Optimization Algorithms for Scalable and Robust Dimensionality Reduction

Seppo Pulkkinen

University of Turku

February 8, 2013
Outline

1. Introduction

2. Maximum Variance Unfolding (MVU)

3. Low-rank Approach to MVU

4. Noise Reduction as a Preprocessing Step for MVU
Motivation

- Dimensionality reduction plays a very important role in data analysis, machine learning and pattern recognition.
- **Curse of dimensionality**: operating on high-dimensional data directly is very costly.

- Consider an example from pattern recognition:
  - A digital image can be considered as a vector in an $n$-dimensional space, where $n$ is the number of pixels.
  - In many cases, a set of images can be parametrized by a small number of variables (e.g. those representing rotation, scaling and illumination).

- Linear methods (such as PCA and MDS) are inadequate to describe complex nonlinear patterns, and thus nonlinear methods are required.
Basic Assumptions

• A manifold $\mathcal{M}$ can be parametrized with respect to a small set of variables in some convex domain $\mathcal{D} \subset \mathbb{R}^m$ with $m \ll d$.
• There exists a (smooth) mapping $f: \mathcal{D} \rightarrow \mathbb{R}^d$ that parametrizes the manifold $\mathcal{M}$. That is, $\mathcal{M} = f(\mathcal{D})$.
• The goal of dimensionality reduction is to map the manifold $\mathcal{M}$ isometrically (i.e. by preserving distances) onto the parameter space $\mathbb{R}^m$. That is, determine $\mathcal{D} = f^{-1}(\mathcal{M})$. 
In practice we only have a finite set of input points \( \{ x_1, x_2, \ldots, x_n \} \) sampled from the manifold \( \mathcal{M} \subset \mathbb{R}^d \).

Determine a set of output points \( \{ y_1, y_2, \ldots, y_n \} \subset \mathbb{R}^m \) such that distances between (neighbouring) samples are preserved.

A commonly used technique is to construct a neighbourhood graph from the input points \( \{ x_i \} \) and determine a low-dimensional representation from the graph.
Maximum Variance Unfolding (MVU) is a well-known dimensionality reduction method that "unfolds" a given point set \( \{x_1, x_2, \ldots, x_n\} \):

- the method constructs a \( k \)-neighbourhood graph (with edge set \( E \))
- pairwise point distances are maximized
- pairwise distances between \( k \)-nearest neighbours are preserved
- the resulting point set is centered to the origin.

The above considerations lead to the optimization problem:

\[
\begin{align*}
\max_{y_1, y_2, \ldots, y_n \in \mathbb{R}^d} & \quad \sum_{i,j=1}^{n} \|y_i - y_j\|^2 \\
\text{s.t.} & \quad \|y_i - y_j\|^2 \leq \|x_i - x_j\|^2, \quad \{i, j\} \in E, \\
& \quad \sum_{i=1}^{n} y_i = 0
\end{align*}
\]

with respect to the output points \( Y = [y_1, y_2, \ldots, y_n]^T \in \mathbb{R}^{n \times d} \).
Maximum Variance Unfolding (MVU)

Example: The Swiss Roll Dataset

- From a given point set $\{x_i\} \subset \mathbb{R}^d$, MVU produces a set of output points $\{y_i\}$ with dimension $m \leq d$.
- The output points $\{y_i\}$ lie close to or on a linear subspace.
- The low-dimensional representation can be obtained from the leading eigenvalues and eigenvectors of the kernel matrix $K = YY^T$.
- In fact, this means applying principal component analysis (PCA) to the output points $\{y_i\}$. 
Other Datasets
SDP Formulation (Weinberger and Saul)

Reformulate the quadratic problem as an SDP (semidefinite program)

\[
\max_{K \in \mathbb{R}^{n \times n}} \text{tr}(K) \\
\text{s.t. } K \succeq 0, \quad K \text{ symmetric} \\
\sum_{i,j=1}^{n} K_{ij} = 0, \\
K_{ii} - 2K_{ij} + K_{jj} \leq \left\| x_i - x_j \right\|^2, \quad \{i, j\} \in E
\]

and optimize with respect to the elements of the kernel matrix \( K \).

- The SDP is a concave maximization problem with a unique solution.
- Interior-point SDP solvers (e.g. CSDP, SeDuMi and SDPT3) are applicable.
- However, the number of variables is \( \frac{n(n+1)}{2} \), which limits the SDP approach to small datasets (say \( n < 1500 \)).
Low-rank Approach to MVU

Nonlinear Low-rank Formulation

- The solution of the MVU problem is usually of low rank when the intrinsic dimension of the input points is low.
- Again, consider the quadratic problem

\[
\text{max}_{y_1, y_2, \ldots, y_n \in \mathbb{R}^d} \sum_{i,j=1}^{n} \| y_i - y_j \|^2 \\
\text{s.t.} \sum_{i=1}^{n} y_i = 0, \\
\| y_i - y_j \|^2 \leq \| x_i - x_j \|^2, \quad \{i, j\} \in E
\]

with some \(d \ll n\). This problem has only \(nd\) variables.
- This is a convex maximization problem. Any "blind" attempt to solve it with some \(d\) would lead to getting trapped into a local maximum.
- Furthermore, the optimal dimension \(d\) needs to be determined.
• **Incremental rank update:** Solve the optimization problem and when necessary, increase the column dimension \( d \) of the matrix \( Y = [ y_1 \ y_2 \ \cdots \ y_n ]^T \) (i.e. rank of the matrix \( K = YY^T \)).

