Incremental Construction of Low-Dimensional Data Representations

Alexander Kuleshov[1] and Alexander Bernstein[1,2](✉)

[1] Skolkovo Institute of Science and Technology, Moscow, Russia
{A.Kuleshov,a.bernstein}@skoltech.ru
[2] Kharkevich Institute for Information Transmission Problems RAS, Moscow, Russia

Abstract. Various Dimensionality Reduction algorithms transform initial high-dimensional data into their lower-dimensional representations preserving chosen properties of the initial data. Typically, such algorithms use the solution of large-dimensional optimization problems, and the incremental versions are designed for many popular algorithms to reduce their computational complexity. Under manifold assumption about high-dimensional data, advanced manifold learning algorithms should preserve the Data manifold and its differential properties such as tangent spaces, Riemannian tensor, etc. Incremental version of the Grassmann&Stiefel Eigenmaps manifold learning algorithm, which has asymptotically minimal reconstruction error, is proposed in this paper and has significantly smaller computational complexity in contrast to the initial algorithm.

Keywords: Machine learning · Dimensionality reduction · Manifold learning · Tangent bundle manifold learning · Incremental learning

1 Introduction

The general goal of data analysis is to extract previously unknown information from a given dataset. Many data analysis tasks, such as pattern recognition, classification, clustering, prognosis, and others, deal with real-world data that are presented in high-dimensional spaces, and the 'curse of dimensionality' phenomena are often an obstacle to the use of many methods for solving these tasks.

Fortunately, in many applications, especially in pattern recognition, the real high-dimensional data occupy only a very small part in the high dimensional 'observation space' R^p; it means that an intrinsic dimension q of the data is small compared to the dimension p (usually, q << p) [1, 2]. Various dimensionality reduction (feature extraction) algorithms, whose goal is a finding of a low-dimensional parameterization of such high-dimensional data, transform the data into their low-dimensional representations (features) preserving certain chosen subject-driven data properties [3, 4].

The most popular model of high-dimensional data, which occupy a small part of observation space R^p, is Manifold model in accordance with which the data lie on or near an unknown Data manifold (DM) of known lower dimensionality q < p embedded in an ambient high-dimensional space R^p (Manifold assumption [5] about high-dimensional data). Typically, this assumption is satisfied for 'real-world' high-dimensional data obtained from 'natural' sources.

F. Schwenker et al. (Eds.): ANNPR 2016, LNAI 9896, pp. 55–67, 2016.
DOI: 10.1007/978-3-319-46182-3_5

Dimensionality reduction under the manifold assumption about processed data are usually referred to as the Manifold learning [6, 7] whose goal is constructing a low-dimensional parameterization of the DM (global low-dimensional coordinates on the DM) from a finite dataset sampled from the DM. This parameterization produces an Embedding mapping from the DM to low-dimensional Feature space that should preserve specific properties of the DM determined by chosen optimized cost function which defines an 'evaluation measure' for the dimensionality reduction and reflects the desired properties of the initial data which should be preserved in their features.

Most manifold learning algorithms include the solution of large-dimensional global optimization problems and, thus, are computationally expensive. The incremental versions of many popular algorithms (Locally Linear Embedding, Isomap, Laplacian Eigenmaps, Local Tangent Space Alignment, Hessian Eigenmaps, etc. [6, 7]), which reduce their computational complexity, were developed [8–17].

The manifold learning algorithms are usually used as a first key step in solution of machine learning tasks: the low-dimensional features are used in reduced learning procedures instead of initial high-dimensional data avoiding the curse of dimensionality [18]: 'dimensionality reduction may be necessary in order to discard redundancy and reduce the computational cost of further operations' [19]. If the low-dimensional features preserve only specific properties of data, then substantial data losses are possible when using the features instead of the initial data. To prevent these losses, the features should preserve as much as possible available information contained in the high-dimensional data [20]; it means the possibility for recovering the initial data from their features with small reconstruction error. Such Manifold reconstruction algorithms result in both the parameterization and recovery of the unknown DM [21].

Mathematically [22], a 'preserving the important information of the DM' means that manifold learning algorithms should 'recover the geometry' of the DM, and 'the information necessary for reconstructing the geometry of the manifold is embodied in its Riemannian metric (tensor)' [23]. Thus, the learning algorithms should accurately recover Riemannian data manifold that is the DM equipped by Riemannian tensor.

Certain requirement to the recovery follows from the necessity of providing a good generalization capability of the manifold reconstruction algorithms and preserving local structure of the DM: the algorithms should preserve a differential structure of the DM providing proximity between tangent spaces to the DM and Recovered data manifold (RDM) [24]. In the Manifold theory [23, 25], the set composed of the manifold points equipped by tangent spaces at these points is called the Tangent bundle of the manifold; thus, a reconstruction of the DM, which ensures accurate reconstruction of its tangent spaces too, is referred to as the Tangent bundle manifold learning.

Earlier proposed geometrically motivated Grassmann&Stiefel Eigenmaps algorithm (GSE) [24, 26] solves the Tangent bundle manifold learning and recovers Riemannian tensor of the DM; thus, it solves the Riemannian manifold recovery problem.

The GSE, like most manifold learning algorithms, includes the solution of large-dimensional global optimization problems and, thus, is computationally expensive.

