



Embedding and function extension on directed graph

Saman Mousazadeh*, Israel Cohen

Department of Electrical Engineering, Technion-Israel Institute of Technology, Technion City, Haifa 32000, Israel



ARTICLE INFO

Article history:

Received 14 July 2014

Received in revised form

13 November 2014

Accepted 16 December 2014

Available online 24 December 2014

Keywords:

Directed graph

Asymmetric kernel

Function extension

ABSTRACT

In this paper, we propose a novel technique for finding the graph embedding and function extension for directed graphs. We assume that the data points are sampled from a manifold and the similarity between the points is given by an asymmetric kernel. We provide a graph embedding algorithm which is motivated by Laplacian type operator on manifold. We also introduce a Nyström type eigenfunctions extension which is used both for extending the embedding to new data points and to extend an empirical function on new data set. For extending the eigenfunctions to new points, we assume that only the distances of the new points from the labelled data are given. Simulation results demonstrate the performance of the proposed method in recovering the geometry of data and extending a function on new data points.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Recent advances in geometric based methods of data mining and machine learning lead to efficient algorithms for lots of applications such as dimensionality reduction, function extension, classification and clustering, just to name a few. Most of these methods are graph based techniques. Graphs offer an advantageous compromise between their simplicity, interpretability and their ability to express complex relationships between data points. The core idea in such algorithms is to construct a weighted graph on data points such that each vertex of the graph represents a data point, and a weighted edge, connecting two vertices to each other, represents the similarity between the two corresponding data points. In the context of networks (e.g., social networks), the data naturally lead themselves to graph modelling [1]. The graph based representation of data combined with Markov chain techniques exhibits extremely successful results. The main idea here is based on the fact that the eigenvectors of Markov matrices can be regarded as coordinates on the data set. Among vast

techniques incorporating Markov chain methods in data processing, kernel eigenmap methods have attracted much research attention recently. The algorithmic consequences of these methods are local linear embedding (LLE) [2], Laplacian eigenmaps [3], Hessian eigenmaps [4], local tangent space alignment [5] and diffusion maps [6].

In most of these kernel eigenmaps based methods, the similarity between points is given by a symmetric positive semi-definite kernel. In some practical applications the similarity between points is not necessarily symmetric. Typical examples are web information retrieval based on hyperlink structure, document classification based on citation graphs [7], web information retrieval based on hyperlink structure, and protein clustering based on pairwise alignment scores [8]. Some works have been done to deal with the ranking problem on link structure of the Web. Although much progress in the field, it is still a hard task to do general data analysis on directed graphs such as classification and clustering. Chen et al. [9] proposed an algorithm for embedding vertices on directed graphs to vector spaces. This algorithm explores the inherent pairwise relation between vertices of the directed graph by using transition probability and the stationary distribution of Markov random walks, and embeds the vertices into vector spaces preserving such relation optimally. Recently,

* Corresponding author.

E-mail addresses: smzadeh@tx.technion.ac.il (S. Mousazadeh), icohen@ee.technion.ac.il (I. Cohen).

Perrault-Joncas and Meilă [10] proposed an algorithm based on the analysis of Laplacian type operators and their continuous limit as generators of diffusions on a manifold. They modelled the observed graph as a sample from a manifold endowed with a vector field, and designed an algorithm that separates and recovers the features of this process: the geometry of the manifold, the data density, and the vector field. The most important shortcoming of these methods is not providing a straightforward procedure to extend the embedding to new points in case only the distances of the new point to the original data points are known, which encountered in applications. In [11], Coifman and Hirn introduced a simple procedure for the construction of a bi-stochastic kernel for an arbitrary data set that is derived from an asymmetric affinity function. The affinity function measures the similarity between points in test set and some reference set.

These geometric based algorithms have been applied in various signal processing applications. In a pioneering work, Shi and Malik [12] used spectral methods for image segmentation. Later many researchers have used geometric based methods in applications such as image clustering [13,14], image completion [15], speech enhancement in the presence of transient noise [16], voice activity detection in the presence of transient noise [17], linear and nonlinear independent component analysis [18,19], parametrization of linear systems [20], and single channel source localization [21]. Most if not all of these applications can be regarded as out of sample function extension, in which an empirical function is extended to unlabelled data. In these applications, usually a large amount of data is involved and the only way to perform a task like clustering, regression, or classification is to subsample the data set \bar{X} in order to reduce the size of the problem, process the new set X , and then extend the results to the original data \bar{X} . Coifman and Lafon [22] proposed a geometric harmonics procedure, inspired from the Nyström method, to perform this task. More specifically, they assumed that the similarity between the data points is given by a symmetric positive semi-definite kernel. Then it is shown that the eigenfunctions of the integration operator defined by this kernel form an orthogonal basis for the space of squared integrable functions defined on \bar{X} (i.e. $L^2(\bar{X})$). In order to extend a function defined on the set X to the data set \bar{X} , first the eigenfunctions computed on X are extended to the data set \bar{X} using the Nyström method. The function is then approximated as the weighted sum of these extended eigenfunctions. Kushnir et al. [23] and Singer et al. [19] introduced a method for parameterizing high dimensional data into its independent physical parameters, which enables the identification of the parameters and a supervised extension of the re-parametrization to new observations. In their work, a novel diffusion processes was used, utilizing only the small observed set, that approximates the isotropic diffusion on the parametric manifold. They utilized Nyström-type extension of the embedding of that small observed data-set to the embedding into the independent components on a much larger data-set.

In this paper, we propose a novel technique for embedding a directed graph to Euclidean space. We model the observed data as samples from a manifold where the similarity between the points is given by an asymmetric

kernel. This asymmetric kernel is modelled utilizing a vector field, and we design an algorithm that separates and recovers the geometry of the manifold, the data density, and the vector field. We further provide a simple Nyström extension procedure which allows us to extend both the embedding and the estimated vector field to new data points. More precisely, we adopt the method presented in [23] into the case when the kernel is asymmetric. The rest of this paper is organized as follows. In Section 2, we provide a model which can be used in directed graph modeling. We also introduce our results regarding the limit of Laplacian type operators and provide an algorithm for obtaining the embedding of a directed graph. We also propose a Nyström extension procedure for extending the embedding and the vector field to new data points. In Section 3 we provide some experimental results. We conclude the paper in Section 4.

2. Problem formulation, embedding and function extension

Let X be a set of n data points sampled according to a distribution $p = e^{-U}$ from an unknown smooth manifold $\mathcal{M} \subset \mathbb{R}^\ell$ with intrinsic dimension $d < \ell$. Let G be a directed graph with n nodes constructed from the data set X , where each nodes of the graph (e.g. the node i) corresponds to a point in the set X (e.g. $x_i \in X$). We assume that the edge weight K_{ij} between nodes i and j is given by a positive asymmetric similarity kernel $k_e(\cdot, \cdot)$ (i.e. $K_{ij} = k_e(x_i, x_j) \geq 0$). We also assume that the directional component of $k_e(\cdot, \cdot)$ is derived by a vector field \mathbf{r} on the manifold \mathcal{M} , which will be precisely defined shortly. This vector field \mathbf{r} is sufficient to characterize any directionality associated with a drift component and as it turns out, the component of \mathbf{r} normal to $\mathcal{M} \subset \mathbb{R}^\ell$ can also be used to characterize any source component, see [10] for further discussion. The problem is finding an embedding of G into \mathbb{R}^m ; $m \leq d$ which approximates the generative process geometry \mathcal{M} , the sampling distribution $p = e^{-U}$, and the directionality \mathbf{r} . This embedding needs to be consistent as sample size increases and the bandwidth of the kernel vanishes.

2.1. Anisotropic diffusion operator

Any kernel $k_e(x, y)$ can be decomposed into symmetric and anti-symmetric parts as follows:

$$k_e(x, y) = h_e(x, y) + a_e(x, y), \quad (1)$$

where $h_e(x, y) = h_e(y, x)$ is the symmetric component and $a_e(x, y) = -a_e(y, x)$ is the antisymmetric component of the kernel. As in [10], we assume that the symmetric and anti-symmetric parts can be written as

$$h_e(x, y) = \frac{h(\|x - y\|^2/\epsilon)}{\epsilon^{d/2}} \quad (2)$$

$$a_e(x, y) = \frac{\mathbf{r}(x, y)}{2} \cdot (y - x) \frac{h(\|x - y\|^2/\epsilon)}{\epsilon^{d/2}}, \quad (3)$$

respectively, where $\mathbf{r}(x, y) = \mathbf{r}(y, x)$ and $h \geq 0$ is an arbitrary exponentially decreasing function when $\|x - y\|$ converges to infinity.

In [10] an embedding algorithm has been proposed based on symmetrization of this kernel. More specifically, assuming that $k_\epsilon(x, y)$ is the asymmetric similarity kernel, the authors defined a symmetric kernel $s_\epsilon(x, y): \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}^+$ as follows:

$$s_\epsilon(x, y) = s_\epsilon(y, x) = \frac{k_\epsilon(x, y) + k_\epsilon(y, x)}{2}. \quad (4)$$

Then the diffusion map [6] has been applied to this symmetric kernel in order to find the graph embedding. The vector field \mathbf{r} and the sampling density p are then obtained by constructing a new diffusion operator using $k_\epsilon(\cdot, \cdot)$. Although this method succeeds in obtaining the embedding and estimating the sampling distribution and the vector field, the method suffers from a serious limitation. Suppose that for a set of data $X = \{x_1, x_2, \dots, x_n\}$ the embedding is found by performing the above mentioned procedure. Now we want to approximate the embedding of a new data point y using Nyström like eigenfunction extension. Since the kernel is asymmetric, we need to know both the distance (i.e. the affinity measure) of the new point y from all of the points in X (i.e. $k_\epsilon(y, x_i); 1 \leq i \leq n$) and the distances of all of the points in X from the new point y (i.e. $k_\epsilon(x_i, y); 1 \leq i \leq n$). In practical applications, one of these sets of distances might be unknown which restricts this method.