• When \( d = n \), any local minimum is also a global minimum.

• It is possible to derive sufficient and necessary conditions for a solution of the quadratic problem to be a solution of the SDP.

• These conditions are in practice satisfied with \( d \ll n \).

---

S. Pulkkinen (2013).
Incremental Low-Rank Approach to Finding Optimal Graph Embeddings
*TUCS Technical Report*, to appear
Ipopt (Interior Point OPTimizer)

- A state-of-the-art software library for large-scale constrained nonlinear optimization.
- Implements an interior-point primal-dual method with a filter-based line search.
- Designed for problems where
  - second-order derivative information is available
  - the Hessian of the Lagrangian and the constraint Jacobian are sparse.
- Handles efficiently problems with a large number of constraints.
- Implemented in C++, and can be linked to C++, C and Fortran code.
- Implements MATLAB and R interfaces and can be called from the AMPL modeling environment.
- Maintained by Carl Laird and Andreas Wächter.
- Won the 2011 Wilkinson Prize for Numerical Software.
- [http://projects.coin-or.org/Ipopt](http://projects.coin-or.org/Ipopt)
## Comparison of Solvers

Computation times of the tested solvers (in seconds) on two synthetically generated test datasets:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>n</th>
<th>CSDP</th>
<th>SDPA</th>
<th>SDPLR</th>
<th>IPOPT-MVU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incomplete tire</td>
<td>800</td>
<td>500.84</td>
<td>427.58</td>
<td>65.53</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td>1600</td>
<td>4639.25</td>
<td>3723.21</td>
<td>233.92</td>
<td>9.76</td>
</tr>
<tr>
<td></td>
<td>2500</td>
<td>-</td>
<td>-</td>
<td>2556.40</td>
<td>19.51</td>
</tr>
<tr>
<td></td>
<td>4000</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>23.99</td>
</tr>
<tr>
<td>Swiss roll</td>
<td>800</td>
<td>597.52</td>
<td>476.24</td>
<td>36.30</td>
<td>5.32</td>
</tr>
<tr>
<td></td>
<td>1600</td>
<td>5062.44</td>
<td>3656.24</td>
<td>193.59</td>
<td>10.41</td>
</tr>
<tr>
<td></td>
<td>2500</td>
<td>24213.58</td>
<td>14571.69</td>
<td>1229.00</td>
<td>14.33</td>
</tr>
<tr>
<td></td>
<td>4000</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>39.75</td>
</tr>
</tbody>
</table>

- CSDP and SDPA are interior-point SDP solvers.
- SDPLR is an augmented Lagrangian-type algorithm for solving the low-rank reformulation of the SDP.
Scalability of Ipopt for Solving the MVU Problem

- Ipopt scales extremely well when applied to the low-rank MVU problem.
- The Hessian of the Lagrangian function and the constraint Jacobian are very sparse (both have only $O(n)$ nonzero elements).
- The sparsity eliminates the computational bottleneck typically arising in the solution of a linear system of KKT equations.
- Here we have used Harwell MA57 for solving the KKT equations.

![Graph showing MVU with Ipopt, Swiss roll dataset.](image)
Noise in Input Data

- A key assumption of MVU (and other graph-based methods) is that the points lie directly on a low-dimensional manifold.
  - If this assumption is not satisfied, MVU usually fails to produce a useful low-dimensional representation.
- Let us then assume that the points are sampled from a set of \( n \) generating functions \( f_i \) with random noise. That is,
  \[
  (X \mid I = i, \Theta = \theta) = f_i(\theta) + \varepsilon, \quad i = 1, 2, \ldots, n,
  \]
  where the random variable \( I \) represents the index of a generating function and \( \Theta \) represents coordinates for the generating function.
- We assume that the random variables are distributed according to
  \[
  p_I(I = i) = \frac{1}{n}, \quad (\Theta \mid I = i) \sim \mathcal{U}(D_i) \text{ and } \varepsilon \sim \mathcal{N}_d(0, \sigma^2),
  \]
  where \( \mathcal{U} \) denotes a uniform distribution and \( \mathcal{N}_d \) denotes a \( d \)-dimensional normal distribution.
Marginalization

Given the values of the random variables $I$ and $\Theta$, the conditional density for $X$ is given by

$$ p_X(x \mid I = i, \Theta = \theta) = \frac{1}{(\sqrt{2\pi}\sigma)^d} \exp \left( -\frac{\|x - f_i(\theta)\|^2}{2\sigma^2} \right). $$

The "trick" is to marginalize the joint density to get rid of the unknown variables $I$ and $\Theta$. By