In this paper, we propose an incremental version of the GSE that reduces the solution of the computationally expensive global optimization problems to the solution of a sequence of local optimization problems solved in explicit form.

The rest of the paper is organized as follows. Section 2 contains strong definition of the Tangent bundle manifold learning and describes main ideas realized in its GSE-solution. The proposed incremental version of the GSE is presented in Sect. 3.

2 Tangent Bundle Manifold Learning

2.1 Definitions and Assumptions

Consider unknown q-dimensional Data manifold with known intrinsic dimension q

$$\mathbf{M} = \{X = g(y) \in R^p : \ y \in \mathbf{Y} \subset R^q\}$$

covered by a single chart g and embedded in an ambient p-dimensional space R^p, q < p. The chart g is one-to-one mapping from open bounded Coordinate space $\mathbf{Y} \subset R^q$ to the manifold $\mathbf{M} = g(\mathbf{Y})$ with differentiable inverse mapping $h_g(X) = g^{-1}(X)$ whose values $y = h_g(X) \in \mathbf{Y}$ give low-dimensional coordinates (representations, features) of high-dimensional manifold-valued data X.

If the mappings $h_g(X)$ and g(y) are differentiable and $J_g(y)$ is p × q Jacobian matrix of the mapping g(y), than q-dimensional linear space $L(X) = \mathrm{Span}(J_g(h_g(X)))$ in R^p is tangent space to the DM $\mathbf{M}$ at the point $X \in \mathbf{M}$; hereinafter, Span(H) is linear space spanned by columns of arbitrary matrix H.

The tangent spaces can be considered as elements of the Grassmann manifold Grass(p, q) consisting of all q-dimensional linear subspaces in R^p.

Standard inner product in R^p induces an inner product on the tangent space L(X) that defines Riemannian metric (tensor) Δ(X) in each manifold point $X \in \mathbf{M}$ smoothly varying from point to point; thus, the DM $\mathbf{M}$ is a Riemannian manifold ($\mathbf{M}$, Δ).

Let $\mathbf{X}_n = \{X_1, X_2, \ldots, X_n\}$ be a dataset randomly sampled from the DM $\mathbf{M}$ according to certain (unknown) probability measure whose support coincides with $\mathbf{M}$.

2.2 Tangent Bundle Manifold Learning Definition

Conventional manifold learning problem, called usually Manifold embedding problem [6, 7], is to construct a low-dimensional parameterization of the DM from given sample $\mathbf{X}_n$, which produces an Embedding mapping $h : \mathbf{M} \subset R^p \to \mathbf{Y}_h = h(\mathbf{M}) \subset R^q$ from the DM $\mathbf{M}$ to the Feature space (FS) $\mathbf{Y}_h \subset R^q$, q < p, which preserves specific chosen properties of the DM.

Manifold reconstruction algorithm, which provides additionally a possibility of accurate recovery of original vectors X from their low-dimensional features y = h(X), includes a constructing of a Recovering mapping g(y) from the FS $\mathbf{Y}_h$ to the Euclidean space R^p in such a way that the pair (h, g) ensures approximate equalities

$$r_{h,g}(X) \equiv g(h(X)) \approx X \text{ for all points } X \in \mathbf{M}. \tag{1}$$

The mappings (h, g) determine q-dimensional Recovered data manifold

$$\mathbf{M}_{h,g} = r_{h,g}(\mathbf{M}) = \{r_{h,g}(X) \in R^p : X \in \mathbf{M}\} = \{X = g(y) \in R^p : y \in \mathbf{Y}_h \subset R^q\} \quad (2)$$

which is embedded in the ambient space R^p, covered by a single chart g, and consists of all recovered values $r_{h,g}(X)$ of manifold points $X \in \mathbf{M}$. Proximities (1) imply manifold proximity $\mathbf{M}_{h,g} \approx \mathbf{M}$ meaning a small Hausdorff distance $d_H(\mathbf{M}_{h,g}, \mathbf{M})$ between the DM $\mathbf{M}$ and RDM $\mathbf{M}_{h,g}$ due inequality $d_H(\mathbf{M}_{h,g}, \mathbf{M}) \le \sup_{X \in \mathbf{M}} |r_{h,g}(X) - X|$.

Let $G(y) = J_g(y)$ be p × q Jacobian matrix of the mapping g(y) which determines q-dimensional tangent space $L_{h,g}(X)$ to the RDM $\mathbf{M}_{h,g}$ at the point $r_{h,g}(X) \in \mathbf{M}_{h,g}$:

$$L_{h,g}(X) = \mathrm{Span}(G(h(X))) \quad (3)$$

Tangent bundle manifold learning problem is to construct the pair (h, g) of mappings h and g from given sample $\mathbf{X}_n$ ensuring both the proximities (1) and proximities

$$L_{h,g}(X) \approx L(X) \text{ for all points } X \in \mathbf{M}; \quad (4)$$

proximities (4) are defined with use certain chosen metric on the Grass(p, q).

The matrix G(y) determines also metric tensor $\Delta_{h,g}(X) = G^T(h(X)) \times G(h(X))$ on the RMD $\mathbf{M}_{h,g}$ which is q × q matrix consisting of inner products $\{(G_i(h(X)), G_j(h(X)))\}$ between i^{th} and j^{th} columns $G_i(h(X))$ and $G_j(h(X))$ of the matrix G(h(X)). Thus, the pair (h, g) determines Recovered Riemannian manifold $(\mathbf{M}_{h,g}, \Delta_{h,g})$ that accurately approximates initial Riemannian data manifold $(\mathbf{M}, \Delta)$.

