(\boldsymbol{X})$.

We call this method *Riemannian LSLDG* (R-LSLDG).

Mode-Seeking on Riemannian Manifolds: Based on the estimated log-density gradient $\widehat{\boldsymbol{g}}(\boldsymbol{X})$, we propose a mode-seeking algorithm on Riemannian manifolds.

Let $\phi_l(\boldsymbol{X}) = \exp\left(-\frac{\delta(\boldsymbol{X}, \boldsymbol{C}_l)^2}{2\sigma^2}\right)$, where Gaussian centers $\{\boldsymbol{C}_l\}_{l=1}^b$ are randomly chosen from data samples $\{\boldsymbol{X}_i\}_{i=1}^n$ without overlap. In the same way as the original LSLDG clustering, we use its gradient as basis function vector $\boldsymbol{\psi}_l(\boldsymbol{X})$:

$$\boldsymbol{\psi}_l(\boldsymbol{X}) = -\frac{\left[\nabla \delta(\boldsymbol{X}, \boldsymbol{C}_l)^2\right] \phi_l(\boldsymbol{X})}{2\sigma^2} = \frac{[\log_{\boldsymbol{X}} \boldsymbol{C}_l] \, \phi_l(\boldsymbol{X})}{\sigma^2},$$

where we used Eq.(4).

Under the assumption that $\sum_{l=1}^b \widehat{\theta}_l \phi_l(\boldsymbol{X}) \neq 0$ analogously to the original mean shift algorithm, the mean shift vector is given as

$$\widehat{\boldsymbol{M}}(\boldsymbol{X}) = \frac{\sum_{l=1}^b \widehat{\theta}_l [\log_{\boldsymbol{X}} \boldsymbol{C}_l] \phi_l(\boldsymbol{X})}{\sum_{l'=1}^b \widehat{\theta}_{l'} \phi_{l'}(\boldsymbol{X})},$$

which always belongs to the tangent space $T_{\boldsymbol{X}} \mathcal{M}^m$. Then it is projected back to the manifold by the exponential map $\boldsymbol{X} \leftarrow \exp_{\boldsymbol{X}} \widehat{\boldsymbol{M}}(\boldsymbol{X})$. After repeating this update for all samples $\{\boldsymbol{X}_i\}_{i=1}^n$ until convergence, samples which converge to the same mode are given the same cluster label. We call this clustering method *R-LSLDG clustering* (R-LSLDGC).

Note that the common parameter formulation of LSLDG can be regarded as the limit of the multi-task LSLDG method (Yamane et al., 2016).

For further improvement, we can use a different bandwidth for each Gaussian center $\boldsymbol{C}_l$ similarly to Comaniciu et al. (2001). Specifically, we use a basis function

$$\bar{\psi}_l(\boldsymbol{X}) = \frac{1}{\sigma_l^2} [\log_{\boldsymbol{X}} \boldsymbol{C}_l] \bar{\phi}_l(\boldsymbol{X}),$$

where $\bar{\phi}_l(\boldsymbol{X}) = \exp(-\delta(\boldsymbol{X}, \boldsymbol{C}_l)^2 / (2\sigma_l^2))$. The mean shift vector is given by

$$\widehat{\boldsymbol{M}}(\boldsymbol{X}) = \frac{\sum_{l=1}^b \frac{\widehat{\theta}_l}{\sigma_l^2} [\log_{\boldsymbol{X}} \boldsymbol{C}_l] \bar{\phi}_l(\boldsymbol{X})}{\sum_{l'=1}^b \frac{\widehat{\theta}_{l'}}{\sigma_{l'}^2} \bar{\phi}_{l'}(\boldsymbol{X})},$$

where $\widehat{\theta}_l$ is learned by a basis function $\bar{\psi}_l(\boldsymbol{X})$.