Here, we solve the above-mentioned problem by choosing a different symmetrization of the affinity kernel. Using the asymmetric kernel $k_\epsilon(x, y)$, we define a symmetric kernel $s_\epsilon(x, y): \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}^+$ as follows:

$$s_\epsilon(x, y) = s_\epsilon(y, x) = \int_{\mathcal{M}} k_\epsilon(x, t)k_\epsilon(y, t) dt. \quad (5)$$

Note that this kernel is positive semi-definite, i.e., for any $m \geq 1$ and any choice of real numbers $\alpha_1, \dots, \alpha_m$, and points x_1, \dots, x_m in \mathcal{M} , we have

$$\begin{aligned} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j s_\epsilon(x_i, x_j) &= \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j \int_{\mathcal{M}} k_\epsilon(x_i, t)k_\epsilon(x_j, t) dt \\ &= \int_{\mathcal{M}} \left(\sum_{i=1}^m \alpha_i k_\epsilon(x_i, t) \right)^2 dt \geq 0. \end{aligned} \quad (6)$$

These two kernels, i.e. the asymmetric kernel $k_\epsilon(x, y)$ and its symmetrized version $s_\epsilon(x, y)$, can be used to construct a family of Laplacian type operators suitable for graph embedding as in [24,25,6]. This family of operators is inspired by anisotropic diffusion operator introduced in [6] for embedding undirected graph. We define a family of diffusion operators as follows. Let $\alpha \in [0, 1]$ and

$$p_{k,\epsilon}(x) = \int_{\mathcal{M}} k_\epsilon(x, y)p(y) dy \quad (7)$$

$$p_{s,\epsilon}(x) = \int_{\mathcal{M}} s_\epsilon(x, y)p(y) dy. \quad (8)$$

We form a new kernel by

$$k_\epsilon^{(\alpha)}(x, y) = \frac{k_\epsilon(x, y)}{p_{k,\epsilon}^\alpha(x)p_{k,\epsilon}^\alpha(y)} \quad (9)$$

$$s_\epsilon^{(\alpha)}(x, y) = \frac{s_\epsilon(x, y)}{p_{s,\epsilon}^\alpha(x)p_{s,\epsilon}^\alpha(y)}. \quad (10)$$

Then, we apply the weighted graph Laplacian normalization to these kernels by setting the outdegree distributions of the newly defined kernels as

$$d_{k,\epsilon}(x) = \int_{\mathcal{M}} k_\epsilon^{(\alpha)}(x, y)p(y) dy \quad (11)$$

$$d_{s,\epsilon}(x) = \int_{\mathcal{M}} s_\epsilon^{(\alpha)}(x, y)p(y) dy, \quad (12)$$

and by defining the anisotropic transition kernel by

$$t_{k,\epsilon,\alpha}(x, y) = \frac{k_\epsilon^{(\alpha)}(x, y)}{d_{k,\epsilon}(x)} \quad (13)$$

$$t_{s,\epsilon,\alpha}(x, y) = \frac{s_\epsilon^{(\alpha)}(x, y)}{d_{s,\epsilon}(x)}. \quad (14)$$

Finally we define our diffusion operator as follows:

$$T_{\epsilon,\alpha}^{aa}f(x) = \int_{\mathcal{M}} t_{k,\epsilon,\alpha}(x, y)f(y)p(y) dy \quad (15)$$

$$T_{\epsilon,\alpha}^{ss}f(x) = \int_{\mathcal{M}} t_{s,\epsilon,\alpha}(x, y)f(y)p(y) dy, \quad (16)$$

where the superscripts a and s indicate which kernel (asymmetric $k_\epsilon(x, y)$ or symmetric $s_\epsilon(x, y)$) is used for obtaining the anisotropic kernel and the out-degree distribution. It is worth mentioning that there are in fact eight possible combinations of kernel and degree distribution. Since only the two above-mentioned diffusion operators are of particular interest in retrieving the generative process geometry \mathcal{M} , the sampling distribution $p = e^{-U}$, and the directionality \mathbf{r} , we concentrate on obtaining the infinitesimal generators of these diffusion operators as ϵ converges to 0.

2.2. Continuous limit of the diffusion operator

In order to find the infinitesimal generators of the diffusion operators we need the following two lemmas.

Lemma 1. *Let \mathcal{M} be a compact, closed, smooth manifold of dimension d and $k_\epsilon(x, y)$ be an asymmetric similarity kernel defined by Eqs. (1)–(3), then for any function $f \in C^3(\mathcal{M})$, the integral operator based on k_ϵ has the asymptotic expansion:*

$$\begin{aligned} \int_{\mathcal{M}} k_\epsilon(x, y)f(y) dy &= m_0f(x) \\ &+ \epsilon \frac{m_2}{2}(\omega(x)f(x) + \Delta f(x) + \mathbf{r} \cdot \nabla f(x) \\ &+ f(x)\nabla \cdot \mathbf{r}) + \mathcal{O}(\epsilon^2) \end{aligned} \quad (17)$$

and

$$\begin{aligned} \int_{\mathcal{M}} k_\epsilon(y, x)f(y) dy &= m_0f(x) + \epsilon \frac{m_2}{2}(\omega(x)f(x) \\ &+ \Delta f(x) - \mathbf{r} \cdot \nabla f(x) - f(x)\nabla \cdot \mathbf{r}) + \mathcal{O}(\epsilon^2), \end{aligned} \quad (18)$$

where $m_0 = \int_{\mathbb{R}^d} h(\|u\|^2) du$ and $m_2 = \int_{\mathbb{R}^d} u_1^2 h(\|u\|^2) du$, u_1 is the first element of a vector u and $\omega(x) = (2/m_2) \int_{u \in \mathbb{R}^d} (Q_2(u)h(\|u\|^2) + Q_4(u)h'(\|u\|^2)) du$.

Proof. See [Appendix A](#).

Lemma 2. Suppose the conditions of [Lemma 1](#) are satisfied, then

$$\begin{aligned} \int_{\mathcal{M}} s_{\epsilon}(x, y) f(y) dy &= \int_{\mathcal{M}} \int_{\mathcal{M}} k_{\epsilon}(x, t) k_{\epsilon}(y, t) dt f(y) dy \\ &= m_0^2 f(x) + \epsilon m_0 m_2 (\omega(x) f(x) + \Delta f(x)) + \mathcal{O}(\epsilon^2). \end{aligned} \quad (19)$$

Proof. Let

$$\begin{aligned} G_{\epsilon}(x) &= \int_{\mathcal{M}} \int_{\mathcal{M}} k_{\epsilon}(x, t) k_{\epsilon}(y, t) f(y) dy dt \\ &= \int_{\mathcal{M}} k_{\epsilon}(x, t) \left(\int_{\mathcal{M}} k_{\epsilon}(y, t) f(y) dy \right) dt. \end{aligned} \quad (20)$$

Using [Lemma 1](#) (Eq. (18)) we have

$$G_{\epsilon}(x) = \int_{\mathcal{M}} k_{\epsilon}(x, t) \left(m_0 f(t) + \epsilon \frac{m_2}{2} g(t) \right) dt + \mathcal{O}(\epsilon^2) \quad (21)$$

where

$$g(x) = \omega(x) f(x) + \Delta f(x) - \mathbf{r} \cdot \nabla f(x) - f(x) \nabla \cdot \mathbf{r}. \quad (22)$$

Now applying [Lemma 1](#) again to Eq. (21) we have

$$\begin{aligned} G_{\epsilon}(x) &= m_0 \left[m_0 f(x) + \epsilon \frac{m_2}{2} g(x) \right] + \epsilon \frac{m_2}{2} \\ &\quad \times \left[\omega(x) \left(m_0 f(x) + \epsilon \frac{m_2}{2} g(x) \right) + \Delta \left(m_0 f(x) + \epsilon \frac{m_2}{2} g(x) \right) \right. \\ &\quad \left. + \mathbf{r} \cdot \nabla \left(m_0 f(x) + \epsilon \frac{m_2}{2} g(x) \right) + \nabla \cdot \mathbf{r} (m_0 f(x) \right. \\ &\quad \left. + \epsilon \frac{m_2}{2} g(x) \right) \right] + \mathcal{O}(\epsilon^2). \end{aligned} \quad (23)$$

Discarding the second order terms we get

$$\begin{aligned} G_{\epsilon}(t) &= m_0^2 f(t) + \epsilon \frac{m_2 m_0}{2} g(t) + \epsilon \frac{m_2}{2} [\omega(x) m_0 f(t) + \Delta(m_0 f(t)) \\ &\quad + \mathbf{r} \cdot \nabla(m_0 f(t)) + \nabla \cdot \mathbf{r} (m_0 f(t))] + \mathcal{O}(\epsilon^2) \\ &= m_0^2 f(t) + \epsilon \frac{m_2 m_0}{2} [g(t) + \omega(x) f(x) + \Delta f(t) + \mathbf{r} \cdot \nabla f(t) \\ &\quad + \nabla \cdot \mathbf{r} f(t)] + \mathcal{O}(\epsilon^2). \end{aligned} \quad (24)$$

Using the definition of $g(x)$ in (22) we get the final result, i.e.

$$G_{\epsilon}(t) = m_0^2 f(t) + \epsilon m_2 m_0 [\omega(x) f(x) + \Delta f(t)] + \mathcal{O}(\epsilon^2). \quad \square \quad (25)$$

The scalar functions $\omega(x)$ was first obtained in [6] and corresponds to an interaction between the symmetric kernel $h_{\epsilon}(\cdot, \cdot)$ and the curvature of \mathcal{M} . [Lemma 1](#) gives a general fact about spectral embedding algorithms: in most cases, Laplacian operators confound the effects of spatial proximity, sampling density and directional flow due to the presence of the various terms above.

The next theorem represents the main result of this paper regarding the infinitesimal generator of the diffusions introduced before.

Theorem 3. Suppose that the kernel $k(x, y)$ is normalized such that $m_0 = 1$ and $m_2 = 2$. Let

$$L_{\epsilon, \alpha}^{aa} = \frac{I - T_{\epsilon, \alpha}^{aa}}{\epsilon} \quad (26)$$

$$L_{\epsilon, \alpha}^{ss} = \frac{I - T_{\epsilon, \alpha}^{ss}}{2\epsilon} \quad (27)$$

be the infinitesimal generator of the Markov chain, where $T_{\epsilon, \alpha}^{aa}$ and $T_{\epsilon, \alpha}^{ss}$ are defined in (15) and (16), respectively. Then

$$\lim_{\epsilon \rightarrow 0} L_{\epsilon, \alpha}^{aa} f = \Delta f - 2(1 - \alpha) \nabla U \cdot \nabla f + \mathbf{r} \cdot \nabla f \quad (28)$$

$$\lim_{\epsilon \rightarrow 0} L_{\epsilon, \alpha}^{ss} f = \Delta f - 2(1 - \alpha) \nabla U \cdot \nabla f \quad (29)$$

Proof. See [Appendix B](#).