2.3 Grassmann&Stiefel Eigenmaps: An Approach

Grassmann&Stiefel Eigenmaps algorithm gives the solution to the Tangent bundle manifold learning problem and consists of three successively performed parts: Tangent manifold learning, Manifold embedding, and Manifold recovery.

Tangent Manifold Learning Part. A sample-based family $\mathbf{H}$ consisting of p × q matrices H(X) smoothly depending on $X \in \mathbf{M}$ is constructed to meet relations

$$L_H(X) \equiv \mathrm{Span}(H(X)) \approx L(X) \text{ for all } X \in \mathbf{M} \quad (5)$$

in certain chosen metric on the Grassmann manifold. In next steps, the mappings h and g will be built in such a way that both the equalities (1) and

$$G(h(X)) \approx H(X) \text{ for all points } X \in \mathbf{M} \quad (6)$$

are fulfilled. Hence, linear space $L_H(X)$ (5) approximates the tangent space $L_{h,g}(X)$ (3) to the RDM $\mathbf{M}_{h,g}$ at the point $r_{h,g}(X)$.

Manifold Embedding Part. Given the family $\mathbf{H}$ already constructed, the embedding mapping y = h(X) is constructed as follows. The Taylor series expansions

$$g(h(X')) - g(h(X)) \approx G(h(X)) \times (h(X') - h(X)) \tag{7}$$

of the mapping g at near points h(X′), h(X) ∈ $\mathbf{Y}_h$, under the desired approximate equalities (1), (6) for the mappings h and g to be specified further, imply equalities:

$$X' - X \approx H(X) \times (h(X') - h(X)) \tag{8}$$

for near points X, X′ ∈ **M**. These equations considered further as regression equations allow constructing the embedding mapping h and the FS $\mathbf{Y}_h = h(\mathbf{M})$.

Manifold Reconstruction Step. Given the family **H** and mapping h(X) already constructed, the expansion (7), under the desired proximities (1) and (6), implies relation

$$g(y) \approx X + H(X) \times (y - h(X)) \tag{9}$$

for near points y, h(X) ∈ $\mathbf{Y}_h$ which is used for constructing the mapping g.

2.4 Grassmann&Stiefel Eigenmaps: Some Details

Details of the GSE are presented below. The numbers $\{\varepsilon_i > 0\}$ denote the algorithms parameters whose values are chosen depending on the sample size n ($\varepsilon_i = \varepsilon_{i,n}$) and tend to zero as n → ∞ with rate $O(n^{-1/(q+2)})$.

Step S1: Neighborhoods (Construction and Description). The necessary preliminary calculations are performed at first step S1.

Euclidean Kernel. Introduce Euclidean kernel $K_E(X, X') = I\{|X' - X| < \varepsilon_1\}$ on the DM at points X, X′ ∈ **M**, here $I\{\cdot\}$ is indicator function.

Grassmann Kernel. An applying the Principal Component Analysis (PCA) [27] to the points from the set $U_n(X, \varepsilon_1) = \{X' \in \mathbf{X}_n\colon |X' - X| < \varepsilon_1\} \cup \{X\}$, results in p × q orthogonal matrix $Q_{PCA}(X)$ whose columns are PCA principal eigenvectors corresponding to the q largest PCA eigenvalues. These matrices determine q-dimensional linear spaces $L_{PCA}(X) = \mathrm{Span}(Q_{PCA}(X))$ in R^p, which, under certain conditions, approximate the tangent spaces L(X):

$$L_{PCA}(X) \approx L(X). \tag{10}$$

In what follows, we assume that sample size n is large enough to ensure a positive value of the q^{th} PCA-eigenvalue in sample points and provide proximities (10). To provide trade-off between ‘statistical error’ (depending on number n(X) of sample points in set $U_n(X, \varepsilon_1)$) and ‘curvature error’ (caused by deviation of the manifold-valued points from the ‘assumed in the PCA’ linear space) in (10), ball radius ε_1 should tend to 0 as n → ∞ with rate $O(n^{-1/(q+2)})$, providing, with high probability, the order $O(n^{-1/(q+2)})$ for the error in (10) [28, 29]; here ‘an event occurs with high probability' means that its probability exceeds the value $(1 - C_\alpha/n^\alpha)$ for any n and $\alpha > 0$, and the constant C_α depends only on α.

Grassmann kernel $K_G(X, X')$ on the DM at points $X, X' \in \mathbf{M}$ is defined as

$$K_G(X, X') = I\{d_{BC}(L_{PCA}(X), L_{PCA}(X')) < \varepsilon_2\} \times K_{BC}(L_{PCA}(X), L_{PCA}(X'))$$

with use Binet-Cauchy kernel $K_{BC}(L_{PCA}(X), L_{PCA}(X')) = \mathrm{Det}^2[S(X, X')]$ and Binet-Cauchy metric $d_{BC}(L_{PCA}(X), L_{PCA}(X')) = \{1 - \mathrm{Det}^2[S(X, X')]\}^{1/2}$ on the Grassmann manifold Grass(p, q) [30, 31], here $S(X, X') = Q^T_{PCA}(X) \times Q_{PCA}(X')$.