Grassmann Manifold: In the experiments in the next section, we use the *Grassmann manifold* (Edelman et al., 1998) as an example of Riemannian manifolds. The Grassmann manifold $\mathcal{G}_{d_1,d_2}$ is the set of d_2-dimensional linear subspace in $\mathbb{R}^{d_1}$ for $d_2 \leq d_1$:

$$\mathcal{G}_{d_1,d_2} := \{\mathrm{span}(\boldsymbol{X}) \mid \boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}_{d_2}, \boldsymbol{X} \in \mathbb{R}^{d_1 \times d_2}\},$$

where $\mathrm{span}(\boldsymbol{X})$ denotes the subspace spanned by the columns of $\boldsymbol{X}$. Denoting by $T_{\boldsymbol{X}} \mathcal{G}_{d_1,d_2}$ the tangent space on the Grassmann manifold $\mathcal{G}_{d_1,d_2}$ at location $\boldsymbol{X} \in \mathcal{G}_{d_1,d_2}$, the canonical metric $\langle \cdot, \cdot \rangle_H$ for the Grassmann manifold is equivalent to the Euclidean metric $\langle \cdot, \cdot \rangle$ (Edelman et al., 1998). Thus, $\langle \boldsymbol{W}, \boldsymbol{Z} \rangle_H = \mathrm{tr}(\boldsymbol{W}^\top \boldsymbol{Z})$ holds for any $\boldsymbol{W}, \boldsymbol{Z} \in T_{\boldsymbol{X}} \mathcal{G}_{d_1,d_2}$, where $\mathrm{tr}(\boldsymbol{A}) = \sum_{i=1}^d \boldsymbol{A}_{i,i}$ for a square matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$. The exponential map for $\boldsymbol{Z} \in T_{\boldsymbol{X}} \mathcal{G}_{d_1,d_2}$ is given by

$$\exp_{\boldsymbol{X}} \boldsymbol{Z} = (\boldsymbol{X} \boldsymbol{V} \cos \boldsymbol{\Sigma} + \boldsymbol{U} \sin \boldsymbol{\Sigma}) \boldsymbol{V}^\top,$$

where $\boldsymbol{U}, \boldsymbol{\Sigma}$, and $\boldsymbol{V}$ come from the compact singular value decomposition of $\boldsymbol{Z}$, i.e., $\boldsymbol{Z} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^\top$, and "cos" and "sin" for matrix $\boldsymbol{\Sigma}$ act element-by-element along the diagonal of $\boldsymbol{\Sigma}$. The logarithm map for $\boldsymbol{Y} \in \mathcal{G}_{d_1,d_2}$ is given by

$$\log_{\boldsymbol{X}} \boldsymbol{Y} = (\boldsymbol{I}_{d_1} - \boldsymbol{X} \boldsymbol{X}^\top) \boldsymbol{Y} \boldsymbol{Y}^\top \boldsymbol{X}.$$

The square geodesic distance between two points on the Grassmann manifold is given by

$$\delta(\boldsymbol{X},\boldsymbol{Y})^2 = d_2 - \mathrm{tr}(\boldsymbol{Y}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{Y}).$$

Thus, we have

$$\nabla\delta(\boldsymbol{X},\boldsymbol{Y})^2 = -2(\boldsymbol{I}_{d_1} - \boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{Y}\boldsymbol{Y}^\top\boldsymbol{X},$$

where we used the transformation from the partial derivative $\frac{\partial}{\partial \boldsymbol{X}} f(\boldsymbol{X})$ of a function $f(\boldsymbol{X})$ to the manifold gradient $\nabla f(\boldsymbol{X})$: $\nabla f(\boldsymbol{X}) = (\boldsymbol{I}_{d_1} - \boldsymbol{X}\boldsymbol{X}^\top)\frac{\partial}{\partial \boldsymbol{X}} f(\boldsymbol{X})$. Then, $\widehat{G}_{l,l'}$ and $\widehat{h}_l$ defined by Eqs.(7) and (8) can be computed as

$$\begin{aligned}
\widehat{G}_{l,l'} &= \frac{1}{n\sigma_l^2\sigma_{l'}^2}\sum_{i=1}^{n}\sum_{j=1}^{d}[\boldsymbol{F}(\boldsymbol{X}_i,\boldsymbol{C}_l)]^{(j)}[\boldsymbol{F}(\boldsymbol{X}_i,\boldsymbol{C}_{l'})]^{(j)} \\
&\quad \times \phi_l(\boldsymbol{X}_i)\phi_{l'}(\boldsymbol{X}_i), \\
\widehat{h}_l &= \frac{1}{n\sigma_l^2}\sum_{i=1}^{n}\sum_{j=1}^{d}\left[(\boldsymbol{I}_{d_1} - \boldsymbol{X}_i\boldsymbol{X}_i^\top)\frac{\partial \boldsymbol{F}(\boldsymbol{X}_i,\boldsymbol{C}_l)}{\partial \boldsymbol{X}^{(j)}}\right]^{(j)}\phi_l(\boldsymbol{X}_i) \\
&\quad + \frac{1}{n\sigma_l^4}\sum_{i=1}^{n}\sum_{j=1}^{d}\left[[\boldsymbol{F}(\boldsymbol{X}_i,\boldsymbol{C}_l)]^{(j)}\right]^2\phi_l(\boldsymbol{X}_i), \qquad (9)
\end{aligned}$$