In the next section, we show how this theorem can be used to recover the geometry of data along with the vector field.

2.3. Manifold learning, estimation of sampling density and the vector field

The geometry of the data can be fully recovered by computation of the Laplace–Beltrami operator on the manifold. In other words, the geometry of a Riemannian manifold is determined by the spectrum of Laplace–Beltrami operator. Additionally, since the eigenfunctions of this operator make a basis for all functions in $L^2(\mathcal{M})$, obtaining the eigenfunctions of this operator plays an important role in manifold learning. In order to fully learn the manifold \mathcal{M} , we also need to estimate the sampling density $p = e^{-U}$ and the vector field \mathbf{r} . As shown in [26,10] the eigenfunctions of the Laplace–Beltrami operator can be estimated by computing the eigenfunctions of $L_{\epsilon, 1}^{ss}$. Furthermore, the sampling density can be obtained by computing the left eigenfunctions of the $L_{\epsilon, 1}^{ss}$ corresponding to eigenvalue 1.

Since

$$\lim_{\epsilon \rightarrow 0} L_{\epsilon, \alpha}^{aa} f - L_{\epsilon, 1}^{ss} f = \mathbf{r} \cdot \nabla f, \quad (30)$$

the components of the vector field \mathbf{r} on the tangent space denoted by $\mathbf{r}_{||}$ can be recovered by

$$\mathbf{r}_{||} = \mathbf{r} \cdot \nabla \psi, \quad (31)$$

where ψ is any diffeomorphic embedding of \mathcal{M} . Because of the fact that the eigenfunctions of the Laplace–Beltrami operator are a diffeomorphic embedding of \mathcal{M} , we use the eigenfunctions of this operator in (31) to recover the vector field \mathbf{r} . The algorithmic consequence of this procedure for learning the manifold using a directed kernel is summarized in [Table 2](#).

2.4. Embedding and function extension on directed graph

In this section, we present a method for embedding and function extension on directed graph which essentially originates from Nyström eigenfunction extension. We adopt the method presented in [23] into the case when the kernel is asymmetric. More specifically, suppose that we have a data set $X \subset \mathcal{M}$ consisting of n data points. Let $\psi(X) \in \mathbb{R}^{n \times m}$ be the embedding of these points into \mathbb{R}^m obtained using [Algorithm 2](#). Furthermore, suppose that we are given a function $f(\cdot)$ whose value is known only on the set X . Now, assume that we are given another set of data points $\bar{X} \subset \mathcal{M}$ and the distances $\{k_{\epsilon}(\bar{x}, x); \bar{x} \in \bar{X}, x \in X\}$. The goal is

extending the embedding and finding the value of the function $f(\cdot)$ on this set $\bar{X} \subset \mathcal{M}$.

As shown earlier in (6), $s_\epsilon(x, y)$, the symmetrized version of the asymmetric kernel $k_\epsilon(x, y)$, is positive semi-definite. Using $s_\epsilon(x, y)$ we define an integral operator by

$$\mathbf{S}f(x) = \int_X s_\epsilon(x, y)f(y) dy. \quad (32)$$

It is trivial to show that this operator is self-adjoint and compact and hence has a discrete sequence of eigenvalues $\{\lambda_j\}$ (in nonincreasing order) and eigenvectors $\{\psi_j\}$ defined on X where

$$\lambda_j \psi_j(x) = \int_X s_\epsilon(x, y)\psi_j(y) dy, \quad (33)$$

and since the operator is positive semi-definite, $\lambda_j \geq 0; \forall j$. The fact that the kernel can be evaluated on the entire \mathcal{M} enables us to take any $\bar{x} \in \mathcal{M}$ in the right-hand side of this identity. This yields the following Nyström extension of ψ_j from X to \mathcal{M} , i.e.

$$\bar{\psi}_j(\bar{x}) = \frac{1}{\lambda_j} \int_X s_\epsilon(\bar{x}, y)\psi_j(y) dy; \quad \bar{x} \in \mathcal{M}, \quad (34)$$

where $\bar{\psi}_j(\bar{x})$ is the *extended* value of the eigenvector $\psi_j(\cdot)$ at the point \bar{x} . Note that since

$$s_\epsilon(\bar{x}, y) = \int_{\mathcal{M}} k_\epsilon(\bar{x}, t)k_\epsilon(y, t) dt, \quad (35)$$

this eigenfunction extension technique only needs the distance from the test point \bar{x} to the training points $y \in \mathcal{M}$. Hence, in cases where the distance from the training points to the test point \bar{x} is not available, this method outperforms the method presented in [10].

Our function extension algorithm relies on the fact that the Laplace–Beltrami operator is a positive semi-definite operator and therefore its eigenfunctions form a basis for the set of functions defined on the manifold. In particular, since the eigenfunctions $\psi_j(x); x \in X$ form a basis for the set of functions on X , any function $f(\cdot)$ on this set can be decomposed as

$$f(x) = \sum_{j=1}^{\infty} \langle \psi_j, f \rangle \psi_j(x); \quad x \in X, \quad (36)$$

hence we can define the Nyström extension of $f(\cdot)$ to the rest of \mathcal{M} to be

$$\bar{f}(\bar{x}) \triangleq \sum_{j=1}^{\delta} \langle \psi_j, f \rangle \bar{\psi}_j(\bar{x}); \quad \bar{x} \in \bar{X}. \quad (37)$$

where $\bar{\psi}_j(\bar{x})$ is defined in (34) and δ is chosen considering stability issues. More specifically, choosing the parameter δ is

application dependent which depends on both the manifold and the function to be extended. In practical cases this parameter can be chosen by cross validation. For further discussion on this issue, interested readers are referred to [27]. It is worth mentioning that the vector field \mathbf{r} which is computed on the set X can also be extended to \mathcal{M} by treating each element of \mathbf{r} as a function defined on the set X and using (37). The overall algorithm for embedding and function extension is summarized in Table 1, where \mathbf{I} and $\mathbf{1}$ are respectively the identity matrix and a vector of ones with appropriate sizes.

3. Experimental results

In this section, we examine the performance of the proposed algorithm in retrieving the geometry, the vector field, and the sampling density using toy examples. We also evaluate the performance of the proposed algorithm in extending an empirical function. Using directed graph embedding as a pre-process procedure, we also consider an application of the proposed directed graph embedding algorithm to a web page classification problem.

For the first simulation, let \mathcal{M} be a one dimensional manifold embedded in \mathbb{R}^3 which is given by the following parametric representation:

$$\mathcal{M} = \left\{ x \in \mathbb{R}^3; x = g(\theta) = \begin{bmatrix} (2 + \cos(8\theta)) \cos(\theta) \\ (2 + \cos(8\theta)) \sin(\theta) \\ \sin(8\theta) \end{bmatrix}; 0 \leq \theta \leq 2\pi \right\}. \quad (38)$$

500 points are sampled from the manifold by sampling θ uniformly from the interval $[0, 2\pi]$. This sampling procedure corresponds to sampling from the manifold by the following distribution [28]:

$$p(x) = p(g(\theta)) = \alpha \sqrt{(\cos(8\theta) + 2)^2 + 64}, \quad (39)$$

where α is a normalization factor such that $\int_{\mathcal{M}} p(x) dx = 1$. The similarity kernel is given by (1)–(3) where we choose $\epsilon = 0.2$ and the vector field $\mathbf{r}(x, y)$ is given by

$$\mathbf{r}(x, y) = w(x) + w(y), \quad (40)$$

where

$$w(x) = w(g(\theta)) = 0.2 \begin{bmatrix} -8 \sin(8\theta) \cos(\theta) - \sin(\theta)(\cos(8\theta) + 2) \\ \cos(\theta)(\cos(8\theta) + 2) - 8 \sin(8\theta) \sin(\theta) \\ 8 \cos(8\theta) \end{bmatrix}^T. \quad (41)$$

The results of Algorithm 2 and the method presented in [10] are depicted in Fig. 1. It is obvious that the two algorithms

Table 1

Proposed algorithm for embedding and function extension on a directed graph.

Algorithm 2: Embedding and function extension

Input: Affinity matrix $K \in \mathbb{R}^{n \times n}$ where n is the number of points in the training set, $W \in \mathbb{R}^{n \times 1}$ the affinity between the test point and the training points, $\psi \in \mathbb{R}^{n \times \delta}$ the eigenvectors of the Laplacian on the training data (obtained by Algorithm 1), $\Lambda \in \mathbb{R}^{\delta \times \delta}$ a diagonal matrix of eigenvalues (obtained by Algorithm 1), $R \in \mathbb{R}^{n \times m}$ the vector field components on the training points in the direction of the corresponding coordinates of the embedding, and $f \in \mathbb{R}^{n \times 1}$ the function value on the training set.

1. $\bar{\psi}(\bar{x}) = WK(\text{diag}(WK\mathbf{1}))^{-1}\psi(\mathbf{I} - \Lambda)^{-1}$

2. $\bar{f}(\bar{x}) = f\bar{\psi}(\bar{x})^T$

3. $\bar{R}(\bar{x})_i = R_i\bar{\psi}(\bar{x})^T$; where $i = 1, 2, \dots, m$ and $\bar{R}(\bar{x})_i$ is the i th element of the extension of the vector field to \bar{x} and $R_i \in \mathbb{R}^{1 \times n}$ is the i th row of matrix R .

Table 2

Proposed algorithm for finding the embedding, the sampling density and the vector field representing the directed graph.

Algorithm 1: Directed graph embedding**Input:** Affinity matrix $K \in \mathbb{R}^{n \times n}$ where n is the number of points and embedding dimension m ($m \leq d$).