Orthogonal p × p matrix $\pi_{PCA}(X) = Q_{PCA}(X) \times Q^T_{PCA}(X)$ is projector onto linear space $L_{PCA}(X)$ which approximates projection matrix $\pi(X)$ onto the tangent space $L(X)$.

Aggregate Kernel. Introduce the kernel $K(X, X') = K_E(X, X') \times K_G(X, X')$, which reflects not only geometrical nearness between points X and X′ but also nearness between the linear spaces $L_{PCA}(X)$ and $L_{PCA}(X')$ (and, thus (10), nearness between the tangent spaces $L(X)$ and $L(X')$), as a product of the Euclidean and Grassmann kernels.

Step S2: Tangent Manifold Learning. The matrices H(X) will be constructed to meet the equalities $L_H(X) = L_{PCA}(X)$ for all points $X \in \mathbf{M}$ that implies a representation

$$H(X) = Q_{PCA}(X) \times v(X), \tag{11}$$

in which q × q matrices v(x) should provide a smooth depending H(X) on point X.

At first, the p × q matrices $\{H_i = Q_{PCA}(X_i) \times v_i\}$ are constructed to minimize a form

$$\Delta_{H,n} = \frac{1}{2}\sum\nolimits_{i,j=1}^{n} K(X_i, X_j) \times ||H_i - H_j||_F^2 \tag{12}$$

over q × q matrices $v_1, v_2, \ldots, v_n$, under normalizing constraint

$$\sum\nolimits_{i=1}^{n} K(X_i) \times (H_i^T \times H_i) = \sum\nolimits_{i=1}^{n} K(X_i) \times (v_i^T \times v_i) = K \times I_q \tag{13}$$

used to avoid a degenerate solution; here $K(X) = \sum\nolimits_{j=1}^{n} K(X, X_j)$ and $K = \sum\nolimits_{i=1}^{n} K(X_i)$.

The quadratic form (12) and the constraint (13) take the forms $(K - \mathrm{Tr}(\mathbf{V}^T \times \Phi \times \mathbf{V}))$ and $\mathbf{V}^T \times F \times \mathbf{V} = K \times I_q$, respectively, here $\mathbf{V}$ is (nq) × q matrix whose transpose consists of the consecutively written transposed q × q matrices $v_1, v_2, \ldots, v_n$, $\mathbf{\Phi} = ||\Phi_{ij}||$ and $\mathbf{F} = ||F_{ij}||$ are nq × nq matrices consisting, respectively, of q × q matrices

$$\{\Phi_{ij} = K(X_i, X_j) \times S(X_i, X_j)\} \text{ and } \{F_{ij} = \delta_{ij} \times K(X_i) \times I_q\}.$$

Thus, a minimization (12), (13) is reduced to the generalized eigenvector problem

$$\mathbf{\Phi} \times \mathbf{V} = \lambda \times \mathbf{F} \times \mathbf{V}, \tag{14}$$

and (nq) × q matrix **V**, whose columns V_1, V_2, …, $V_q \in R^{nq}$ are orthonormal eigenvectors corresponding to the q largest eigenvalues in the problem (14), determines the required q × q matrices v_1, v_2, …, v_n.

The value H(X) (11) at arbitrary point X ∈ **M** is chosen to minimize a form

$$d_{H,n}(H) = \sum\nolimits_{j=1}^{n} K(X, X_j) \times ||Q_{PCA}(X) \times v(X) - H_j||_F^2 \tag{15}$$

over v(X) under condition Span(H) = L_{PCA}(X), whose solution is

$$H(X) = Q_{PCA}(X) \times v(X) = Q_{PCA}(X) \times \frac{1}{K(X)} \sum\nolimits_{j=1}^{n} K\left(X, X_j\right) \times S\left(X, X_j\right) \times v_j. \tag{16}$$

It follows from above formulas that the q × p matrix

$$G_h(X) = H^{-}(X) \times \pi_{PCA}(X) = v^{-1}(X) \times Q_{PCA}^{T}(X)$$

estimates Jacobian matrix J_h(X) of Embedding mapping h(X) constructed afterward, here H^{-}(X) is q × p pseudoinverse Moore-Penrose matrix of p × q matrix H(X) [32].

Step S3: Manifold Embedding. Embedding mapping h(X) with already known (estimated) Jacobian G_h(X) is constructed to meet equalities (8) written for all pairs of near points X, X′ ∈ **M** which can be considered as regression equations.

At first, the vector set $\{h_1, h_2, …, h_n\} \subset R^q$ is computed as a standard least squares solution in this regression problem by minimizing the residual

$$\Delta_{h,n} = \sum\nolimits_{i,j=1}^{n} K\left(X_i, X_j\right) \times \left|X_j - X_i - H_i \times (h_j - h_i)\right|^2 \tag{17}$$

over the vectors h_1, h_2, …, h_n under normalizing condition $h_1 + h_2 + … + h_n = \mathbf{0}$.