where $[\,\cdot\,]^{(j)}$ denotes the j-th element of the vectorization of the matrix, $d = d_1 d_2$, $\boldsymbol{F}(\boldsymbol{X},\boldsymbol{C}) = (\boldsymbol{I}_{d_1} - \boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{C}\boldsymbol{C}^\top\boldsymbol{X}$. Details for the derivation of Eq.(9) are deferred to the supplementary material.

4 Experiments

In this section, we experimentally compare the performance of the proposed R-LSLDG clustering (R-LSLDGC) algorithm with the original mean shift (MS), the Riemannian mean shift (R-MS), the LSLDG clustering (LSLDGC), and *spectral clustering* (SC) (Ng et al., 2002) in terms of the *adjusted Rand index* (ARI) (Hubert and Arabie, 1985), which takes the maximum value 1 when the obtained clustering solution perfectly matches the ground truth.

SC requires the number of clusters to be fixed in advance. For this reason, we provide the true number of clusters only to SC. The similarity between samples used in SC is defined as $\exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/(2\tau^2))$, where τ is the median of $\{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|\}_{i,j=1}^n$.

The number of basis functions in LSLDGC and R-LSLDGC is set at $b = \min(100, n)$. The bandwidth in all algorithms and the regularization parameter in LSLDGC and R-LSLDGC are chosen by 5-fold cross-validation from the 8 candidates values $\{10^{-3},\ldots,10\}$ at the regular interval in logarithmic scale.

All experiments were carried out using a PC equipped with two 2.60GHz Intel® Xeon® E5-2640 v3 CPUs.

Toy Data: Let $\{\boldsymbol{X}_i \in \mathbb{R}^{d_1\times d_2}\}_{i=1}^n$ be samples containing 3 clusters on a Grassmann manifold. For $d_1 = 3,\ldots,7$, $d_2 = 2$, and $n = 150$, each sample $\boldsymbol{X}_i$ is generated as

$$\boldsymbol{X}_i = \left(\begin{array}{cc|c} \cos\tau_i & -\sin\tau_i & \\ \sin\tau_i & \cos\tau_i & \boldsymbol{O}_{2,d_1-2} \\ \hline & \boldsymbol{O}_{d_1-2,2} & \boldsymbol{I}_{d_1-2} \end{array}\right)\boldsymbol{S}\begin{pmatrix}\cos\eta_i & -\sin\eta_i \\ \sin\eta_i & \cos\eta_i\end{pmatrix}, \quad (10)$$

where $\boldsymbol{S}$ is a randomly generated element on a Grassmann manifold $\mathcal{G}_{d_1,d_2}$ and $\boldsymbol{O}_{d,d'}$ is the $d\times d'$ null matrix. For $N(\mu,\sigma^2)$ being the normal distribution with mean μ and variance σ^2, τ_i and η_i are generated as

$$\begin{aligned}
\tau_i &\sim \begin{cases} N\left(0, \frac{\pi^2}{15^2}\right) & \text{for } i = 1,\ldots,\frac{n}{3}, \\ N\left(\frac{2\pi}{3}, \frac{\pi^2}{15^2}\right) & \text{for } i = \frac{n}{3}+1,\ldots,\frac{2n}{3}, \\ N\left(\frac{4\pi}{3}, \frac{\pi^2}{15^2}\right) & \text{for } i = \frac{2n}{3}+1,\ldots,n, \end{cases} \\
\eta_i &\sim N\left(0,\gamma^2\right),
\end{aligned}$$

where $\gamma = 0, \pi/8, \pi/4, \pi/2$ controls the variance of the angle of right rotation matrix. Note that when γ is zero, the right rotation matrix is reduced to the identity matrix. In Eq.(10), the left rotation matrix is rotation of the subspace for generating 3 clusters, while the right rotation matrix is rotation within the subspace. As plotted in Figure 3, the larger γ collapses the cluster structure on the Euclidean space but not on the Grassmann manifold.