1. $S = KK^T$
2. $P_s = \text{diag}(\sum_{i=1}^n S_{ij})$
3. $S^1 = P_s^{-1} S P_s^{-1}$
4. $D_s = \text{diag}(\sum_{i=1}^n S_{ij}^1)$
5. $T_s = D_s^{-1} S^1$
6. $L_s = \frac{I - T_s}{2\epsilon}$
7. Compute $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ the eigenvalues of the matrix L_s and $\psi \in \mathbb{R}^{n \times n}$ a matrix of orthonormal columns containing the right eigenvector of L_s . Eigenvectors 2 to $m+1$ of L_s (i.e. 2nd through $(m+1)$ th column of ψ) are the m coordinates of the embedding.
8. Compute $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$, a diagonal matrix of eigenvalues.
9. Compute π the left eigenvector of L_s with eigenvalue 0.
10. $\pi = \frac{\pi}{\sum_{i=1}^n \pi_i}$ is the density distribution over the embedding.
11. $P_a = \text{diag}(\sum_{i=1}^n K_{ij})$
12. $K^1 = P_a^{-1} K P_a^{-1}$
13. $D_a = \text{diag}(\sum_{i=1}^n K_{ij}^1)$
14. $T_a = D_a^{-1} K^1$
15. $L_a = \frac{I - T_a}{\epsilon}$
16. $R = (L_a - L_s)\psi$. Columns 2 to $m+1$ of R are the vector field components in the direction of the corresponding coordinates of the embedding.

are capable of retrieving correctly the geometry, the vector field and the sampling distribution. In this figure, the first and the second rows represent the results of the proposed algorithm and the method presented in [10], respectively. The left column shows the manifold together with the vector field. The embedding using the first two nondegenerate eigenvectors together with the estimated vector field is depicted in the second column. The actual and the estimated sampling distributions are represented in the third column. In order to show the effectiveness of the Algorithm 1 in extending the embedding and the vector field to new data points, we generate 1000 random points uniformly from the interval $[0, 2\pi]$ and obtain the corresponding points on the manifold using (38). Then we compute their distances from the 500 points used in the first part of this experiment. The embedding is then extended using Algorithm 1. The result is depicted in Fig. 2. The results of embedding extension using Nyström extension and geometric harmonics for extending the vector field are also shown in Fig. 2. Using the definition of the vector field in this experiment (i.e. Eqs. (40) and (41)) it can be easily verified that the estimated vector field must be tangent to the embedding. Fig. 2(a) shows that the estimated vector field using the proposed algorithm is nearly tangent to the embedding while the estimated vector field using geometric harmonics is not. This is due to the essential assumption of symmetry of the kernel in geometric harmonics which does not hold in this experiment. As it can be seen from the results, there exists a clear error between actual and estimated sampling distribution for both methods. Although this error is smaller in the proposed method it is still significant. In order to reduce this error, while the number of available data is constant, one must increase the kernel width (i.e. increase the parameter ϵ). While increasing the kernel width decreases the error in estimating the sampling distribution, it increases the error in embedding. In this simulation we have

chosen the kernel width such that the embedding is close to the actual embedding which causes error in estimating the sampling distribution.

In the second experiment we show the advantage of the proposed method in function extension. Let \mathcal{M} be the two dimensional interval $\mathcal{M} = \{x = [x_1, x_2]; -1 \leq x_1, x_2 \leq 1\}$, and the asymmetric similarity kernel be defined by

$$k_\epsilon(x, y) = \begin{cases} \exp\left(-\frac{\|x-y\|^2}{\epsilon^2}\right), & x_1 \leq y_1 \\ 0 & \text{otherwise,} \end{cases} \quad (42)$$

where $\epsilon = 0.5$. Let X be a set of 100 points extracted uniformly from \mathcal{M} . We evaluate the function $f(x) = 1 + \sin(x_1) \cos(x_2)$ on this set. The test set \bar{X} is built by deriving 4000 samples from the manifold uniformly. We use both the proposed method and the geometric harmonics for extending the function on these points. In the extension phase, we assume that we only know the distance of the points in \bar{X} to X . The other distances (i.e. the distance of the points in X to \bar{X}) are assumed to be unknown. The results of function extension are depicted in Fig. 3. It is obvious that the proposed method performs much better than the geometric harmonics. We attribute to the tacit assumption of kernel symmetry in the construction of the geometric harmonics, which is not the case in this simulation. The accuracy of the function extension was quantified by the normalized root mean square error (NRMSE), given by

$$\text{NRMSE} = \sqrt{\frac{\sum_{\bar{x} \in \bar{X}} (f(\bar{x}) - \bar{f}(\bar{x}))^2}{\sum_{\bar{x} \in \bar{X}} f^2(\bar{x})}}. \quad (43)$$

The NRMSE for the aforementioned function extension was 0.0717, compared to 0.2137 for the geometric harmonics approach.

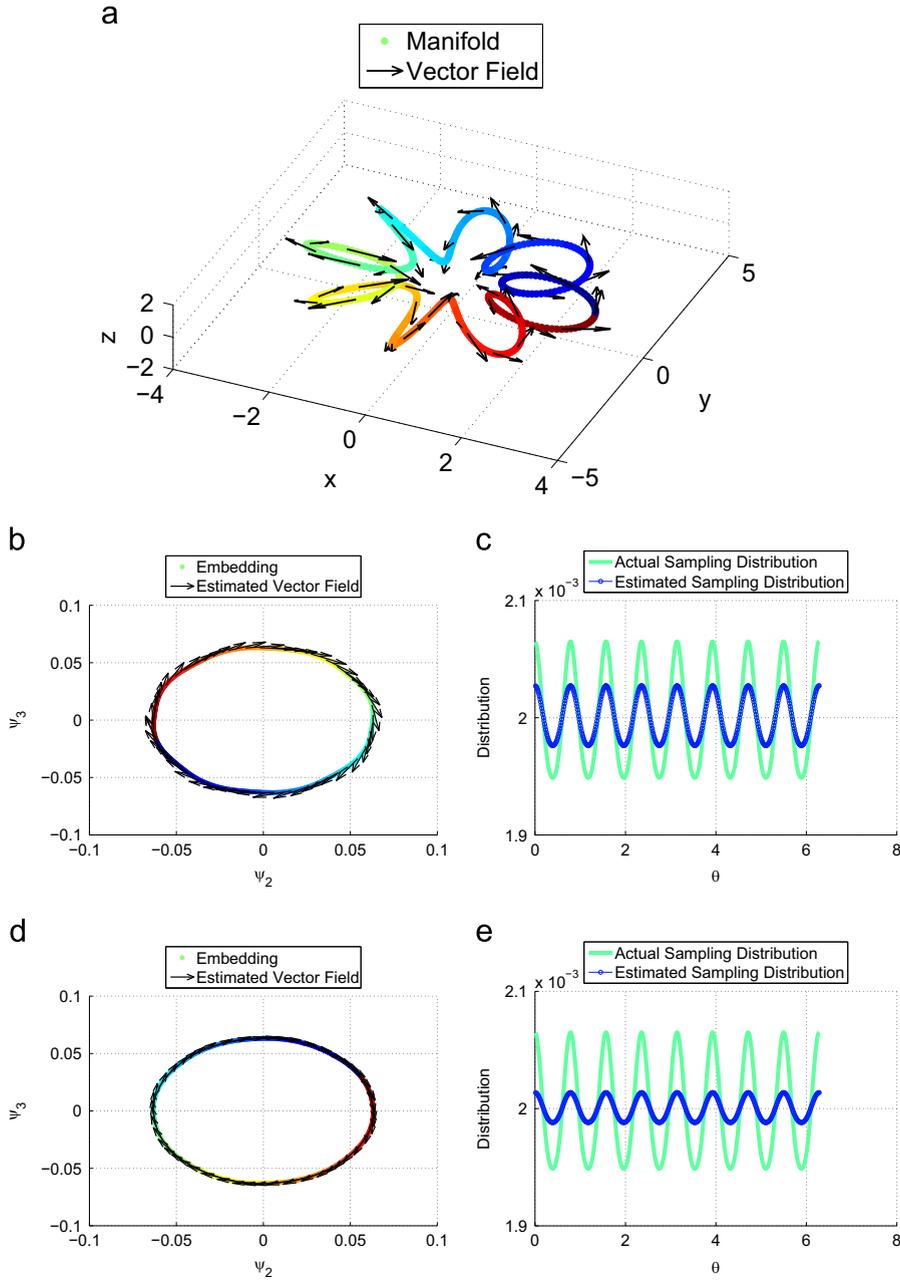


Fig. 1. Comparison of Algorithm 2 (second row) and the method proposed in [10] (third row): (a) One dimensional manifold embedded in \mathbb{R}^3 . (b) and (c) The embedding using the first two nondegenerate eigenvectors together and the estimated vector field, respectively. (d) and (e) The embedding using the first two nondegenerate eigenvectors together and the estimated vector field, respectively.

In order to show the advantage of the proposed directed graph embedding, we consider the web-page classification problem. The World Wide Web can be regarded as a weighted directed graph with binary weights. In this graph each vertex (i.e. v_i) corresponds to a web-page (i.e. web-page number i) and directed edge from v_i to v_j has unity weight if there exists a link from page i to page j and zero otherwise. Using our notation in the previous section a corpus containing n web pages can be modelled with a weighted graph whose affinity matrix $K = \{K_{ij}\}_{i,j=1}^n$ is

given by

$$K_{ij} = \begin{cases} 1, & i=j \text{ or there exists a link from page } i \text{ to page } j \\ 0 & \text{otherwise} \end{cases} \quad (44)$$

In this simulation, we used the WebKb data set. This data set contains a subset of the WWW-pages collected from computer science departments of various universities in January 1997 by the World Wide Knowledge Base (Web-Kb)

project of the Carnegie Mellon University (CMU) text learning group. The 1051 pages contain pages from the four universities Cornell (243), Texas (254), Washington (298) and Wisconsin (256). We use three different directed graph embedding as a pre-process procedure, and then use K-means clustering to classify to four different categories. The affinity matrix and the results are depicted in Fig. 4(a) and (b), respectively. We compare our classification results to those of the methods due to Joncas–Meila (presented in [10]) and Chen–Yang–Tang (presented in [9]). The mean success rates versus number of training samples are

depicted in Fig. 4(b). The mean success rate is computed as follows. Let m be the number of training samples. We randomly choose m web-pages from the corpus for training and use the rest as a test set. The success rate will be the ratio of the number of successfully classified samples to the size of the test set. For each m , we perform this train-test procedure 100 times and report the average. It is obvious that the proposed method outperforms the competing methods.

