# Subspace Interpolation and Indexing on Stiefel and Grassmann Manifolds as a Lightweight Inference Engine

Wenqing Hu

Missouri University of Science and Technology
(formerly University of Missouri, Rolla)

October 14, 2025

# Thanks to the Organization Committee!

# Research Collaboration

This presentation is based on research in collaboration with:

- **PhD. Student Jiali Zhang** from Missouri University of Science and Technology (on the job market!);
- **Dr. Zhu Li** and **his research group** from University of Missouri, Kansas City;
- **Dr. Tiefeng Jiang** from University of Minnesota, Twin Cities & The Chinese University of Hong Kong (Shenzhen).

# Table of Contents

# Table of Contents

# Introduction

- Subspace selection algorithms, such as Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA), have limitations in capturing the nonlinearity and local variation of high-dimensional data sets.
- Nonlinear methods like kernel algorithms, manifold learning, and Deep Neural Networks (DNNs) have been proposed to address these limitations but are computationally expensive.
- To overcome these challenges, we introduce Subspace Indexing Model with Interpolation (SIM-I), which constructs a piecewise linear, locality-aware, and globally nonlinear model of low-dimensional embedding.
- SIM-I utilizes the "center of mass" calculation on Stiefel and Grassmann manifolds for smooth regularization. Additionally, SIM-I can be used to build a Lightweight Inference Engine (LIE) that carries similar feature extraction capabilities as pre-trained DNNs.

# What is LPP model?

- Locality Preserving Projection: a dimension reduction method.
- High-dimensional data $x \in \mathbb{R}^D \Rightarrow$ Low-dimensional embedding $y = W^T x \in \mathbb{R}^d$, $d \ll D$. Projection Matrix $W \in \mathbb{R}^{D \times d}$.
- How do we know that the projection preserves the geometric features of the original dataset $\{x_1, ..., x_n\}$?
- Locality: intrinsic relative geometric relations between the data points in the original high-dimensional space.
- Affinity matrix $S = (s_{ij})$.
- Example: $s_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$ when $\|x_i - x_j\| < \varepsilon$ and $s_{ij} = 0$ otherwise, where $1 \le i, j \le n$.

# What is LPP model?

- Optimization Problem:

$$\min_{W} \phi(W) = \frac{1}{2} \sum_{i,j=1}^{n} s_{ij} \|y_i - y_j\|^2 \ .$$

- $y_i = Wx_i$ and $y_j = Wx_j$.
- We are seeking for the embedding matrix $W$ such that close pairs of image points $x_i$ and $x_j$ will be mapped to close pairs of embeddings $y_i = W^T x_i$ and $y_j = W^T x_j$, and vice versa.
- This helps to preserve the local geometry of the data set, a.k.a the locality.

# What is LPP model?

- We introduce a weighted fully-connected graph $G$ where the vertex set consists of all data points $x_1, ..., x_n$ and the weight on the edge connecting $x_i$ and $x_j$ is given by $s_{ij} \geq 0$.
- Consider the diagonal matrix $\mathcal{D} = \mathrm{diag}(\mathcal{D}_{11}, ..., \mathcal{D}_{nn})$ where $\mathcal{D}_{ii} = \sum_{j=1}^{n} s_{ij}$, and we then introduce the graph Laplacian $L = \mathcal{D} - S$.
- Normalization constraint $\sum_{i=1}^{n} \mathcal{D}_{ii} y_i^2 = 1$,
- Minimization problem reduces to the following generalized eigenvalue problem

$$XLX^T \boldsymbol{w} = \lambda X\mathcal{D}X^T \boldsymbol{w} \ ,$$

where $X = [x_1, ..., x_n] \in \mathbb{R}^{D \times n}$.

