Think Globally, Fit Locally: Summary

Dustin Leatherman

May 16, 2020

Contents

1 Preface

This document consists of a Summary of "Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds" by Lawrence K. Saul and Sam T. Rowels.

The three-pass technique is utilized here.

1.1 Three-Pass Technique

For the uninitiated, the three-pass technique is a method for reading dense article or papers that breaks down reading into three steps.

1. Read the Abstract and headers.
2. Read the First sentence of each paragraph from end-to-end.
3. Read the entire paper end-to-end.

For (1), each header in this document pertains to a header of the paper.

For (2), the first sentence of each paragraph has been paraphrased. If the first sentence was not descriptive enough, the second sentence was used. Any formulas from a previous paragraph that was referenced in the first sentence of the following paragraph, the formula was recorded.

For (3), a "Third Pass" sub-header following the paragraph contains details picked up on the third read.

The conclusion of the Three-pass technique should result in a reconstruction of the original paper.

1.2 Notations

There are not any mathematical notations that deviate from the norm in this document.

> Comments or questions interjected by myself are placed in "quote" bodies to provide delineation between the source material and my own thoughts as I grapple its meaning.

1.3 Keywords

- Dimension Reduction
- Manifolds
- Locally Linear Embedding
- Unsupervised Learning

2 Introduction

> The introduction contains a series of definitions of key concepts gleaned from the paper. This is different than other sections in that extra care and attention was spent upfront in ensuring the definitions of key terms were understood.

Preprocessing: Obtain more useful representations of raw signals for subsequent operations. i.e. classification, denoising, interpolation, visualization, outlier detection.

Unsupervised Learning: Framework of automatic methods to discover hidden structures of statistical regularities.

Density Estimation: Learn parameters of a prob. model for prediction (like λ in Poisson)

Dimensionality Reduction: Compact representations of original data which contain enough information to make decisions.

> Dimensionality Reduction is applied here in a non-prob and non-parametric setting.

Problem trying to be addressed: Compute a low dim embedding of high dim data sampled with noise from an underlying manifold. Intersection of Geometry + Statistics. Use this to discover the *few degrees of freedom* underlying the observations.

> **Manifold**: A topological space that locally represents a Euclidean space near each point. Manifolds are continuous (homeomorphic) to n-dimensions. Manifolds cannot intersect (like a figure 8). Examples include: a line, a circle, Surfaces (plane, sphere, torus). A Manifold may *not* be homeomorphic beyond n-dimensions.

PCA: Compute linear projections of greatest variance from top eigenvectors of the covariance matrix.

Multidimensional Scaling (MDS): Compute low dim embedding that best preserves pairwise distances between points.

If using euclidean distance, MDS equivalent to PCA. Both powerful *Linear* methods for dim reduction

Locally Linear Embedding (LLE): Dimension Reduction technique for non-linear data. Involves a sparse eigenvalue problem that scales well to high dim datasets.

- More accurate for reconstructing manifolds in lower dimensions than linear methods (PCA/MDS)

3 Algorithm

3.1 1

Based on simple geometric intuitions. Computes low dim embedding with the property that nearby points in the high dim space are nearby and co-located with respect to one another in the low dim space.

3.1.1 Third Pass

- Embedding optimized to preserve local configurations of nearest neighbors.

  This is accomplished by calculating weights based on the distance between K-nearest neighbors and using those weights in the Embedding cost function.

- LLE doesn't need to use measures of distance or relation to far away points

  This is because the algorithm restricts distance measures to the K nearest neighbors.

3.2 2

Data consists of N real-valued vectors $\vec{X}_i$ of dimensionality D, sampled from an underlying smooth manifold.

3.2.1 Third Pass

What does "Smooth" mean in mathematical terms?

- Each data point and its neighbors should lie on or close to the manifold.

  In this case, the manifold is equivalent to the original image so when talking about being close to the manifold, this means that sampled points shouldn't be far-fetched from the original points.

- "Smooth" and "well-sampled" mean that the data point has on the order of 2d neighbors which define an approximately linear patch on the manifold.

  This allows us to treat $\vec{X}_i$ as a linear combination of its neighbors. Later, this allows an approximation of $\vec{X}_i$ to be constructed based off its neighbors.

  - How many neighboring data points are considered orthogonal?

3.3 3

In the simplest form of LLE, identify the K nearest neighbors per data point by Euclidean Distance.

Reconstruction Errors - Cost Function

$$E(W) = \Sigma_i |\vec{X}_i - \Sigma_j W_{ij} \vec{X}_j|^2 \tag{1}$$

3.3.1 Third Pass

- Reconstruction Cost Function adds up squared distances between all points and their reconstructions.