In our last simulation, we use the proposed function extension method for supervised system identification using speech signals. Our goal is to identify a single pole stable

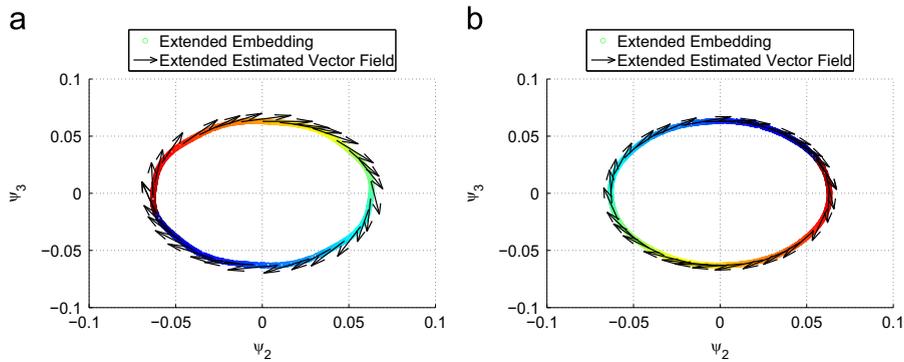


Fig. 2. Comparison of Algorithm 1 (a) and the method proposed in [22] (b) for embedding and vector field extension.

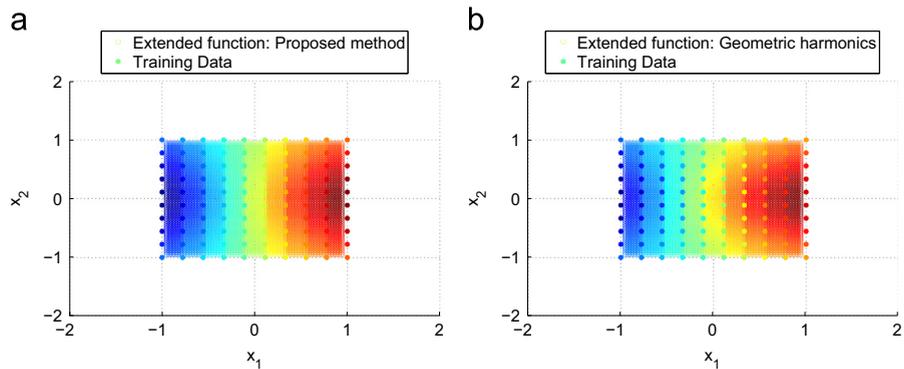


Fig. 3. Comparison of Algorithm 1 (a) and the geometric harmonics (b) for function extension.

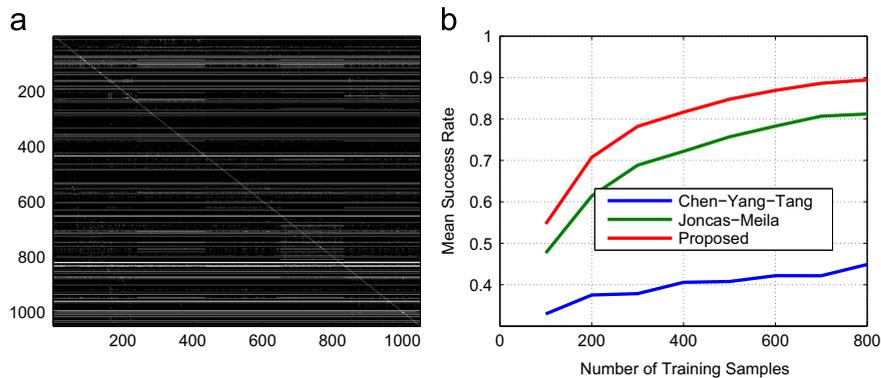


Fig. 4. Affinity matrix (a) and the comparison of the proposed method and the methods due to Joncas–Meila (presented in [10]) and Chen–Yang–Tang (presented in [9]) for web page classification.

discrete time filter (SPF) using arbitrary speech signals. More specifically, the problem is to identify the filter (i.e. finding the location of the pole of the filter) given a filtered version of speech signal. We assume that we only have N_t filtered speech signals $y_i(t) = x_i(t) * h_i(t)$; $1 \leq i \leq N_t$, where $x_i(t)$'s are arbitrary unknown speech signals, $h_i(t)$; $1 \leq i \leq N_t$ are the impulse responses of SPF's corresponding to poles p_i ; $1 \leq i \leq N_t$ and $*$ corresponds to discrete time convolution. The problem is to find $h_{test}(t)$ (or equivalently p_{test} the pole of an unknown test filter) given $y_{test}(t) = x_{test}(t) * h_{test}(t)$ where x_{test} is also an unknown speech signal. Without loss of generality, we assume that all output of filters have the same length. Concisely, our function is a mapping from $y_i(t)$ to p_i which is to be extended to $y_{test}(t)$.

We define our asymmetric kernel as follows:

$$K(y_i, y_j) = K(p_i, p_j) = \begin{cases} \exp\left(-\frac{z}{\epsilon}\right), & z > 0 \\ 0 & \text{otherwise} \end{cases} \quad (45)$$

$$z = \text{Kur}(h_j^{-1}(t) * y_i(t)) - \text{Kur}(x_i(t)) \quad (46)$$

where h_j^{-1} is the inverse of the impulse responses of SPF corresponding to poles p_j and $\text{Kur}(f(t))$ is the empirical kurtosis of the sequence $f(t)$.

In this simulation, we assume that the $0.2 \leq p_i \leq 0.6$. We uniformly extract 60 samples from this interval and produce 60 SPF's. In order to generate training signals, for each SPF, we chose randomly a speech signal from the TIMIT [29] database of length 3 s and filter it with the SPF. The similarity matrix is depicted in Fig. 5(a).

In the testing phase, we uniformly extract 100 samples from the interval $[0.2 \leq p_i^{test} \leq 0.6]$, and produce 100 test SPF's $h_i^{test}(t)$. In order to generate test signals, for each SPF, we chose randomly a speech signal $x_i^{test}(t)$ from the TIMIT [29] database of length 3 s and not used in training step filter it with the SPF (i.e. $y_i^{test}(t) = h_i^{test}(t) * x_i^{test}(t)$). Then we use our algorithm and the algorithm presented in [10] for estimating the pole. The result is depicted in Fig. 5(b). The normalized root mean square error obtained of the algorithm presented in [10] and the proposed method is 14.31% and 2.05%, respectively. It is obvious that the proposed method outperforms the competing method.

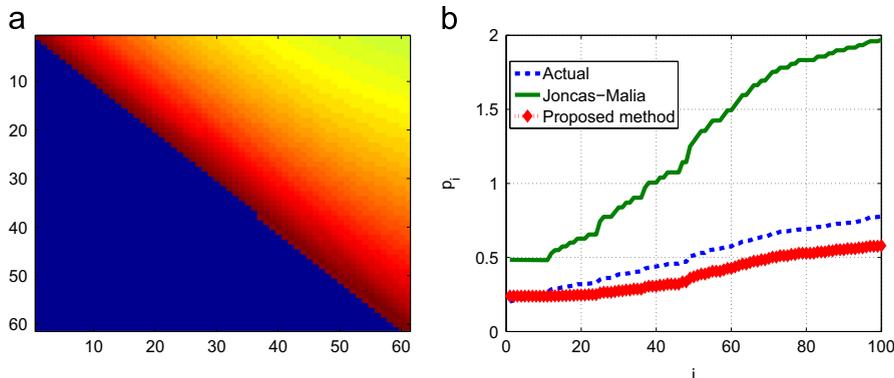


Fig. 5. Affinity matrix (a) and the comparison of the proposed method and the method due to Joncas–Meila (presented in [10]) for supervised system identification.

4. Conclusion

We have proposed a novel method based on Laplace operator for embedding a directed graph. We modelled the observed graph as a sample from a manifold endowed with a vector field, and designed an algorithm that separates and recovers the features of this process: the geometry of the manifold, the data density and the vector field. More specifically, we introduced an anisotropic diffusion type operator and we showed that under certain conditions this operator converges to a Laplace–Beltrami operator on the manifold. Using the fact that eigenfunctions of Laplace–Beltrami operator can be regarded as coordinates on the data set, we derived our embedding procedure. We also proposed a Nyström type embedding extension and a novel technique for extending an empirical function on a directed graph. Simulation results demonstrated the advantage of the proposed method in function extension on a directed graph.

Acknowledgements

The authors thank the anonymous reviewers for their helpful comments. This research was supported by the Israel Science Foundation (Grant no. 1130/11).

Appendix A. Proof of Lemma 1

In this appendix we provide the proof to Lemma 1. This lemma was introduced in [10] and the proof is given in a technical report. For consistency and since we used this lemma to prove Lemma 2, we provide the proof of this lemma here. We adopt the same notation as in the proof of Lemma 8 in [6]. Let

$$G_\epsilon f(x) = \int_{\mathcal{M}} k_\epsilon(x, y) f(y) dy, \quad (47)$$

and $0 < \gamma < 1/2$. Then we have

$$\left| \int_{y \in \mathcal{M}: \|x-y\| > \epsilon^\gamma} k_\epsilon(x, y) f(y) dy \right| = \left| \int_{y \in \mathcal{M}: \|x-y\| > \epsilon^\gamma} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) h_\epsilon(x, y) f(y) dy \right|$$

$$\begin{aligned}
&\leq \|f\|_\infty \left(1 + \left\| \frac{\mathbf{r}(x, y)}{2} \right\|_\infty\right) \frac{1}{\epsilon^{d/2}} \int_{y \in \mathcal{M}: \|x-y\| > \epsilon^r} \|x-y\| h \\
&\quad \times \left(\frac{\|x-y\|^2}{\epsilon} \right) dy \Big| = \|f\|_\infty \left(1 + \left\| \frac{\mathbf{r}(x, y)}{2} \right\|_\infty\right) \\
&\quad \times \left| \frac{1}{\epsilon^{d/2}} \int_{y \in \mathcal{M}: \|y\| > \epsilon^r} \|y\| h \left(\frac{\|y\|^2}{\epsilon} \right) dy \right| = \|f\|_\infty \\
&\quad \times \left(1 + \left\| \frac{\mathbf{r}(x, y)}{2} \right\|_\infty\right) \left| \sqrt{\epsilon} \int_{y \in \mathcal{M}: \|y\| > \epsilon^{r-1/2}} \|y\| h(\|y\|^2) dy \right| \\
&= C \|f\|_\infty \left(1 + \left\| \frac{\mathbf{r}(x, y)}{2} \right\|_\infty\right) \left| \sqrt{\epsilon} \int_{y \in \mathbb{R}^d: y > \epsilon^{r-1/2}} y^d h(y^2) dy \right|, \quad (48)
\end{aligned}$$

where C is a constant and we used generalized spherical coordinates for obtaining the last equation. Using the exponential decay property of h , this term is exponentially small and bounded by $\mathcal{O}(\epsilon^{3/2})$. Hence, we have

$$\begin{aligned}
G_\epsilon f(x) &= \int_{y \in \mathcal{M}} k_\epsilon(x, y) f(y) dy = \int_{y \in \mathcal{M}: \|x-y\| \leq \epsilon^r} \\
&\quad \times \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) h_\epsilon(x, y) f(y) dy + \mathcal{O}(\epsilon^{3/2}) \quad (49)
\end{aligned}$$