The ARI values are summarized in Table 1, showing that R-LSLDGC significantly outperforms other methods when $\gamma = \pi/4, \pi/2$, while SC tends to perform very well when $\gamma = 0, \pi/8$. However, we should note that this comparison is not completely fair since SC is provided with the true number of clusters, while other methods estimate the number of clusters from data.

Compared with the plain LSLDGC, R-LSLDGC performs better for large γ, thanks to the geometry-aware formulation. On the other hand, the plain LSLDGC tends to outperform R-LSLDGC when $\gamma = 0$. This is caused by the difference in modeling: R-LSLDGC adopts the common-parameter formulation and thus only a single model is used for jointly estimating the gradient of all dimensions. In contrast, the plain LSLDGC adopts the coordinate-wise formulation, i.e., the gradient along each dimension is estimated separately. Due to this high flexibility, when $\gamma = 0$ (i.e., no manifold distortion is introduced), the plain LSLDGC sometimes performs better than R-LSLDGC.

R-MS tends to outperform the plain MS and plain LSLDGC when $\gamma = \pi/2$, thanks to the geometry-aware formulation. However, its performance degrades as the dimension d_1 increases. In contrast, R-LSLDGC performs reliably even for large d_1, which substantiates

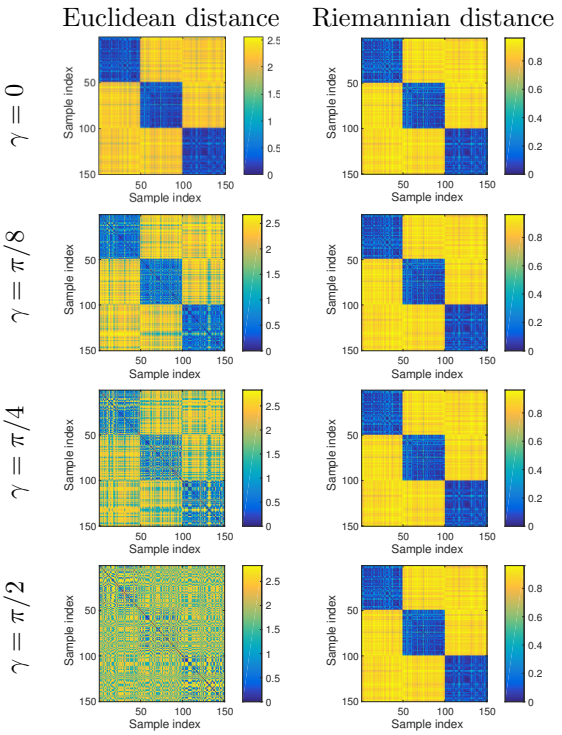


Figure 3: Distance matrices with the Euclidean and Riemannian distances when $d_1 = 3$, $d_2 = 2$, and $n = 150$. A larger γ collapses the cluster structure on the Euclidean space, while γ does not influence the cluster structure on the Grassmann manifold.

the usefulness of direct gradient estimation without going through kernel density estimation on Riemannian manifolds.

Figure 4 plots the computation time of each method for different d_1 averaged over $\gamma = 0, \pi/8, \pi/4, \pi/2$. The graph shows that SC, MS, and LSLDGC are quite fast, taking less than a few seconds. On the other hand, R-MS and R-LSLDGC take around 10 seconds, due to relatively heavy calculation of the logarithm map. This is the price we have to pay for better accuracy.

Image Clustering: The MNIST data set (Lecun et al., 1998) contains $0, \ldots, 9$ handwritten digits images. The images are down-sampled to 7×7 pixels from its original size 28×28. Following the experimental setup in Wang et al. (2014) partially, 20 images are drawn from one digit class, and these images are vectorized and concatenated to form a matrix of size $7^2 \times 20 = 980$. Singular value decomposition (SVD) $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is then applied to the matrix, and the top 3 left singular vectors are used as a sample on Grassmann manifold $\mathcal{G}_{49,3}$. Three digit classes are systematically selected to make the 3-clusters data sets of size $n = 150$, and we draw 50 samples from each digit class.

The ARI values are reported in Table 2, showing that the proposed R-LSLDGC outperforms other methods.

Table 1: The average and standard error of clustering accuracy measured by ARI (larger is better) over 20 runs for toy data with $d_2 = 2$. Bold face denotes the best and comparable methods in terms of the mean ARI according to the t-test at the significance level 5%.