Then, considering the obtained vectors $\{h_j\}$ as preliminary values of the mapping h(X) at sample points, choose the value

$$h(X) = \frac{1}{K(X)} \sum\nolimits_{i=1}^{n} K(X, X_i) \times \{h_i + G_h(X) \times (X - X_i)\} \tag{18}$$

for arbitrary point X ∈ **M** as a result of minimizing over h the residual

$$d_{h,n}(h) = \sum\nolimits_{j=1}^{n} K\left(X, X_j\right) \times \left|X_j - X - H(X) \times (h_j - h)\right|^2. \tag{19}$$

The mapping (18) determines Feature sample $\mathbf{Y}_{h,n} = \{y_{h,i} = h(X_i), i = 1, 2, …, n\}$.

Step S4: Manifold Recovery. A kernel on the FS $\mathbf{Y}_h$ and, then, the recovering mapping g(y) and its Jacobian matrix G(y) are constructed in this step.

Kernel on the Feature Space. It follows from (8) that proximities

$$|X - X_i| \approx d(y,\ y_{h,i}) = \{(y - y_{h,i})^T \times [H^T(X_i) \times H(X_i)] \times (y - y_{h,i})\}^{1/2}$$

hold true for near points $y = h(X)$ and $y_{h,i} \in \mathbf{Y}_{h,n}$. Let $u_E(y, \varepsilon_1) = \{y_{h,i}: d(y, y_{h,i}) < \varepsilon_1\}$ be a neighborhood of the feature $y = h(X)$ consisting of sample features which are images of the sample points from $U_n(X, \varepsilon_1)$.

An applying the PCA to the set $h^{-1}(u_E(y, \varepsilon_1)) = \{X_i: y_{h,i} \in u_E(y, \varepsilon_1)\}$ results in the linear space $L_{PCA^*}(y) \in \mathrm{Grass}(p, q)$ which meets proximity $L_{PCA^*}(h(X)) \approx L_{PCA}(X)$.

Introduce feature kernel $k(y, y_{h,i}) = I\{y_{h,i} \in u_E(y, \varepsilon_1)\} \times K_G(L_{PCA^*}(y), L_{PCA^*}(y_{h,i}))$ that meets equalities $k(h(X), h(X')) \approx K(X, X')$ for near points $X \in \mathbf{M}$ and $X' \in \mathbf{X}_n$.

Constructing the Recovering Mapping and its Jacobian. The matrix G(y), which should meet both the conditions (6) and constraint $\mathrm{Span}(G(y)) = L_{PCA^*}(y)$, is chosen by minimizing quadratic form $\sum_{j=1}^{n} k\left(y, y_{h,j}\right) \times ||G(y) - H_j||_F^2$ over G, that results in

$$G(y) = \pi^*(y) \times \frac{1}{k(y)} \sum_{j=1}^{n} k\left(y, y_{h,j}\right) \times H_j, \tag{20}$$

here $\pi^*(y)$ is the projector onto the linear space $L_{PCA^*}(y)$ and $k(y) = \sum_{j=1}^{n} k\left(y, y_{h,j}\right)$.

Based on expansions (9) written for features $y_{h,j} \in u_E(y, \varepsilon_1)$, g(y) is chosen by minimizing quadratic form $\sum_{j=1}^{n} k\left(y, y_{h,j}\right) \times \left|X_j - g(y) - G(y) \times (y_{h,j} - y)\right|^2$ over g, thus

$$g(y) = \frac{1}{k(y)} \sum_{j=1}^{n} k\left(y, y_{h,j}\right) \times \{X_j + G(y) \times (y - y_{h,j})\}. \tag{21}$$

The constructed mappings (18), (21) allow recovering the DM **M** and its tangent spaces L(X) by the formulas (2) and (4).

2.5 Grassmann&Stiefel Eigenmaps: Some Properties

Under asymptotic n → ∞, when $\varepsilon_1 = O(n^{-1/(q+2)})$, relation $d_H(\mathbf{M}_{h,g}, \mathbf{M}) = O(n^{-2/(q+2)})$ hold true uniformly in points $X \in \mathbf{M}$ with high probability [33]. This rate coincides with the asymptotically minimax lower bound for the Hausdorff distance $d_H(\mathbf{M}_{h,g}, \mathbf{M})$ [34]; thus, the RDM $\mathbf{M}_{h,g}$ estimates the DM **M** with optimal rate of convergence.

The main computational complexity of the GSE-algorithm is in the second and third steps, in which global high-dimensional optimization problems are solved.

First problem is generalized eigenvector problem (14) with nq × nq matrices **F** and **Φ**. This problem is solved usually with use the Singular value decomposition (SVD) [32] whose computational complexity is $O(n^3)$ [35].

Second problem is regression problem (17) for nq-dimensional estimated vector. This problem is reduced to the solution of the linear least-square normal equations with nq × nq matrix whose computational complexity is $O(n^3)$ also [32].

Thus, the GSE has total computational complexity $O(n^3)$ and is computationally expensive under large sample size n.

3 Incremental Grassmann&Stiefel Eigenmaps

The incremental version of the GSE divides the most computationally expensive generalized eigenvector and regression problems into n local optimization procedures, each time k solved in explicit form for one new variable (matrix H_k and feature h_k) only, k = 1, 2, …, n.

The proposed incremental version includes an additional preliminary step $S1^+$ performed after the Step S1, in which a weighted undirected sample graph $\Gamma(\mathbf{X}_n)$ consisting of the sample points $\{X_i\}$ as nodes is constructed and the shortest ways between arbitrary node chosen as an origin of the graph and all the other nodes are calculated.