W_{ij}: contribution of the jth data point to the ith reconstruction.
Weights are computed using Least Squares with two constraints

1. Sparseness
   - Each $\vec{X}_i$ is reconstructed from only its neighbors.
   - $W_{ij} = 0$ if $\vec{X}_j$ not in the set.
2. In-variance
   - $\Sigma_j W_{ij} = 1$

3.4 4

The constrained weights that minimize the reconstruction errors have several important symmetries: For any data point, they are invariant to rotations, rescalings, and translations from that data point to its neighbors. This also means they are invariant to global rotations, rescalings, and translations.

Embedding Cost Function

$$\Phi(Y) = \Sigma_i |\vec{Y}_i - \Sigma_j W_{ij} \vec{Y}_j|^2 \tag{2}$$

3.4.1 Third Pass

- Invariance from Rotations and rescaling follows from (1)
- Invariance to translation enforced by sum-to-one constraint.
- Reconstruction of weights **not** invariant to shear transformations.

> Shear mapping is a linear map that displaces each point in a fixed direction, by an amount proportional to its signed distance from the line that is parallel to that direction and goes through the origin.
> **Horizontal Shear example**
> $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x + my \\ y \end{bmatrix} = \begin{bmatrix} 1 & m \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
> **Vertical Shear Example**
> $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ mx + y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ m & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
> I believe its not invariant because a translation or rescaling affects all elements of a matrix whereas a shear transformation affects individual elements. This means that the matrix doesn't keep the same "shape"?

This means that the reconstruction weights do not depend on a particular frame of reference.

3.5 5

Steps

1. Compute neighbors of each $\vec{X}_i$
2. Compute weights W_{ij} that best reconstruct each $\vec{X}_i$ from its neighbors, minimizing the Weight cost function (1) by using constrained Linear fits.
3. Compute $\vec{Y}_i$ best reconstructed by W_{ij}, minimizing the quadratic form (2) by its bottom non-zero eigenvectors.

3.6 6

Suppose data lie on or near a manifold of $d << D$. To a good approximation, we imagine there exists a linear mapping that maps the high dim coords to each neighborhood to global internal coords on the manifold.

3.6.1 Third Pass

Since the reconstruction weights W_{ij} are invariant to translation, rotation, and rescaling transformations, it is expected that the weights equally and correctly describe the reconstructed patches in both D and d dimensions.

3.7 7

Imagine cutting out locally linear patches of the manifold and rearranging them in the low dim embedding space. If the placement of each patch involves no more than a translation, rotation, and/or rescaling, then angles between data points will be preserved.

3.8 8

LLE constructs a neighborhood mapping on the above idea.

3.8.1 Third Pass

- The third step of LLE maps a neighborhood from D to d dimensions.
- $\vec{Y_i}$ are chosen to minimize the Embedded cost function (2)
  - Similar to (1) except the weights are fixed and the outputs $\vec{Y_i}$ are optimized.
- Note that no $\vec{X_i}$ is used. Only W_{ij}

3.9 9

The embedding cost function (2) defines a quadratic form in the outputs $\vec{Y_i}$. Subject to constraints that make the problem well-posed, the cost function has a unique global minimum.

3.9.1 Third Pass

- The Embedded cost function has a unique global minimum which is returned as the output of LLE (low dimensional embedding).
  - It can be minimized using a sparse $N \times N$ eigenvalue problem.

    The eigenvalue problem is defined as
    $Av = \lambda v$

3.10 10

Note that while reconstruction weights for each data point are computed from its local neighborhood (independent of the weights for other data points) the embedding coordinates are computed by an $N \times N$ eigensolver, a global operation that couples all data points that lie in the same connected component of the graph defined by the neighbors.

Is "connected" the same as continuous and homeomorphic?

3.10.1 Third Pass

- The algorithm discovers the *global* structure by integrating information from overlapping neighborhoods.
- The dth coordinate output by LLE corresponds to the $(d+1)^{st}$ smallest eigenvector of the cost matrix.
  - LLE Coordinates are ordered/nested and immutable when new dimensions are added.

3.11 11

Implementation is straightforward. In the simplest formulation of LLE, there is only one free parameter: number of neighbors per data point K.

3.11.1 Third Pass

- Optimal weights W_{ij} and $\vec{Y_i}$ are computed using Least Squares.
- Single pass through the three steps to find the global minima of the reconstruction and embedding costs.
- No learning rates, annealing schedules are required. No random initialization or local optima affect the final results

> Annealing is a probabilistic heuristic technique for finding a global optimum. Usually over a discrete set of values, it runs Monte Carlo simulations to determine optimum.
>
> In this context, I think an annealing schedule is referring to a "warm up" period that could help reduce the amount of points to choose from.

4 Examples

4.1 1

Embeddings discovered by LLE are easiest to visualize for data samples from 2-dim manifolds.

4.2 2

Under the right conditions, LLE can learn the stereo-graphic mapping from sphere to plane.