Now we can use a Taylor expansion since everything is localized around x . The two terms in the above integration can be expanded as follows. Let

$$g(y) = \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) f(y) \quad (50)$$

then

$$\begin{aligned}
g(y) &= g(x) + \sum_{i=1}^d s_i \frac{\partial g}{\partial s_i}(x) + \frac{1}{2} \sum_{i=1}^d \sum_{j=1}^d s_i s_j \frac{\partial^2 g}{\partial s_i \partial s_j}(x) \\
&\quad + Q_3(s_1, s_2, \dots, s_d) + \mathcal{O}(\epsilon^2) \\
&= f(x) + \sum_{i=1}^d s_i \left[\frac{\partial}{\partial s_i} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \right]_{y=x} f(x) + \frac{\partial f}{\partial s_i}(x) \\
&\quad \times \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \Big|_{y=x} \\
&\quad + \frac{1}{2} \sum_{i=1}^d \sum_{j=1}^d s_i s_j \left[\frac{\partial^2}{\partial s_i \partial s_j} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \right]_{y=x} f(x) \\
&\quad + \frac{\partial f}{\partial s_j}(x) \frac{\partial}{\partial s_i} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \Big|_{y=x} \\
&\quad + \frac{\partial^2 f}{\partial s_i \partial s_j}(x) \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \Big|_{y=x} + \frac{\partial f}{\partial s_i}(x) \frac{\partial}{\partial s_j}(x) \\
&\quad \times \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \Big|_{y=x} + Q_3(u) + \mathcal{O}(\epsilon^2), \quad (51)
\end{aligned}$$

where s_i is the i -th geodesic coordinate and $Q_m(s_1, s_2, \dots, s_d)$ denotes a homogeneous polynomial of degree m of the variables s_1, s_2, \dots, s_d which depends on x and might change from line to line. This expansion can be written in the local coordinates of the tangent plane at point x as

$$g(u) = f(x) + \sum_{i=1}^d u_i \left[\frac{\partial}{\partial s_i} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \right]_{y=x} f(x) + \frac{\partial f}{\partial s_i}(x)$$

$$\begin{aligned}
&+ \frac{1}{2} \sum_{i=1}^d \sum_{j=1}^d u_i u_j \left[\frac{\partial^2}{\partial s_i \partial s_j} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \right]_{y=x} f(x) \\
&+ \frac{\partial f}{\partial s_j}(x) \frac{\partial}{\partial s_i} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \Big|_{y=x} \\
&+ \left. \frac{\partial^2 f}{\partial s_i \partial s_j}(x) + \frac{\partial f}{\partial s_i}(x) \frac{\partial}{\partial s_j} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \right]_{y=x} \\
&+ Q_3(u) + \mathcal{O}(\epsilon^2), \quad (52)
\end{aligned}$$

where u_i is the i -th local coordinate of the tangent plane at point x and u is the orthogonal projection of y on the tangent plane at point x . Similarly, using Lemma 7 of [6], the kernel can be Taylor expanded as

$$h\left(\frac{\|x-y\|^2}{\epsilon}\right) = h\left(\frac{\|u\|^2}{\epsilon}\right) + \left(\frac{Q_5(u) + Q_4(u)}{\epsilon}\right) h'\left(\frac{\|u\|^2}{\epsilon}\right) + \mathcal{O}(\epsilon^2). \quad (53)$$

Substituting (52) and (53) into (49), and changing the variable $(y-x) \rightarrow u$, we get

$$\begin{aligned}
G_\epsilon f(x) &= \int_{y \in \mathcal{M}: \|x-y\| \leq \epsilon^r} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) h_\epsilon(x, y) f(y) dy \\
&+ \mathcal{O}(\epsilon^{3/2}) = \frac{1}{\epsilon^{d/2}} \int_{\|u\| \leq \epsilon^r} g(u) h\left(\frac{\|u\|^2}{\epsilon}\right) (1 + Q_2(u) \\
&+ Q_3(u)) du + \mathcal{O}(\epsilon^{3/2}). \quad (54)
\end{aligned}$$

Using the exponential decay property of the kernel, we can extend the domain of integration to \mathbb{R}^d . Hence, we have

$$\begin{aligned}
G_\epsilon f(x) &= \frac{1}{\epsilon^{d/2}} \int_{u \in \mathbb{R}^d} g(u) h\left(\frac{\|u\|^2}{\epsilon}\right) (1 + Q_2(u) + Q_3(u)) du \\
&+ \mathcal{O}(\epsilon^2) = \frac{1}{\epsilon^{d/2}} \int_{u \in \mathbb{R}^d} \left\{ f(x) + \sum_{i=1}^d u_i \right. \\
&\times \left[\frac{\partial}{\partial s_i} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \right]_{y=x} f(x) + \frac{\partial f}{\partial s_i}(x) \\
&+ \frac{1}{2} \sum_{i=1}^d \sum_{j=1}^d u_i u_j \left[\frac{\partial^2}{\partial s_i \partial s_j} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \right]_{y=x} f(x) \\
&+ \frac{\partial f}{\partial s_j}(x) \frac{\partial}{\partial s_i} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \Big|_{y=x} + \frac{\partial^2 f}{\partial s_i \partial s_j}(x) \\
&+ \left. \frac{\partial f}{\partial s_i}(x) \frac{\partial}{\partial s_j} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \right]_{y=x} \Big\} \\
&\times \left[h\left(\frac{\|u\|^2}{\epsilon}\right) + \frac{Q_5(u) + Q_4(u)}{\epsilon} h'\left(\frac{\|u\|^2}{\epsilon}\right) \right] \\
&\times [1 + Q_2(u) + Q_3(u)] du + \mathcal{O}(\epsilon^2). \quad (55)
\end{aligned}$$

Setting the integration of the odd function to zero, we get

$$\begin{aligned}
G_\epsilon f(x) &= \frac{1}{\epsilon^{d/2}} \int_{u \in \mathbb{R}^d} \left\{ f(x) + \frac{1}{2} \sum_{i=1}^d u_i^2 \right. \\
&\times \left[\frac{\partial^2}{\partial s_i^2} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \right]_{y=x} f(x) \\
&+ \left. 2 \frac{\partial f}{\partial s_i}(x) \frac{\partial}{\partial s_i} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x)\right) \right]_{y=x} + \frac{\partial^2 f}{\partial s_i^2}(x) \Big\} \\
&\times \left[h\left(\frac{\|u\|^2}{\epsilon}\right) + \frac{Q_4(u)}{\epsilon} h'\left(\frac{\|u\|^2}{\epsilon}\right) \right]
\end{aligned}$$

$$\times [1 + Q_2(u)] du + \mathcal{O}(\epsilon^2). \quad (56)$$

Changing variables according to $u \rightarrow \sqrt{\epsilon}u$ we have

$$\begin{aligned} G_\epsilon f(x) &= \int_{u \in \mathbb{R}^d} \left\{ f(x) + \frac{1}{2} \sum_{i=1}^d \epsilon u_i^2 \left[\frac{\partial^2}{\partial s_i^2} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x) \right) \right]_{y=x} f(x) \right. \\ &\quad \left. + 2 \frac{\partial f}{\partial s_i}(x) \frac{\partial}{\partial s_i} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x) \right) \right|_{y=x} + \frac{\partial^2 f}{\partial s_i^2}(x) \Big\} \\ &\quad \times \left[h(\|u\|^2) + \epsilon Q_4(u) h'(\|u\|^2) \right] \times [1 + \epsilon Q_2(u)] du + \mathcal{O}(\epsilon^2) \\ &= f(x) \int_{u \in \mathbb{R}^d} h(\|u\|^2) du + \epsilon f(x) \int_{u \in \mathbb{R}^d} (Q_2(u) h(\|u\|^2) \\ &\quad + Q_4(u) h'(\|u\|^2)) du \\ &\quad + \frac{\epsilon}{2} \sum_{i=1}^d \left[\frac{\partial^2}{\partial s_i^2} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x) \right) \right]_{y=x} f(x) \\ &\quad + 2 \frac{\partial f}{\partial s_i}(x) \frac{\partial}{\partial s_i} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x) \right) \Big|_{y=x} + \frac{\partial^2 f}{\partial s_i^2}(x) \Big] \\ &\quad \times \int_{u \in \mathbb{R}^d} u_i^2 h(\|u\|^2) dt + \mathcal{O}(\epsilon^2). \end{aligned} \quad (57)$$