The second and third steps S2 and S3 are replaced by common incremental step S2–3 in which the matrices $\{H_k\}$ and features $\{h_k\}$ are computed sequentially at the graph nodes, moving along the shortest paths starting from the chosen origin of the graph. Step S4 in the GSE remains unchanged in the incremental version.

3.1 Step $S1^+$: Sample Graph

Introduce a weighted undirected sample graph $\Gamma(\mathbf{X}_n)$ consisting of the sample points $\{X_i\}$ as nodes. The edges in $\Gamma(\mathbf{X}_n)$ connect the nodes X_i and X_j if and only when $K(X_i, X_j) > 0$; the lengths of such edge (X_i, X_j) equal to $|X_i - X_j|/K(X_i, X_j)$.

Choose arbitrary node $X_{(1)} \in \Gamma(X_n)$ as an origin of the graph. Using the Dijksra algorithm [36], compute the shortest paths between the chosen node and all the other nodes $X_{(2)}, X_{(3)}, \ldots, X_{(n)}$ writing in ascending order of the lengths of the shortest paths from the origin $X_{(1)}$. Denote Γ_k a subgraph consisting of the nodes $\{X_{(1)}, X_{(2)}, \ldots, X_{(k)}\}$ and connected them edges.

Note. The origin $X_{(1)}$ can be chosen as a node with minimal eccentricity; an eccentricity of some node equals to maximum of lengths of the shortest paths between the node under consideration and all the other nodes. But a calculation of the shortest ways between all nodes in the graph $\Gamma(\mathbf{X}_n)$, which should be computed for this construction, require n-fold applying of the Dijksra algorithm.

3.2 Step S2–3: Incremental Tangent Manifold Learning and Manifold Embedding

Incremental version computes sequentially the matrices H(X) and h(X) at the points $X_{(1)}, X_{(2)}, \ldots, X_{(n)}$, starting from matrix $H_{(1)}$ and $h_{(1)}$ (initialization). Thus, step S2–3 consists of n substeps $\{S2\text{–}3_k, k = 1, 2, \ldots, n\}$ in which initialization substep is

Initialization substep S2–3$_1$. Put $v_{(1)} = I_q$ and $h_{(1)} = \mathbf{0}$; thus, $H(X_{(1)}) = Q_{PCA}(X_{(1)})$.

At the k-th substep S2–3$_k$, $k > 1$, when the matrices $H_{(j)}$, $j < k$, have already computed, quadratic form $\Delta_{H,k}$, similar to the form (12) but written only for the points X_i, $X_j \in \Gamma_k$, is minimized over single unknown matrix $H_{(k)} = Q_{PCA}(X_{(k)}) \times v_{(k)}$. This problem, in turn, is reduced to a minimization over $v_{(k)}$ of the form $d_{H,k}(H_{(k)})$, similar to the form $d_{H,n}(H_{(k)})$ (15) but written only for points $X_j \in \Gamma_{k-1}$. Its solution $v_{(k)}$, which is similar to the solution (16), is written in explicit form.

Let $\Delta_{h,k}$ be a quadratic form, similar to the form $\Delta_{h,n}$ (17) but written only for points X_i, $X_j \in \Gamma_k$. The value $h_{(k)}$, under the already computed values $h_{(j)}$, $j < k$, is calculated by minimizing the quadratic form $\Delta_{h,k}$ over single vector $h_{(k)}$. This problem, in turn, is reduced to a minimization over $h_{(k)}$ the form $d_{h,k}(h_{(k)})$, similar to the form $d_{h,n}(h_{(k)})$ (19) but written only for points $X_j \in \Gamma_{k-1}$; its solution, similar to the solution (18), is written in explicit form also.

Thus, the substeps S2–3$_k$, $k = 1, 2, \ldots, n$, are:

Typical substep S2–3$_k$, $1 < k \le n$. Given $\{(H_{(j)}, h_{(j)}), j < k\}$ already obtained, put

$$H_{(k)} = Q_{PCA}\left(X_{(k)}\right) \times v_{(k)} = Q_{PCA}\left(X_{(k)}\right) \times \frac{\sum_{j<k} K\left(X_{(k)}, X_{(j)}\right) \times S\left(X_{(k)}, X_{(j)}\right) \times v_{(j)}}{\sum_{j<k} K\left(X_{(k)}, X_{(j)}\right)}, \tag{22}$$

$$h_{(k)} = \frac{\sum_{j<k} K\left(X_{(k)}, X_{(j)}\right) \times \left\{h_{(j)} + v_{(k)}^{-1} \times Q_{PCA}^{T}\left(X_{(k)}\right) \times \left(X_{(k)} - X_{(j)}\right)\right\}}{\sum_{j<k} K\left(X_{(k)}, X_{(j)}\right)}. \tag{23}$$

Given $\{(H_{(k)}, h_{(k)}), k = 1, 2, \ldots, n\}$, the value $H(X) = Q_{PCA}(X) \times v(X)$ and $h(X)$ at arbitrary point $X \in \mathbf{M}$ are calculated with use formulas (16) and (18), respectively.