γ	d_1	MS	LSLDGC	SC	R-MS	R-LSLDGC
0	3	.12 (.01)	.82 (.03)	**1.00 (.00)**	.51 (.06)	.81 (.04)
	4	.12 (.01)	**.88 (.03)**	**.94 (.04)**	.28 (.05)	.78 (.04)
	5	.12 (.01)	.89 (.02)	**1.00 (.00)**	.17 (.01)	.77 (.05)
	6	.12 (.01)	.90 (.03)	**1.00 (.00)**	.22 (.03)	.81 (.03)
	7	.09 (.01)	**.92 (.02)**	**.97 (.03)**	.18 (.01)	.85 (.03)
$\frac{\pi}{8}$	3	.10 (.01)	.62 (.03)	**.96 (.01)**	.51 (.06)	.81 (.04)
	4	.08 (.00)	.69 (.04)	**.95 (.03)**	.28 (.05)	.78 (.04)
	5	.08 (.01)	.67 (.05)	**.85 (.06)**	.17 (.01)	**.77 (.05)**
	6	.09 (.01)	.61 (.05)	**.92 (.05)**	.22 (.03)	.81 (.03)
	7	.09 (.01)	.66 (.04)	**.87 (.05)**	.18 (.01)	**.85 (.03)**
$\frac{\pi}{4}$	3	.19 (.04)	.34 (.03)	.50 (.04)	.51 (.06)	**.81 (.04)**
	4	.13 (.03)	.36 (.02)	.56 (.04)	.28 (.05)	**.78 (.04)**
	5	.05 (.02)	.37 (.03)	.49 (.05)	.17 (.01)	**.77 (.05)**
	6	.04 (.00)	.31 (.03)	.35 (.04)	.22 (.03)	**.81 (.03)**
	7	.04 (.00)	.36 (.03)	.44 (.04)	.18 (.01)	**.85 (.03)**
$\frac{\pi}{2}$	3	.13 (.02)	.21 (.02)	.09 (.01)	.51 (.06)	**.81 (.04)**
	4	.10 (.02)	.21 (.02)	.08 (.01)	.28 (.05)	**.78 (.04)**
	5	.02 (.00)	.15 (.02)	.09 (.01)	.17 (.01)	**.77 (.05)**
	6	.02 (.00)	.14 (.03)	.11 (.01)	.22 (.03)	**.81 (.03)**
	7	.02 (.00)	.17 (.02)	.09 (.01)	.18 (.01)	**.85 (.03)**

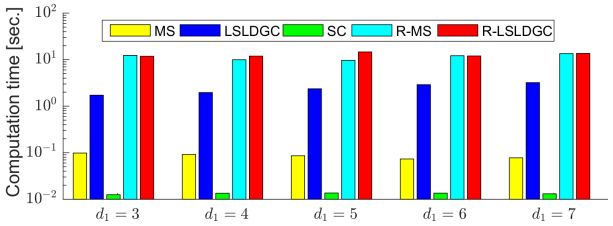


Figure 4: The average computation time of each method over all γ on toy data.

Figure 5 plots the average computation time over 20 runs. R-LSLDGC takes more than 100 seconds, but its computation time is still comparable to R-MS.

Motion Segmentation: The Hopkins 155 data set (Tron and Vidal, 2007) contains feature vectors automatically extracted from motions sequences of frame length F (see examples of the sequences in Figure 6). Under the planer scenes assumption, trajectories $\{\boldsymbol{a}_i \in \mathbb{R}^{2F}\}_{i=1}^n$ from the same motion lies on a 3-dimensional subspace of $\mathbb{R}^{2F}$, i.e., Grassmann manifold $\mathcal{G}_{2F,3}$ (Kanatani, 2002; Subbarao and Meer, 2009). We draw 3 trajectories $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, and $\boldsymbol{a}_3$ from the same motion, and then choose the top 3 left singular vectors from the matrix $[\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3]$ by applying SVD. Then we create data sets of size $n = 100$ or $n = 150$ by drawing 50 samples from each motion.