- summing over the domain of the discrete variable $I$
- integrating over the domains of the continuous variable $\Theta$
- using the definition of joint probability density

we obtain the marginal density

$$ p_X(x) = \sum_{i=1}^{n} \int_{D_i} p_{X,I,\Theta}(x, i, \theta) d\theta $$

$$ = \sum_{i=1}^{n} \int_{D_i} p_X(x \mid I = i, \Theta = \theta)p_{\Theta}(\theta \mid I = i)p_I(i) d\theta. $$
Ridges of the Kernel Density

- **Ridges** of the marginal density induced by the model give a good approximation of the generating functions.
- In practice, we estimate the marginal density from the samples \( \{y_1, y_2, \ldots, y_n\} \) nonparametrically with Gaussian kernels

\[
\hat{p}(x) = \frac{1}{n(2\pi)^{d/2}h^d} \sum_{i=1}^{n} \exp \left( -\frac{\|x - y_i\|^2}{2h^2} \right).
\]
Definition of a Ridge

Definition

A point $\mathbf{x} \in \mathbb{R}^d$ belongs to the $m$-dimensional ridge set $R_p^m$, where $0 \leq m < d$, of a twice differentiable probability density $p : \mathbb{R}^d \rightarrow \mathbb{R}$ if

$$
\nabla p(\mathbf{x})^T \mathbf{v}_i(\mathbf{x}) = 0, \quad \text{for all } i > m,
$$

$$
\lambda_{m+1}(\mathbf{x}) < 0,
$$

$$
\lambda_1(\mathbf{x}) > \lambda_2(\mathbf{x}) > \cdots > \lambda_{m+1}(\mathbf{x}), \quad \text{if } m > 0,
$$

where $\lambda_1(\mathbf{x}) \geq \lambda_2(\mathbf{x}) \geq \cdots \geq \lambda_d(\mathbf{x})$ and $\{\mathbf{v}_i(\mathbf{x})\}_{i=1}^d$ denote the eigenvalues and the corresponding eigenvectors of $\nabla^2 p(\mathbf{x})$, respectively.

- An $m$-dimensional ridge set is a generalization of a set of modes (maxima) of a probability density.
- When $m = 0$, a ridge set reduces to a set of modes.
Trust Region Newton Method for Ridge Projection

- Successively maximize the quadratic model

\[
Q_k(s) = \hat{p}(x_k) + \nabla \hat{p}(x_k)^T s + \frac{1}{2} s^T \nabla^2 \hat{p}(x_k) s
\]

of the kernel density estimate within a trust region of radius \(\Delta_k\).

- Maximize the quadratic model in the subspace spanned by the eigenvectors corresponding to the \(d - m\) smallest eigenvalues:

\[
\max_s Q_k(s) \quad \text{s.t.} \quad \begin{cases} 
\|s\| \leq \Delta_k, \\
s \in \text{span}(v_{m+1}(x_k), v_{m+2}(x_k), \ldots, v_d(x_k))
\end{cases}
\]

- A generalization of projection onto a principal hyperplane (as in PCA).
The proposed algorithm (GTRN) has significantly lower computation times than the subspace-constrained means-shift (SCMS) algorithm by Ozertem and Erdogmus (2011).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SCMS</th>
<th>GTRN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>num. eval.</td>
<td>CPU time</td>
</tr>
<tr>
<td>Circle</td>
<td>13 965</td>
<td>7.85</td>
</tr>
<tr>
<td>DistortedHalfCircle</td>
<td>12 161</td>
<td>6.81</td>
</tr>
<tr>
<td>DistortedSSShape</td>
<td>10 561</td>
<td>5.93</td>
</tr>
<tr>
<td>HalfCircle</td>
<td>9 878</td>
<td>5.54</td>
</tr>
<tr>
<td>Helix</td>
<td>24 520</td>
<td>91.94</td>
</tr>
<tr>
<td>Spiral</td>
<td>15 984</td>
<td>15.19</td>
</tr>
<tr>
<td>Spiral3d</td>
<td>12 070</td>
<td>14.37</td>
</tr>
<tr>
<td>Zigzag</td>
<td>12 214</td>
<td>7.12</td>
</tr>
</tbody>
</table>

Function evaluations and CPU times used by the SCMS and GTRN algorithms for ridge projection, synthetically generated datasets.
Examples

Spiral3d

Helix

Kernel density ridge projections of the Spiral3d and Helix datasets.
Main Contributions

- **Scalable** solution approach to the MVU problem:
  - A low-rank approximation to reduce the number of variables.
  - Solution of the low-rank approximation by Ipopt, a state-of-the-art nonlinear interior point solver.
  - Theoretical results that ensure a globally optimal solution.

- A **noise reduction** method to facilitate the solution of the MVU problem:
  - A generative model for a noisy point set.
  - Using ridges of the underlying density as an estimator for the generating functions.
  - A computationally efficient trust region Newton method for ridge projection.

---

S. Pulkkinen and M.M. Mäkelä and N. Karmitsa (2012).
A Generalized Trust Region Newton Method Applied to Noise Reduction

*TUCS Technical Report 1061*, submitted to COAP