3.3 Incremental GSE: Properties

Computational Complexity. Incremental GSE works mainly with sample data lying in a neighborhood of some point X contained in ε_1-ball $U_n(X, \varepsilon_1)$ centered at X. The number n(X) of sample points fallen into this ball, under $\varepsilon_1 = \varepsilon_{1,n} = O(n^{-1/(q+2)})$, with high probability equals to $n \times O(n^{-q/(q+2)}) = O(n^{2/(q+2)})$ uniformly on $X \in \mathbf{M}$ [37].

The sample graph $\Gamma(\mathbf{X}_n)$ consists of $V = n$ nodes and E edges connecting the graph nodes $\{X_k\}$. Each node X_k is connected with no more than $n(X_k)$ other nodes, thus $E < 0.5 \times n \times \max_k n(X_k) = O(n^{(q+4)/(q+2)})$ and, hence, $\Gamma(\mathbf{X}_n)$ is sparse graph.

The running time of the Dijksra algorithm (Step S1$^+$), which computes the shortest paths in the sparse connected graph $\Gamma(\mathbf{X}_n)$, is $O(E \times \ln V) = O(n^{(q+4)/(q+2)} \times \ln n)$ in the worst case; the Fibonacci heap improves this rate to $O(E + V \times \ln V) = O(n^{(q+4)/(q+2)})$ [38].

The running time of k-th Step S2–3$_k$ (formulas (22) and (23)) is proportional to $n(X_k)$; thus total running time of the Step S2–3 is n × $O(n^{-q/(q+2)}) = O(n^{(q+4)/(q+2)})$.

Therefore, the running time of the incremental version of the GSE is $O(n^{(q+4)/(q+2)})$, in contrast to the running time $O(n^3)$ of the initial GSE.

Accuracy. It follows from (18), (21) that $X - r_{h,g}(X) \approx \left(\pi^T_{PCA}(X) \times e(X)\right) \times |\delta(X)|$, in which $\delta(X) = X - \frac{1}{K(X)}\sum_{i=1}^{n} K(X, X_i) \times X_i$ and $e(X) = \delta(X)/|\delta(X)|$. The first and second multipliers are majorized by the PCA-error in (10) and $\varepsilon_{1,n}$, respectively, each of them has rate $O(n^{-1/(q+2)})$. Thus, reconstruction error $(X - r_{h,g}(X))$ in the incremental GSE has the same asymptotically optimal rate $O(n^{-2/(q+2)})$ as in the original GSE.

4 Conclusion

The incremental version of the Grassmann&Stiefel Eigenmaps algorithm, which constructs the low-dimensional representations of high-dimensional data with asymptotically minimal reconstruction error, is proposed. This version has the same optimal convergence rate $O(n^{-2/(q+2)})$ of the reconstruction error and a significantly smaller computational complexity on the sample size n: running time $O(n^{(q+4)/(q+2)})$ of the incremental version in contrast to $O(n^3)$ of the original algorithm.

Acknowledgments. This work is partially supported by the Russian Foundation for Basic Research, research project 16-29-09649 ofi-m.

References

1. Donoho, D.L.: High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality. Lecture at the “Mathematical Challenges of the 21st Century” Conference of the AMS, Los Angeles (2000). http://www-stat.stanford.edu/donoho/Lectures/AMS2000/AMS2000.html
2. Verleysen, M.: Learning high-dimensional data. In: Ablameyko, S., Goras, L., Gori, M., Piuri, V. (eds.) Limitations and Future Trends in Neural Computation. NATO Science Series, III: Computer and Systems Sciences, vol. 186, pp. 141–162. IOS Press, Netherlands (2003)
3. Bengio, Y., Courville, A., Vincent, P.: Representation Learning: A Review and New Perspectives, pp. 1–64 (2014). arXiv:1206.5538v3[cs.LG]. Accessed 23 Apr 2014
4. Bernstein, A., Kuleshov, A.: Low-dimensional data representation in data analysis. In: El Gayar, N., Schwenker, F., Suen, C. (eds.) ANNPR 2014. LNCS, vol. 8774, pp. 47–58. Springer, Heidelberg (2014)
5. Seung, H.S., Lee, D.D.: The manifold ways of perception. Science **290**(5500), 2268–2269 (2000)
6. Huo, X., Ni, X., Smith, A.K.: Survey of manifold-based learning methods. In: Liao, T.W., Triantaphyllou, E. (eds.) Recent Advances in Data Mining of Enterprise Data, pp. 691–745. World Scientific, Singapore (2007)
7. Ma, Y., Fu, Y. (eds.): Manifold Learning Theory and Applications. CRC Press, London (2011)