4.2.1 Third Pass

- Required conditions
  - the North pole must be excluded
  - Data sampled uniformly in manifold coordinates.
- Assumes that density increases as one approaches the north pole
- suggests that Local angles but not distances are preserved.
- Still an open item if this mapping can be discovered by LLE (as of 2003).

> What is the latest research on applying LLE to stereographic mappings successfully?

4.3 3

Figure 5 shows another 2-dim manifold, but one living in a much higher dimensional space.

4.3.1 Third Pass

- Figure 5 shows LLE reconstruction of a series of faces to be superior to PCA.

4.4 4

Low dimensional outputs of LLE can be used to index the original collection of high dimensional images. Fast and accurate indexing is an essential component of example-based video synthesis from a large library of stored frames.

4.5 5

LLE scales relatively well to large datasets because it generates *sparse* intermediate results and eigenproblems.

5 Implementation

The algorithm consists of three steps:

1. Nearest neighbor search (to identify the non-zero elements of the weight matrix)
2. Constrained Least Squares Fits (to compute the values of these weights)
3. Singular Value Decomposition (to perform the embedding)

5.1 Neighborhood Search

5.1.1 1

In the simplest formulation of the algorithm, one identifies a fixed number of nearest neighbors, K, per data point, as measured by Euclidean Distance.

1. Third Pass

   Other Criteria for choosing neighbors

   - all points within a ball of fixed radius

     Is a ball different than a sphere?
     Yes, a ball encompasses all points inside a spherical object. The points on a sphere only encompass the surface of the sphere.

- locally derived distance metrics that deviate significantly from the euclidean norm
  - based on prior knowledge
  - estimated curvature
  - pairwise distances
  - nonparametric techniques like box-counting
- Number of neighbors can differ between each point. Good place to introduce domain knowledge.

5.1.2 2

The results of LLE are typically stable of a range of neighborhood sizes. The size of the that range depends on various features of the data, such as the sampling density and the manifold geometry.

1. Third Pass Criteria to keep in mind when selecting K
   - $d < K$. Some margin between d and K is required to obtain a topology-preserving embedding but the exact margin is unknown

     Has this been answered by newer research?
   - a neighborhood must be locally linear. Meaning smaller K is needed for datasets with curvature.
   - When $K > D$ (low dimension data), each datapoint can be reconstructed perfectly from its neighbors and the weights are no longer uniquely defined. Regularization must be added to "break" this.

5.1.3 3

The nearest neighbor step in LLE is simple to implement, though it can be time consuming for large datasets ($N \leq 10^4$) if performed *without* any optimizations.

1. Third Pass
   - $O(DN^2)$ for KNN
   - K-D Trees or Ball Trees can be used to compute neighbors in $O(Nlog(N))$

A Ball Tree is a Binary Tree where the Leaf Node is a Ball. A ball is comprised of a set of points where the distance to the center of the ball is minimized.
A K-D tree is a K-dimensional tree where the Leaf Node is K-dimensional point. This is similar to a ball tree except in how the contents of the Leaf node is decided.

- Approximation methods may also be used but do so at your own risk.

5.1.4 4

An implementation of LLE also needs to check that the graph formed by linking each data point to its neighbors is connected.

1. Third Pass

   - If the graph is disconnected or weakly connected, apply LLE separately to the data of each graph's strongly connected components
     - One may also refine the neighborhood selection rule

     I am pretty sure this is confirming the assumption of homeomorphic/continuity within the neighborhood of the points.
     Is each neighborhood considered convex?

5.2 Constrained Least Squares Fits

5.2.1 1

The second step of LLE is to reconstruct each data point from its nearest neighbors. The optimal reconstruction weights can be computed in closed form.

Reconstruction Error

$$\epsilon = |\vec{x} - \Sigma_j w_j \vec{\eta_j}|^2 = |\Sigma_j w_j (\vec{x} - \vec{\eta_j})|^2 = \Sigma_{jk} w_j w_k G_{jk} \tag{3}$$

$$G_{jk} = (\vec{x} - \eta_j) \cdot (\vec{x} - \vec{\eta_k}) \tag{4}$$

Optimal Weights using Inverse of Gram Matrix

$$w_j = \frac{\Sigma_k G_{jk}^{-1}}{\Sigma_{lm} G_{lm}^{-1}} \tag{5}$$

1. Third Pass
    - We assumed that the sum of the weights = 1.
        - First identity exploits this
    - Second identity introduces a local Gram matrix
      $G = A^T A$
    - Reconstruction error can be minimized analytically using a Lagrange Multiplier to enforce the sum-to-one constraint.
    - It is faster to solve a linear system of equations then re scale the weights so the sum to one, than calculate the inverse of the Gram matrix as described in (5)