It is easy to verify that

$$\nabla \cdot \mathbf{r} = \sum_{i=1}^d \frac{\partial^2}{\partial s_i^2} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x) \right) \Big|_{y=x} \quad (58)$$

$$\nabla f \cdot \mathbf{r} = \sum_{i=1}^d 2 \frac{\partial f}{\partial s_i}(x) \frac{\partial}{\partial s_i} \left(1 + \frac{\mathbf{r}(x, y)}{2} \cdot (y-x) \right) \Big|_{y=x} \quad (59)$$

$$\Delta f = \sum_{i=1}^d \frac{\partial^2 f}{\partial s_i^2}(x), \quad (60)$$

where $\nabla \cdot \mathbf{r}$, ∇f and Δf are the divergence of the vector field \mathbf{r} , the gradient of function $f(\cdot)$ and the Laplace–Beltrami operator on the manifold, respectively. Substituting (58)–(60) into (57) we have

$$\begin{aligned} G_\epsilon f(x) &= f(x) \int_{u \in \mathbb{R}^d} h(\|u\|^2) du + \epsilon f(x) \int_{u \in \mathbb{R}^d} (Q_2(u) h(\|u\|^2) \\ &\quad + Q_4(u) h'(\|u\|^2)) du + \frac{\epsilon}{2} (\nabla \cdot \mathbf{r} + \nabla f \cdot \mathbf{r} + \Delta f) \\ &\quad \times \int_{u \in \mathbb{R}^d} u_i^2 h(\|u\|^2) dt + \mathcal{O}(\epsilon^2) = m_0 f(x) \\ &\quad + \epsilon \frac{m_2}{2} (\omega(x) f(x) + \Delta f(x) + \mathbf{r} \cdot \nabla f(x) + f(x) \nabla \cdot \mathbf{r}) + \mathcal{O}(\epsilon^2). \end{aligned} \quad (61)$$

where

$$m_0 = \int_{\mathbb{R}^d} h(\|u\|^2) du \quad (62)$$

$$m_2 = \int_{\mathbb{R}^d} u_i^2 h(\|u\|^2), du \quad (63)$$

$$\omega(x) = \frac{2}{m_2} \int_{u \in \mathbb{R}^d} (Q_2(u) h(\|u\|^2) + Q_4(u) h'(\|u\|^2)) du. \quad (64)$$

In order to prove (18), note that

$$\int_{\mathcal{M}} k_\epsilon(y, x) f(y) dy = \int_{\mathcal{M}} \left(1 + \frac{\mathbf{r}(y, x)}{2} \cdot (x-y) \right) h_\epsilon(y, x) f(y) dy$$

$$= \int_{\mathcal{M}} \left(1 - \frac{\mathbf{r}(y, x)}{2} \cdot (y-x) \right) h_\epsilon(y, x) f(y) dy, \quad (65)$$

Since $\mathbf{r}(y, x)$ and $h_\epsilon(y, x)$ are assumed to be symmetric, we have

$$\begin{aligned} \int_{\mathcal{M}} k_\epsilon(y, x) f(y) dy &= \int_{\mathcal{M}} \left(1 - \frac{\mathbf{r}(y, x)}{2} \cdot (y-x) \right) h_\epsilon(y, x) f(y) dy \\ &= \int_{\mathcal{M}} \left(1 - \frac{\mathbf{r}(x, y)}{2} \cdot (y-x) \right) h_\epsilon(x, y) f(y) dy \\ &= m_0 f(x) + \epsilon \frac{m_2}{2} (\omega(x) f(x) \\ &\quad + \Delta f(x) - \mathbf{r} \cdot \nabla f(x) - f(x) \nabla \cdot \mathbf{r}) + \mathcal{O}(\epsilon^2). \end{aligned} \quad (66)$$

where the last equation is simply obtained by substituting $-\mathbf{r}(y, x)$ instead of $\mathbf{r}(y, x)$ in (17). \square

Appendix B. Proof of Theorem 3

In this appendix we provide the proof to Theorem 3. First we prove (28). Note that

$$\begin{aligned} T_{\epsilon, \alpha}^{aa} f(x) &= \int_{\mathcal{M}} t_{k, \epsilon, \alpha}(x, y) f(y) p(y) dy \\ &= \int_{\mathcal{M}} \frac{k_\epsilon^{(\alpha)}(x, y)}{d_{k, \epsilon}(x)} f(y) p(y) dy \\ &= \frac{\int_{\mathcal{M}} k_\epsilon^{(\alpha)}(x, y) f(y) p(y) dy}{\int_{\mathcal{M}} k_\epsilon^{(\alpha)}(x, y) p(y) dy}, \end{aligned} \quad (67)$$

where

$$k_\epsilon^{(\alpha)}(x, y) = \frac{k_\epsilon(x, y)}{p_{k, \epsilon}^\alpha(x) p_{k, \epsilon}^\alpha(y)}. \quad (68)$$

Using Lemma 1 and the definition of $p_{k, \epsilon}(x)$ in (7) we have

$$\begin{aligned} p_{k, \epsilon}(x) &= \int_{\mathcal{M}} k_\epsilon(x, y) p(y) dy \\ &= p(x) + \epsilon (\omega(x) p(x) + \Delta p(x) + \mathbf{r} \cdot \nabla p(x) + p(x) \nabla \cdot \mathbf{r}) \end{aligned} \quad (69)$$

up to a term of order $\mathcal{O}(\epsilon^2)$. Hence,

$$\frac{1}{p_{k, \epsilon}^\alpha(x)} = p_{k, \epsilon}^{-\alpha}(x) = p^{-\alpha}(x) \left(1 - \alpha \epsilon \left(\omega(x) + \frac{\Delta p(x)}{p(x)} + \frac{\mathbf{r} \cdot \nabla p(x)}{p(x)} + \nabla \cdot \mathbf{r} \right) \right) \quad (70)$$

up to a term of order $\mathcal{O}(\epsilon^2)$. We define

$$K_\epsilon^\alpha \phi(x) = \int_{\mathcal{M}} k_\epsilon^{(\alpha)}(x, y) \phi(y) p(y) dy. \quad (71)$$

Then using the definition of $k_\epsilon^{(\alpha)}(x, y)$, we have

$$\begin{aligned} K_\epsilon^\alpha \phi(x) &= \int_{\mathcal{M}} \frac{k_\epsilon(x, y)}{p_{k, \epsilon}^\alpha(x) p_{k, \epsilon}^\alpha(y)} \phi(y) p(y) dy \\ &= p_{k, \epsilon}^{-\alpha}(x) \int_{\mathcal{M}} k_\epsilon(x, y) \phi(y) \frac{p(y)}{p_{k, \epsilon}^\alpha(y)} dy. \end{aligned} \quad (72)$$

Substituting the approximation of $p_{k, \epsilon}^{-\alpha}(x)$ obtained in (70) into (72) we have

$$\begin{aligned} K_\epsilon^\alpha \phi(x) &= p_{k, \epsilon}^{-\alpha}(x) \int_{\mathcal{M}} k_\epsilon(x, y) \phi(y) p^{1-\alpha}(y) \left(1 - \alpha \epsilon \left(\omega(y) \right. \right. \\ &\quad \left. \left. + \frac{\Delta p(y)}{p(y)} + \frac{\mathbf{r} \cdot \nabla p(y)}{p(y)} + \nabla \cdot \mathbf{r} \right) \right) dy. \end{aligned} \quad (73)$$

Defining

$$g(x) = \omega(x) + \frac{\Delta p(x)}{p(x)} + \frac{\mathbf{r} \cdot \nabla p(x)}{p(x)} + \nabla \cdot \mathbf{r}, \quad (74)$$

and using Lemma 1 we have

$$\begin{aligned} K_{\epsilon}^{\alpha} \phi(x) &= p_{k,\epsilon}^{-\alpha}(x) \int_{\mathcal{M}} k_{\epsilon}(x, y) \phi(y) \phi(y) p^{1-\alpha}(y) (1 - \alpha \epsilon g(y)) dy \\ &= p_{k,\epsilon}^{-\alpha}(x) [\phi(x) p^{1-\alpha}(x) (1 - \alpha \epsilon g(x)) + \epsilon (\omega(x) \\ &\quad \times (\phi(x) p^{1-\alpha}(x) (1 - \alpha \epsilon g(x))) + \Delta(\phi(x) p^{1-\alpha}(x) \\ &\quad \times (1 - \alpha \epsilon g(x))) + \mathbf{r} \cdot \nabla(\phi(x) p^{1-\alpha}(x) (1 - \alpha \epsilon g(x))) \\ &\quad + (\phi(x) p^{1-\alpha}(x) (1 - \alpha \epsilon g(x))) \nabla \cdot \mathbf{r})]. \end{aligned} \quad (75)$$

Discarding terms of order $\mathcal{O}(\epsilon^2)$ we have

$$K_{\epsilon}^{\alpha} \phi = p_{k,\epsilon}^{-\alpha} [\phi p^{1-\alpha} (1 - \alpha \epsilon g) + \epsilon (\omega \phi p^{1-\alpha} + \Delta(\phi p^{1-\alpha}) + \mathbf{r} \cdot \nabla(\phi p^{1-\alpha}) + \phi p^{1-\alpha} \nabla \cdot \mathbf{r})], \quad (76)$$

where we have dropped the argument of each function (i.e. (x)) for simplicity. Substituting $g(x)$ from (74) into (76) we have

$$\begin{aligned} K_{\epsilon}^{\alpha} \phi &= p_{k,\epsilon}^{-\alpha} \left[\phi p^{1-\alpha} - \epsilon \left(\alpha \phi p^{1-\alpha} \omega + \alpha \phi p^{1-\alpha} \frac{\Delta p}{p} \right. \right. \\ &\quad \left. \left. + \alpha \phi p^{1-\alpha} \frac{\mathbf{r} \cdot \nabla p}{p} + \alpha \phi p^{1-\alpha} \nabla \cdot \mathbf{r} \right) \right. \\ &\quad \left. + \epsilon (\omega \phi p^{1-\alpha} + \Delta(\phi p^{1-\alpha}) + \mathbf{r} \cdot \nabla(\phi p^{1-\alpha}) \right. \\ &\quad \left. + \phi p^{1-\alpha} \nabla \cdot \mathbf{r}) \right] = p_{k,\epsilon}^{-\alpha} [\phi p^{1-\alpha} + \epsilon ((1 - \alpha) p^{1-\alpha} (\phi \omega \\ &\quad + \phi \nabla \cdot \mathbf{r}) - \alpha \phi \left(\frac{\Delta p}{p} + \frac{\mathbf{r} \cdot \nabla p}{p} \right) + \Delta(\phi p^{1-\alpha}) \\ &\quad + \mathbf{r} \cdot \nabla(\phi p^{1-\alpha}))] \\ &= p_{k,\epsilon}^{-\alpha} p^{1-\alpha} \left[\phi + \epsilon ((1 - \alpha) (\phi \omega + \phi \nabla \cdot \mathbf{r}) \right. \\ &\quad \left. - \alpha \phi \left(\frac{\Delta p}{p} + \frac{\mathbf{r} \cdot \nabla p}{p} \right) + \frac{\Delta(\phi p^{1-\alpha})}{p^{1-\alpha}} + \frac{\mathbf{r} \cdot \nabla(\phi p^{1-\alpha})}{p^{1-\alpha}} \right]. \end{aligned} \quad (77)$$