8. Law, M.H.C., Jain, A.K.: Nonlinear manifold learning for data stream. In: Berry, M., Dayal, U., Kamath, C., Skillicorn, D. (eds.) Proceedings of the 4th SIAM International Conference on Data Mining, Like Buena Vista, Florida, USA, pp. 33–44 (2004)
9. Law, M.H.C., Jain, A.K.: Incremental nonlinear dimensionality reduction by manifold learning. IEEE Trans. Pattern Anal. Mach. Intell. **28**(3), 377–391 (2006)
10. Gao, X., Liang, J.: An improved incremental nonlinear dimensionality reduction for isometric data embedding. Inf. Process. Lett. **115**(4), 492–501 (2015)
11. Saul, L.K., Roweis, S.T.: Think globally, fit locally: unsupervised learning of low dimensional manifolds. J. Mach. Learn. Res. **4**, 119–155 (2003)
12. Kouropteva, O., Okun, O., Pietikäinen, M.: Incremental locally linear embedding algorithm. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) SCIA 2005. LNCS, vol. 3540, pp. 521–530. Springer, Heidelberg (2005)
13. Kouropteva, O., Okun, O., Pietikäinen, M.: Incremental locally linear embedding. Pattern Recogn. **38**(10), 1764–1767 (2005)
14. Schuon, S., Đurković, M., Diepold, K., Scheuerle, J., Markward, S.: Truly incremental locally linear embedding. In: Proceedings of the CoTeSys 1st International Workshop on Cognition for Technical Systems, 6–8 October 2008, Munich, Germany, p. 5 (2008)
15. Jia, P., Yin, J., Huang, X., Hu, D.: Incremental Laplacian eigenmaps by preserving adjacent information between data points. Pattern Recogn. Lett. **30**(16), 1457–1463 (2009)
16. Liu, X., Yin, J.-w., Feng, Z., Dong, J.: Incremental manifold learning via tangent space alignment. In: Schwenker, F., Marinai, S. (eds.) ANNPR 2006. LNCS (LNAI), vol. 4087, pp. 107–121. Springer, Heidelberg (2006)
17. Abdel-Mannan, O., Ben Hamza, A., Youssef, A.: Incremental line tangent space alignment algorithm. In: Proceedings of 2007 Canadian Conference on Electrical and Computer Engineering (CCECE 2007), 22–26 April 2007, Vancouver, pp. 1329–1332. IEEE (2007)
18. Kuleshov, A., Bernstein, A.: Manifold learning in data mining tasks. In: Perner, P. (ed.) MLDM 2014. LNCS, vol. 8556, pp. 119–133. Springer, Heidelberg (2014)
19. Lee, J.A., Verleysen, M.: Nonlinear Dimensionality Reduction. Information Science and Statistics. Springer, New York (2007)
20. Lee, J.A., Verleysen, M.: Quality assessment of dimensionality reduction: rank-based criteria. Neurocomputing **72**(7–9), 1431–1443 (2009)
21. Bernstein, A.V., Kuleshov, A.P.: Data-based manifold reconstruction via tangent bundle manifold learning. In: ICML-2014, Topological Methods for Machine Learning Workshop, Beijing, 25 June 2014. http://topology.cs.wisc.edu/KuleshovBernstein.pdf
22. Perrault-Joncas, D., Meilă, M.: Non-linear Dimensionality Reduction: Riemannian Metric Estimation and the Problem of Geometric Recovery, pp. 1–25 (2013). arXiv:1305.7255v1 [stat.ML]. Accessed 30 May 2013
23. Jost, J.: Riemannian Geometry and Geometric Analysis, 6th edn. Springer, Berlin (2011)
24. Bernstein, A.V., Kuleshov, A.P.: Manifold learning: generalizing ability and tangent proximity. Int. J. Softw. Inf. **7**(3), 359–390 (2013)
25. Lee, J.M.: Manifolds and Differential Geometry. Graduate Studies in Mathematics, vol. 107. American Mathematical Society, Providence (2009)
26. Bernstein, A.V., Kuleshov, A.P.: Tangent bundle manifold learning via Grassmann&Stiefel eigenmaps, pp. 1–25, December 2012. arXiv:1212.6031v1[cs.LG]
27. Jollie, T.: Principal Component Analysis. Springer, New York (2002)
28. Singer, A., Wu, H.-T.: Vector diffusion maps and the connection Laplacian. Commun. Pure Appl. Math. **65**(8), 1067–1144 (2012)
29. Tyagi, H., Vural, E., Frossard, P.: Tangent space estimation for smooth embeddings of Riemannian manifold, pp. 1–35 (2013). arXiv:1208.1065v2[stat.CO]. Accessed 17 May 2013

30. Hamm, J., Daniel, L.D.: Grassmann discriminant analysis: a unifying view on subspace-based learning. In: Proceedings of the 25th International Conference on Machine Learning (ICML 2008), pp. 376–383 (2008)
31. Wolf, L., Shashua, A.: Learning over sets using kernel principal angles. J. Mach. Learn. Res. **4**, 913–931 (2003)
32. Golub, G.H., Van Loan, C.F.: Matrix Computation, 3rd edn. Johns Hopkins University Press, Baltimore (1996)
33. Kuleshov, A., Bernstein, A., Yanovich, Y.: Asymptotically optimal method in manifold estimation. In: Abstracts of the XXIX-th European Meeting of Statisticians, 20–25 July 2013, Budapest, Hungary, p. 325 (2013). http://ems2013.eu/conf/upload/BEK086_006.pdf
34. Genovese, C.R., Perone-Pacifico, M., Verdinelli, I., Wasserman, L.: Minimax manifold estimation. J. Mach. Learn. Res. **13**, 1263–1291 (2012)
35. Trefethen, L.N.: Bau III, David: Numerical Linear Algebra. SIAM, Philadelphia (1997)
36. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to Algorithms. MIT Press, Cambridge (2001)
37. Yanovich, Y.: Asymptotic properties of local sampling on manifolds. J. Math. Stat. (2016)
38. Fredman, M.L., Tarjan, R.E.: Fibonacci heaps and their uses in improved network optimization algorithms. J. Assoc. Comput. Mach. **34**(3), 596–615 (1987)