-

$$W = [\boldsymbol{w}_1, ..., \boldsymbol{w}_d] \ .$$

# Table of Contents

# SIM-I Methodology

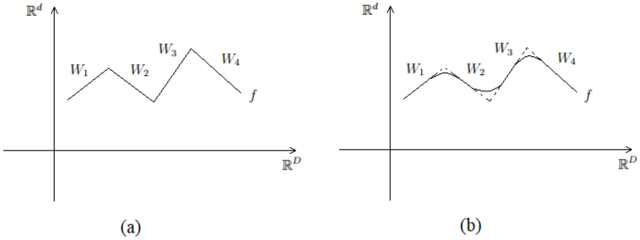


Fig. 1: (a) Step 1; (b) step 2

SIM-I is constructed via two steps:

- **Step 1**: We build a piecewise linear affinity-aware subspace model under a given partition of the data set.
- **Step 2**: We interpolate between several adjacent linear subspace models constructed in the first step using the "center of mass" calculation on Stiefel and Grassmann manifolds.

# Piecewise linear Locality Preserving Projection (LPP) model

**Objective**: LPP aims to find a linear transformation that maps high-dimensional data to a lower-dimensional space.

**Example**: Given an input data set $\mathcal{X} = \{x_1, ..., x_n\}$ where each $x_i \in \mathbb{R}^D$, either labelled or unlabelled,

- Apply techniques like PCA (Principal Component Analysis) to reduce the dimensionality of the data PCA to $\mathcal{X}$, which selects the first $d$ bases $[a_1, ..., a_d]$ with the largest variance.
- Then apply a k-d tree based partition scheme to divide the whole data set $\mathcal{X}$ into non-overlapping subsets $C_1, ..., C_{2^h}$ where $h$ is the depth of the tree.
- For each $C_k$, a low-dimensional embedding matrix $W_k \in \mathbb{R}^{D\times d}$ can be constructed. This is usually done via *Locality Preserving Projection* (LPP).

# Stiefel & Grassmann manifold

### Definition (Stiefel manifold)

The compact *Stiefel manifold* $\mathrm{St}(d, D)$ is a submanifold of the Euclidean space $\mathbb{R}^{D \times d}$ such that

$$\mathrm{St}(d, D) = \{X \in \mathbb{R}^{D \times d} : X^T X = I_d\} \ . \tag{1}$$

### Definition (Grassmann manifold)

The Grassmann manifold $\mathrm{Gr}(d, D)$ is defined to be the quotient manifold $\mathrm{Gr}(d, D) = \mathrm{St}(d, D)/O(d)$. A point on $\mathrm{Gr}(d, D)$ is defined by an equivalence class $[W] = \{WO_d, O_d \in O(d)\}$ where $W \in \mathrm{St}(d, D)$.

# Stiefel center-of-mass

## Definition (center-of-mass)

Given a sequence of matrices $W_1, \dots, W_I \in \mathrm{St}(d, D)$ and a sequence of weights $w_1, \dots, w_I > 0$, the *Stiefel center-of-mass* with respect to the distance $\mathrm{d}(W_1, W_2)$ on $\mathrm{St}(d, D)$ is defined as a matrix $W_c = W_c^{\mathrm{St}}(W_1, \dots, W_I; w_1, \dots, w_I) \in \mathrm{St}(d, D)$ such that

$$\begin{aligned} W_c &= W_c^{\mathrm{St}}(W_1, \dots, W_I; w_1, \dots, w_I) \\ &\equiv \arg \min_{W \in \mathrm{St}(d,D)} \sum_{j=1}^{I} w_j \mathrm{d}^2(W, W_j) \ . \end{aligned} \tag{2}$$

# Grassmann center-of-mass

## Definition (center-of-mass)

Similarly, if the corresponding equivalent classes are $[W_1], ..., [W_l] \in \mathrm{Gr}(d, D)$, then the *Grassmann center-of-mass* with respect to the distance $\mathrm{d}([W_1], [W_2])$ on $\mathrm{Gr}(d, D)$ is defined as the equivalence class $[W_c]$, where $W_c = W_c^{\mathrm{Gr}}(W_1, ..., W_l; w_1, ..., w_l) \in \mathrm{St}(d, D)$ is such that

$$\begin{aligned} W_c &= W_c^{\mathrm{Gr}}(W_1, ..., W_l; w_1, ..., w_l) \\ &\equiv \arg \min_{W \in \mathrm{St}(d,D)} \sum_{j=1}^{l} w_j \mathrm{d}^2([W], [W_j]) \ . \end{aligned} \tag{3}$$