The ARI values are summarized in Table 3, showing that overall R-LSLDGC achieves higher clustering performance than other methods. Figure 7 plots the average computation time over 20 runs. On the whole,

Table 2: The average and standard error of clustering accuracy measured by ARI (larger is better) over 20 runs for the MNIST data set. Three clusters, $c_1 = 0$, $c_2 = 1$, and $c_3 = \{2, 3, \ldots, 9\}$ are picked to construct the clustering task. Bold face denotes the best and comparable methods in terms of the mean ARI according to the t-test at the significance level 5%.

c_3	MS	LSLDGC	SC	R-MS	R-LSLDGC
2	.00 (.00)	.08 (.02)	.13 (.02)	.12 (.01)	**.45 (.02)**
3	.00 (.00)	.06 (.02)	.16 (.02)	.10 (.01)	**.41 (.02)**
4	.00 (.00)	.06 (.02)	.16 (.02)	.12 (.01)	**.39 (.02)**
5	.00 (.00)	.06 (.02)	.16 (.01)	.08 (.02)	**.48 (.02)**
6	.00 (.00)	.06 (.02)	.11 (.02)	.03 (.01)	**.47 (.02)**
7	.00 (.00)	.06 (.02)	.16 (.02)	.00 (.00)	**.42 (.02)**
8	.00 (.00)	.08 (.02)	.11 (.01)	.12 (.01)	**.35 (.02)**
9	.00 (.00)	.06 (.02)	.17 (.02)	.05 (.01)	**.42 (.02)**

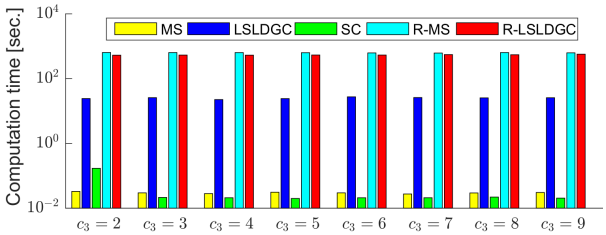


Figure 5: The average computation time of each method over 20 runs for the MNIST data set.

(a) arm (2, 30, 77) (b) articulated (3, 31, 150) (c) cars1 (2, 20, 307)

(d) cars3 (3, 20, 548) (e) cars5 (3, 374, 391) (f) cars7 (2, 25, 502)

(g) cars9 (3, 61, 220) (h) head (2, 60, 99) (i) people1 (2, 41, 504)

(j) three-cars (3, 15, 173) (k) truck1 (2, 30, 188) (l) truck2 (2, 22, 331)

Figure 6: Examples of sequences (at time $t = 0$) from the Hopkins 155 data set. The number in parentheses denotes (#Motions, #Frames, #Trajectories).

Table 3: The average and standard error of clustering accuracy measured by ARI (larger is better) over 20 runs on each sequence from the Hopkins 155 data set. The sequence names can be found in Figure 6. Bold face denotes the best and comparable methods in terms of the mean ARI according to the t-test at the significance level 5%.

Seq.	MS	LSLDGC	SC	R-MS	R-LSLDGC
(a)	.05 (.02)	.09 (.01)	.12 (.02)	.26 (.02)	**.65 (.04)**
(b)	.03 (.02)	.38 (.01)	.15 (.02)	.76 (.01)	**.81 (.01)**
(c)	.18 (.03)	.08 (.01)	.36 (.04)	.70 (.02)	**.75 (.02)**
(d)	.21 (.04)	.27 (.02)	.36 (.04)	.72 (.01)	**.80 (.01)**
(e)	.18 (.03)	.03 (.02)	.25 (.03)	.42 (.01)	**.45 (.01)**
(f)	.12 (.02)	.03 (.01)	.28 (.03)	**.51 (.01)**	.45 (.01)
(g)	.08 (.02)	.06 (.01)	.16 (.02)	.63 (.01)	**.70 (.01)**
(h)	.02 (.01)	.01 (.00)	.03 (.01)	.01 (.00)	**.36 (.02)**
(i)	.43 (.04)	.34 (.02)	.48 (.03)	.50 (.04)	**.59 (.01)**
(j)	.11 (.02)	.04 (.01)	.28 (.04)	.65 (.05)	**.76 (.01)**
(k)	.06 (.02)	.02 (.01)	.28 (.03)	**.76 (.04)**	**.83 (.02)**
(l)	.04 (.01)	.06 (.02)	.28 (.03)	.59 (.02)	**.64 (.01)**

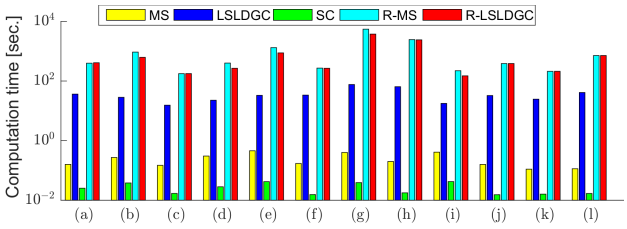


Figure 7: The average computation time of each method over 20 runs on each sequence from the Hopkins 155 data set. The sequence names can be found in Figure 6.