Using Eq. (67), the definition of the operator K_{ϵ}^{α} in (71) and Eq. (77), we get

$$T_{\epsilon,\alpha}^{aa} \phi = \frac{K_{\epsilon}^{\alpha} \phi(x)}{K_{\epsilon}^{\alpha} 1} = \frac{\phi + \epsilon \left((1 - \alpha) (\phi \omega + \phi \nabla \cdot \mathbf{r}) - \alpha \phi \left(\frac{\Delta p}{p} + \frac{\mathbf{r} \cdot \nabla p}{p} \right) + \frac{\Delta(\phi p^{1-\alpha})}{p^{1-\alpha}} + \frac{\mathbf{r} \cdot \nabla(\phi p^{1-\alpha})}{p^{1-\alpha}} \right)}{1 + \epsilon \left((1 - \alpha) (\omega + \nabla \cdot \mathbf{r}) - \alpha \left(\frac{\Delta p}{p} + \frac{\mathbf{r} \cdot \nabla p}{p} \right) + \frac{\Delta(p^{1-\alpha})}{p^{1-\alpha}} + \frac{\mathbf{r} \cdot \nabla(p^{1-\alpha})}{p^{1-\alpha}} \right)}. \quad (78)$$

Utilizing the approximation $(1 + \epsilon a)^{-1} \approx 1 - \epsilon a$ and discarding terms of order $\mathcal{O}(\epsilon^2)$ we get

$$\begin{aligned} T_{\epsilon,\alpha}^{aa} \phi &= \phi + \epsilon \left((1 - \alpha) (\phi \omega + \phi \nabla \cdot \mathbf{r}) - \alpha \phi \left(\frac{\Delta p}{p} + \frac{\mathbf{r} \cdot \nabla p}{p} \right) \right. \\ &\quad \left. + \frac{\Delta(\phi p^{1-\alpha})}{p^{1-\alpha}} + \frac{\mathbf{r} \cdot \nabla(\phi p^{1-\alpha})}{p^{1-\alpha}} \right) \\ &\quad - \phi \epsilon \left((1 - \alpha) (\omega + \nabla \cdot \mathbf{r}) - \alpha \left(\frac{\Delta p}{p} + \frac{\mathbf{r} \cdot \nabla p}{p} \right) \right. \\ &\quad \left. + \frac{\Delta(p^{1-\alpha})}{p^{1-\alpha}} + \frac{\mathbf{r} \cdot \nabla(p^{1-\alpha})}{p^{1-\alpha}} \right) \end{aligned}$$

$$\begin{aligned} &= \phi + \epsilon \left(\frac{\Delta(\phi p^{1-\alpha})}{p^{1-\alpha}} - \frac{\Delta(p^{1-\alpha})}{p^{1-\alpha}} \phi \right. \\ &\quad \left. + \frac{\mathbf{r} \cdot \nabla(\phi p^{1-\alpha})}{p^{1-\alpha}} - \frac{\mathbf{r} \cdot \nabla(p^{1-\alpha})}{p^{1-\alpha}} \phi \right) \\ &= \phi + \epsilon \left(\Delta \phi + 2 \frac{\nabla \phi \cdot \nabla p^{1-\alpha}}{p^{1-\alpha}} + \mathbf{r} \cdot \nabla \phi \right), \end{aligned} \quad (79)$$

where for obtaining the last equation we have used the following identities:

$$\Delta(fg) = g \Delta f + f \Delta g + 2 \nabla f \cdot \nabla g \quad (80)$$

$$\nabla(fg) = f \nabla g + g \nabla f. \quad (81)$$

Using the assumption $p = e^{-U}$, the definition of $L_{\epsilon,\alpha}^{aa}$ in (26) and taking the limit when $\epsilon \rightarrow 0$, we get

$$\begin{aligned} \lim_{\epsilon \rightarrow 0} L_{\epsilon,\alpha}^{aa} f &= \lim_{\epsilon \rightarrow 0} \frac{1 - T_{\epsilon,\alpha}^{aa} f}{\epsilon} \\ &= \Delta f + 2 \frac{\nabla f \cdot \nabla e^{-(1-\alpha)U}}{e^{-(1-\alpha)U}} + \mathbf{r} \cdot \nabla f \\ &= \Delta f - 2(1 - \alpha) \nabla f \cdot \nabla U + \mathbf{r} \cdot \nabla f, \end{aligned} \quad (82)$$

where we have used $\nabla e^{-(1-\alpha)U} = -(1 - \alpha) e^{-(1-\alpha)U} \nabla U$.

To prove (29) note that, similar to the proof of the first part, all the results are based on a Taylor approximation of the kernel used in the integration operator used for obtaining the diffusion operator $T_{\epsilon,\alpha}^{aa}$. In order to obtain the limit of infinitesimal generator of the Markov chain (i.e. $L_{\epsilon,\alpha}^{ss}$) constructed from $T_{\epsilon,\alpha}^{ss}$, note that the Taylor approximation of the kernel used in the integration operator for obtaining the diffusion operator $T_{\epsilon,\alpha}^{ss}$ is the same as that of $T_{\epsilon,\alpha}^{aa}$ when $\mathbf{r} = 0$ and ϵ is replaced by 2ϵ . Hence, using exactly the same procedure that led to (82), the result of (29) can be obtained. \square

References

- [1] P.D. Hoff, A.E. Raftery, M.S. Handcock, Latent space approaches to social network analysis, *J. Am. Stat. Assoc.* 97 (460) (2002) 1090–1098.
- [2] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* 290 (no. 5500) (2000) 2323–2326.
- [3] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Comput.* 15 (2003) 1373–1396.
- [4] D.L. Donoho, C. Grimes, Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data, *Proc. Natl. Acad. Sci.* 100 (2003) 5591–5596.
- [5] Z. Zhang, H. Zha, Principal manifolds and nonlinear dimension reduction via local tangent space alignment, arXiv preprint cs/0212008.

- [6] R.R. Coifman, S. Lafon, Diffusion maps, *Appl. Comput. Harmon. Anal.* 21 (2006) 5–30.
- [7] C.L. Giles, K.D. Bollacker, S. Lawrence, An automatic citation indexing system, in: *Proceedings of the Third ACM Conference on Digital Libraries*, ACM, Pittsburgh, PA, USA, 1998, pp. 89–98.
- [8] W. Pentney, M. Meila, Spectral clustering of biological sequence data, in: M.M. Veloso, S. Kambhampati (Eds.), *AAAI, AAAI Press/The MIT Press*, Pittsburgh, Pennsylvania, USA, 2005, pp. 845–850.
- [9] T. Chen, Q. Yang, X. Tang, Directed graph embedding, in: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2007, pp. 2707–2712.
- [10] D. Perrault-Joncas, M. Meilä, Directed graph embedding: an algorithm based on continuous limits of Laplacian-type operators, in: *Advances in Neural Information Processing Systems*, vol. 25.
- [11] Coifman Ronald R., Matthew J. Hirn, Bi-stochastic kernels via asymmetric affinity functions, *Applied and Computational Harmonic Analysis* 35.1 (2013) 177–180.
- [12] J. Shi, J. Malik, Normalized cuts and image segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (8) (2000) 888–905.
- [13] C. Fowlkes, S. Belongie, F. Chung, J. Malik, Spectral grouping using the Nyström method, *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (2004) 214–225.
- [14] L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, in: *Advances in Neural Information Processing Systems*, vol. 17, 2005, pp. 1601–1608.
- [15] S. Gepshtein, Y. Keller, Image completion by diffusion maps and spectral relaxation, *IEEE Trans. Image Process.* 22 (8) (2013) 2983–2994.
- [16] R. Talmon, I. Cohen, S. Gannot, Transient noise reduction using nonlocal diffusion filters, *IEEE Trans. Audio Speech Lang. Process.* 19 (6) (2011) 1584–1599.
- [17] S. Mousazadeh, I. Cohen, Voice activity detection in presence of transient noise using spectral clustering, *IEEE Trans. Audio Speech Lang. Process.* 21 (6) (2013) 1261–1271.
- [18] A. Singer, Spectral independent component analysis, *Appl. Comput. Harmon. Anal.* 21 (1) (2006) 135–144.
- [19] A. Singer, R. Coifman, Non-linear independent component analysis with diffusion maps, *Appl. Comput. Harmon. Anal.* 25 (2) (2008) 226–239.
- [20] R. Talmon, D. Kushnir, R.R. Coifman, I. Cohen, S. Gannot, Parametrization of linear systems using diffusion kernels, *IEEE Trans. Signal Process.* 60 (3) (2012) 1159–1173.
- [21] R. Talmon, I. Cohen, S. Gannot, Supervised source localization using diffusion kernels, in: *Proceedings of 2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA-2011*, New Paltz, NY, October 16–19, 2011.
- [22] R. Coifman, S. Lafon, Geometric harmonics: a novel tool for multi-scale out-of-sample extension of empirical functions, *Appl. Comput. Harmon. Anal.* 21 (2006) 31–52.
- [23] Kushnir, Dan, Ali Haddad, Ronald R. Coifman, Anisotropic diffusion on sub-manifolds with application to earth structure classification., *Applied and Computational Harmonic Analysis* 32.2 (2012) 280–294.
- [24] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, F. Warner, S. Zucker, Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps, in: *Proceedings of the National Academy of Sciences*, 2005, pp. 7426–7431.
- [25] D. Ting, L. Huang, M. Jordan, An analysis of the convergence of graph Laplacians, arXiv preprint [arXiv:1101.5435](https://arxiv.org/abs/1101.5435).
- [26] R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, S.W. Zucker, Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps, *Proc. Natl. Acad. Sci. USA* 102 (21) (2005) 7426–7431.
- [27] R.R. Coifman, S. Lafon, Geometric harmonics: A novel tool for multiscale out-of-sample extension of empirical functions, *Appl. Comput. Harmon. Anal.* 21 (2006) 31–52.
- [28] P. Diaconis, S. Holmes, M. Shahshahani, Sampling from a manifold, in: *Advances in Modern Statistical Theory and Applications: A Festschrift in Honor of Morris L. Eaton*, Institute of Mathematical Statistics, 2013, pp. 102–125.
- [29] J.S. Garofolo, Getting Started with the DARPA TIMIT CD-ROM: An Acoustic-Phonetic Continuous Speech Database, National Institute of Standards and Technology (NIST), Gaithersburg, MD, February 1993.