# Stiefel & Grassmann Interpolation

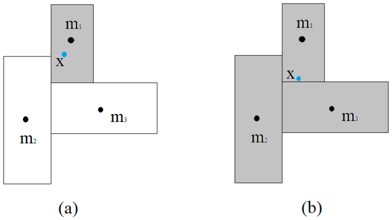


Fig. 2: Stiefel & Grassmann Interpolation

There are $l = 3$ nearby subsets $C_1, C_2, C_3$ with means $m_1, m_2, m_3$ in the training set.
(a) Test point $x$ is apparently close to $m_1$, and thus the low-dimensional embedding $f(x)$ is taken as $f(x) = W_1^T x$, where $W_1$ is the LPP subspace based on $C_1$;
(b) Test point $x$ has approximately the same distances to $m_1, m_2, m_3$, and thus the embedding $f(x)$ is taken as $f(x) = W_c^T x$, where $W_c$ is the Stiefel/Grassmann center-of-mass for the LPP subspace models $W_1, W_2, W_3$ based on $C_1, C_2, C_3$.

# Algorithm I

---

Algorithm 1: SIM-I: Subspace Indexing Model with Interpolation

---

1: **Input**: Data set $\mathcal{X} = \{x_1, ..., x_n \in \mathbb{R}^D\}$ and its corresponding affinity matrix $S = (s_{ij})_{1 \le i,j \le n}$; test point $x \in \mathbb{R}^D$; threshold ratio $r_{\text{thr}} > 1$; tree depth $h$; parameter $K > 0$

2: Using an initial PCA and a k-d tree based partition scheme, decompose the data set $\mathcal{X}$ into subsets $C_1, ..., C_{2^h}$, where $h$ is the depth of the tree

3: For each subset $C_k$, calculate its mean (center) $m_k \in \mathbb{R}^D$ and its LPP embedding matrix $W_k \in \mathrm{St}(d, D)$ based on the affinity matrix $S$

4: Sort the distances $\|x - m_k\|$ in ascending order $\|x - m_{k_1(x)}\| \le ... \le \|x - m_{k_{2^h}(x)}\|$, $\{k_1(x), ..., k_{2^h}(x)\} = \{1, ..., 2^h\}$

5: Determine $I$, which is the first sub-index $i$ of $k_i(x)$ such that $\|x - m_{k_{I+1}(x)}\| > r_{\text{thr}}\|x - m_{k_1(x)}\|$

# Algorithm II

6: Set $j_i(x) = k_i(x)$ for $i = 1, 2, ..., I$ and obtain the embedding matrices $W_{j_1(x)}, ..., W_{j_I(x)} \in \mathrm{St}(d, D)$ or their corresponding subspaces $[W_{j_1(x)}], ..., [W_{j_I(x)}] \in \mathrm{Gr}(d, D)$, together with the choice of weights $w_i \equiv 1$ or $w_i = \exp(-K\|x - m_{j_i(x)}\|^2) > 0$ for $i = 1, ..., I$

7: Find a center-of-mass $W_c = W_c^{\mathrm{St}}(W_{j_1(x)}, ..., W_{j_I(x)}; w_1, ..., w_I)$ (Stiefel case) or $[W_c] = [W_c^{\mathrm{Gr}}(W_{j_1(x)}, ..., W_{j_I(x)}; w_1, ..., w_I)]$ (Grassmann case) according to Definition and Theorems in last section.

8: **Output**: The low-dimensional embedding $f(x) = W_c^T x \in \mathbb{R}^d$

# Table of Contents

# Lightweight Inference Engine(LIE)

LIE is a novel inference framework that utilizes SIM-I's efficient data representation.