R-LSLDGC is not a computationally efficient method, but its computational time is still comparable to R-MS and it performs much better than R-MS.

5 Conclusions

Mean shift is a promising approach to mode-seeking clustering. In this paper, we extended the mean shift clustering algorithm so that Riemannian generalization and direct gradient estimation are both incorporated. Through experiments on Grassmann manifolds, we demonstrated the usefulness of the proposed method. In our future work, we will test the proposed method for other Riemannian manifolds such as Lie groups, the Stiefel manifold, and symmetric positive definite matrices. We will also investigate a computationally efficient approximation scheme for speedup.

Acknowledgements

We thank Prof. Kazumasa Kuwada and Mr. Ikko Yamane for fruitful discussion. MA was supported by KAKENHI 26280054, HS was supported by KAKENHI 15H06103, TS was supported by KAKENHI 15J09111, and MS was supported by KAKENHI 26280054.

References

B. Clarke, E. Fokoue, and H. H. Zhang. *Principles and Theory for Data Mining and Machine Learning*. Springer, 2009.

K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

M. A. Carreira-Perpinán. A review of mean-shift algorithms for clustering. Technical Report 1503.00687, arXiv, 2015.

J. Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and video segmentation by anisotropic kernel mean shift. In *European Conference on Computer Vision*, pages 238–249. Springer, 2004.

W. Tao, H. Jin, and Y. Zhang. Color image segmentation based on mean shift and normalized cuts. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(5):1382–1389, 2007.

D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 142–149, 2000.

R. T. Collins. Mean-shift blob tracking through scale space. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 234–240, 2003.

W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Gulf Professional Publishing, 2003.

O. Tuzel, R. Subbarao, and P. Meer. Simultaneous multiple 3D motion estimation via mode finding on Lie groups. In *Tenth IEEE International Conference on Computer Vision*, volume 1, pages 18–25, 2005.

R. Subbarao and P. Meer. Nonlinear mean shift for clustering over analytic manifolds. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1168–1175, 2006.

R. Subbarao and P. Meer. Nonlinear mean shift over Riemannian manifolds. *International Journal of Computer Vision*, 84(1):1–20, 2009.

H. E. Cetingul and R. Vidal. Intrinsic mean shift for clustering on Stiefel and Grassmann manifolds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1896–1902, 2009.

D. D. Cox. A penalty method for nonparametric estimation of the logarithmic derivative of a density function. *Annals of the Institute of Statistical Mathematics*, 37(1):271–288, 1985.

H. Sasaki, A. Hyvärinen, and M. Sugiyama. Clustering via mode seeking by direct estimation of the gradient of a log-density. In *Machine Learning and Knowledge Discovery in Databases Part III - European Conference*, pages 19–34, 2014.

B. Pelletier. Kernel density estimation on Riemannian manifolds. *Statistics & probability letters*, 73(3):297–304, 2005.

G. Arvanitidis, L. K. Hansen, and S. Hauberg. A locally adaptive normal distribution. In *Advances in Neural Information Processing Systems 29*, pages 4251–4259, 2016.

P. Petersen. *Riemannian Geometry*. Springer New York Inc., New York, NY, USA, 2 edition, 2006.

J. M. Lee. *Introduction to Smooth Manifolds*. Springer New York Inc., New York, NY, USA, 2012.

I. Yamane, H. Sasaki, and M. Sugiyama. Regularized multitask learning for multidimensional log-density gradient estimation. *Neural Computation*, 28(7): 1388–1410, 2016.

D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. In *Proceedings of Eighth IEEE International Conference on Computer Vision*, volume 1, pages 438–445, 2001.

A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856, 2002.

L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

B. Wang, Y. Hu, J. Gao, Y. Sun, and B. Yin. Low rank representation on Grassmann manifolds. In *Proceedings of 12th Asian Conference on Computer Vision*, pages 81–96, 2014.

R. Tron and R. Vidal. A benchmark for the comparison of 3-D motion segmentation algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

K. Kanatani. Evaluation and selection of models for motion segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 335–349. Springer, 2002.