5.2.2 2

In unusual cases, it can arise that the Gram matrix in (4) is singular or nearly singular. For example, when there are more neighbors than input dimensions ($K > D$), or when the data points are not in general position. When $K > D$, Least squares problem for finding the weight does not have a unique solution. Thus elements of the Gram matrix need to be conditioned before solving.

$$G_{jk} \leftarrow G_{jk} + \delta_{jk}(\frac{\Delta^2}{K})Tr(G)$$

1. Third Pass
    - Least Squares doesn't provide a unique solution so regularities must be applied
    - δ_{jk}: 1 if j == k, else 0
    - $Tr(G)$: Trace of G
    - $\Delta^2 << 1$

      This means that $\Delta < 1$ is a hyper-parameter chosen ahead of time.
    - Adds a regularization term to the reconstruction cost that measures summed squared magnitudes of the weights.

5.2.3 3

The regularization term ($\frac{\Delta^2}{K}$) acts to penalize large weights that exploit correlations beyond some level of precision in the data sampling process. It may also introduce some robustness to noise and outliers.

5.2.4 4

Computing the reconstruction weights W_{ij} is typically the least expensive step of the LLE algorithm.

1. Third Pass

    - $O(DNK^3)$ operations required to solve a $K \times K$ set of linear equations for each data point.
    - Weight can be stored as a sparse matrix with NK Nonzero elements.

5.3 Eigenvalue Problem

5.3.1 1

The final step of LLE is to compute a low dimensional embedding based on the reconstruction weights W_{ij} of the high dimensional inputs $\vec{X}_i$. Only information captured by the weights W_{ij} is used to construct the embedding.

$$\Phi(Y) = \Sigma_{ij} M_{ij} (\vec{Y}_i \cdot \vec{Y}_j) \tag{6}$$

$$M_{ij} = \delta_{ij} - W_{ij} - Wji + \Sigma_k W_{ki} W_{kj} \tag{7}$$

$$\Sigma_i \vec{Y}_i = \vec{0} \tag{8}$$

$$\frac{1}{N} \Sigma_i \vec{Y}_i \vec{Y}_i^T = I \tag{9}$$

1. Third Pass

- Low Dimension outputs $\vec{Y_i}$ found by minimizing (2) for fixed weights $\$W_{ij}$.
  - The Embedded cost function is minimized when W_{ij} are used to construct Y_i
- M is sparse, symmetric, and semipositive definite.

5.3.2 2

The optimization of (6) is performed subject to constraints that make the problem well-posed.

1. Third Pass
   - $\vec{Y_i}$ can be translated by a constant without affecting cost, $\Phi(Y)$ in (2).
   - We remove this translational degree of freedom by requiring the outputs to be centered on the origin. $\Sigma_i \vec{Y_i} = \vec{0}$
   - We remove the rotational degree of freedom and fixing scaling by constraining $\vec{Y_i}$ to have unit covariance and outer products that satisfy (9). This constraint is chosen for three reasons
     - different coordinates in embedding space should be uncorrelated to second-order

       What does this mean?
     - reconstruction errors for these coordinates should be measured on the same scale.
     - the scale should be an order of unity.

       What is an order of unity?

5.3.3 3

Under these restrictions, the optimal embedding - up to a trivial global rotation of the embedding space - is found by minimizing (2) subject to the constraints in $(8) - (9)$. This can be done in many ways, but the most straightforward is to find the bottom $d+1$ eigenvectors of the cost matrix, M. (Bottom or Top eigenvectors correspond to largest or smallest eigenvalues).

1. Third Pass
   - Equivalence of optimization between normalized Quadratic form and the computation of the largest or smallest eigenvectors is a version of the Rayleitz-Ritz theorem.

- Lagrange multipliers enforce (8) − (9)
- Setting Gradients with respect to $\vec{Y_i} = 0$ yield a symmetric eigenvalue problem (like PCA).
- The bottom eigenvector is a unit vector with all equal components. This is discarded to enforce (8). This is because the other eigenvectors are orthogonal to the bottom one so the rest of the Y_i sum to 0.
- The rest of the d eigenvectors are the d embedding coordinates found by LLE.

5.3.4 4

Note that the bottom $d+1$ eigenvectors of the sparse, symmetric matrix M can be found **without** performing a full matrix diagonalization.

1. Third Pass

   M can be represented as a product of 2 sparse matrices. This gives large computational savings for large N.

   $$M = (I - W)^T (I - W)$$

   Look up eigensolvers.

5.3.5 5

The final step of LLE is typically the most computationally expensive. Without special optimizations, computing the bottom eigenvectors scales as $O(dN^2)$.

5.3.6 6

Note that the d^{th} coordinate output by LLE always corresponds to the $(d+1)^{st}$ smallest eigenvector of the matrix M, regardless of the total number of outputs computed or the order in which they are calculated.

6 Extensions

6.1 LLE from Pairwise Distances