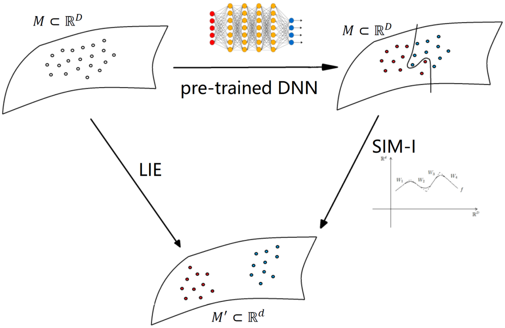


Fig. 3: SIM-I combined with a pre-trained learning model and the data manifold enables Lightweight Inference Engine (LIE).

# LIE with Label

**LIE with Pre-trained Model Label:** LIE utilizes labels from high-dimensional training data, as provided by pre-trained models such as DNNs.

- This approach can be seen as mapping directly from the high-dimensional data manifold to classification labels, leveraging the feature extraction capabilities of the pre-trained model.

**LIE with Natural Label:** LIE can also be effectively applied using the original, natural labels of the training samples.

- This methodology enables LIE to function as an independent learning model, achieving comparable classification accuracy to standard models like DNNs.

# LIE insights

- **Shallow Network Design**: SIM-I based LIE resembles a shallow neural network with a wide first layer and ReLU-like nonlinearities, differing from typical deep neural networks.
- **Deep to Wide Transition**: It represents a shift from deep to wide but shallow network architectures, especially relevant when integrating with pre-trained DNNs.
- **Unique Activation and Feedback**: Unlike standard networks, LIE features non-standard, implicit activation functions and feedback circuits, setting it apart from traditional models.
- **Alternative to NTK Models**: LIE serves as an alternative to Neural Tangent Kernel (NTK) wide networks, potentially offering new insights into the interpretability of deep neural networks.

# Table of Contents

# PCA Recovery for SIFT Dataset

- **Dataset and Setup**: Used the SIFT dataset, where each data point has 128 elements. A total of 10068850 sample points were used, with $200 \times 213$ elements selected for training. The data was projected onto a 16-dimensional space using a Stiefel matrix.
- **Procedure**: The data set was divided into 8192 subsets using a k-d tree partition, with PCA embedding matrices calculated for each subset. The recovery efficiency was compared between the standard PCA method and SIM-I.

# Results

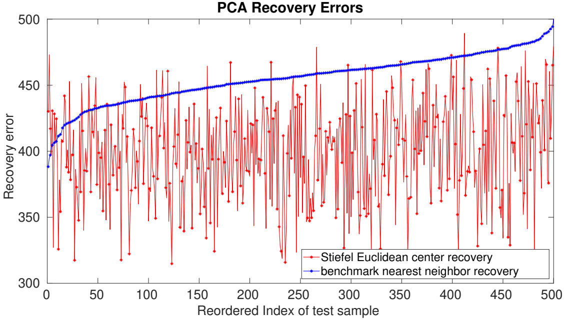


Fig. 4: Comparison of the PCA Recovery Errors: Blue = benchmark case using closest subset PCA recovery, with the error sorted from low to high; Red = using SIM-I based on Stiefel center-of-mass and $\mathrm{d}(W_1, W_2) = \|W_1 - W_2\|_F$.

The SIM-I method demonstrated improved recovery efficiency for about 93% of test points compared to the benchmark PCA method. The empirical average errors for SIM-I and the benchmark were 399.786223 and 455.537462, respectively.

# Nearest-Neighbor Classification with SIM-I on MNIST Dataset

- **Data Selection and Processing**: The MNIST dataset was used, with a training set of 60,000 and a test set of 10,000. Initial projections were made onto a lower-dimensional subspace using PCA.
- **Data Clustering and LPP Embedding**: Using the kd-tree method, the training data was divided into clusters. For each cluster, a Locality Preserving Projection (LPP) embedding matrix was computed. These matrices were then used to form a representation of the data in a Grassmann manifold.

# Results

- **Classification Tests**: Two tests were performed using different numbers of nearest neighbors (knn).

| Method | knn = 1 | knn = 75 |
|---|---|---|
| Baseline Method | 93.58% | 87.52% |
| SIM-I Method | 95.55% | 94.21% |

Table 1: Comparison of Classification Rates between Baseline and SIM-I Methods

- **Experiment with Pre-trained LeNet-5 Labels**: Experimented with training sample labels provided by a pre-trained LeNet-5 model. The SIM-I based LIE achieved a classification rate of 96.14%, surpassing the baseline method's 88.76%.

- **Conclusion**: The experiments demonstrated that the SIM-I method consistently outperformed the baseline method in terms of classification accuracy on the MNIST dataset. This indicates the efficacy of the SIM-I approach and its application in building a LIE for classification tasks.

# Classification on PCA Embedded Face Images Dataset I

- **Dataset and Image Processing**: The experiment utilized a large dataset of over 400,000 face images with 900 subjects. The images were preprocessed using face detection tools from OpenCV to crop out just the faces. These cropped images were then resized to standardize their dimensions.

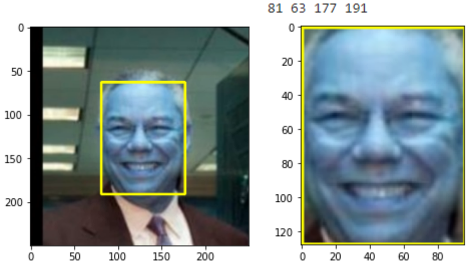


Fig. 5: An example with calculated face coordinates from LFW image set.

# Classification on PCA Embedded Face Images Dataset II

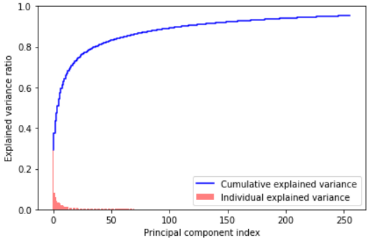


Fig. 6: The Explained Variance of 256 dimensional PCA embedding for face dataset.

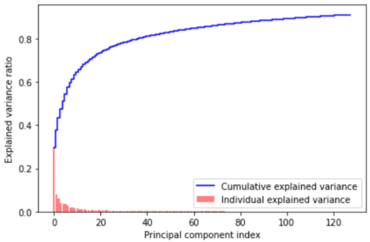


Fig. 7: The Explained Variance of 128 dimensional PCA embedding for face dataset.

- **Dimensionality Reduction with PCA**: Due to the large size of the dataset, PCA (Principal Component Analysis) was applied to reduce the dimensionality of the data. Two tests were performed: one reducing the dimensions to 256 and the other to 128. The explained variances were 95.4% and 91% for 256 and 128 dimensions, respectively.

# Results

| Method | 256-Dim PCA | 128-Dim PCA |
|---|---|---|
| Neural Network (NN) | 97.0% | 96.5% |
| LIE (Grassmann Center-of-Mass) | 96.3% | 95.0% |

Table 2: Comparative Classification Rates for NN and LIE

- **Conclusion**: The experiment showed that the SIM-I based LIE achieved a high level of classification accuracy, comparable to neural network models, while using a simpler architecture.

# Table of Contents

# Conclusion

- **Introduction of SIM-I**: The paper introduces the Subspace Indexing Model with Interpolation (SIM-I), which utilizes Locality Preserving Projection (LPP) embedding and center-of-mass calculations on Stiefel and Grassmann manifolds.
- **Development of LIE**: Building upon the SIM-I framework and using a pre-trained learning model, the research develops a Lightweight Inference Engine (LIE).
- **Future Insights into DNNs**: The paper proposes that the approach of transforming DNNs into wide and shallow networks, as exemplified by LIE, can offer more insights into the interpretability of DNNs in the future.

Thank You!