A tractable latent variable model for nonlinear dimensionality reduction

Lawrence Saul

Department of Computer Science and Engineering
University of California, San Diego

November 16, 2020 — Applied Math Seminar — Yale University
How to find low-dimensional structure in high-dimensional data?

Outline

1. Problem and motivation
2. Model and methods
3. Results and discussion

For more details, see:

Problem and motivation

High-dimensional data arises in many areas of research:

- High-resolution images
- Word-document counts
- Gene-expression profiles

Low-dimensional representations have many uses:

- To help visualize modes of variability.
- To help accelerate queries of large data collections.
- To reveal hidden structure in complex data.
**Problem statement**

- **Intuitively**
  Unfold, compress, and/or project the data into a space of much lower dimensionality, *but without losing information*. 

- **Mathematically** (*to be made more precise*)

  Given \( n \) high-dimensional inputs \( \{x_i\}_{i=1}^{n} \) with \( x_i \in \mathbb{R}^D \), infer \( n \) low-dimensional outputs \( \{\mu_i\}_{i=1}^{n} \) with \( \mu_i \in \mathbb{R}^d \), such that:

  (i) \( d < D \) (and preferably \( d \ll D \)),

  (ii) only *nearby* inputs are mapped to *nearby* outputs.
Linear methods

We can use principal component analysis (PCA) to compute subspaces of maximum variance.

But obviously PCA cannot solve problems in nonlinear dimensionality reduction (NLDR).
Spectral, graph-based methods for NLDR

Isomap (Tenenbaum et al, 2000)
Locally linear embedding (Roweis & Saul, 2000)
Laplacian eigenmaps (Niyogi & Belkin, 2003)
Maximum variance unfolding (Weinberger & Saul, 2004)
Diffusion maps (Coifman et al, 2005)

These are rooted in tractable, well-understood optimizations:

- All shortest paths through a graph
- Least-squares reconstructions
- Eigenvectors of the discrete graph Laplacian
- Semidefinite programming

They work well on many problems. But for visualization, there has emerged a more popular class of methods.
Probabilistic methods for NLDR

- **tSNE**  t-distributed Stochastic Neighborhood Embedding  
  (van der Maaten & Hinton, 2008)

- **LargeVis**  (Tang et al., 2016)

- **UMAP**  Uniform Manifold Approximation and Projection  
  (McInnes, Healy, & Melville, 2018)

All these methods optimize probabilistic notions of similarity  
(e.g., minimizing KL divergences by gradient descent).

**These methods work very well in practice:**

- They produce remarkable 2d/3d visualizations.
- They scale to very large data sets.
Example

- Grayscale images of handwritten digits \((D = 28 \times 28)\)
  
  \[
  \begin{array}{cccccccc}
  0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9
  \end{array}
  \]

- Visualization \((d = 2)\) of digit images \((n = 70000)\)

  \[
  \begin{array}{c}
  \text{tSNE} \\
  \text{UMAP}
  \end{array}
  \]
tSNE, LargeVis, and UMAP are widely used in practice. Each can also be used to validate the others.

What is left to explore?
Quite a bit ...
Questions

1. **What sorts of optimizations should we expect for NLDR?**

   The objective functions for tSNE, LargeVis, and UMAP are *differentiable*, but that's about it.

   Can we exploit more elegant optimizations for NLDR? Sparse graphs seem to encode a lot of structure.
What probabilistic models should we expect for NLDR?

There are widely used latent variable models (LVMs) for clustering, factor analysis, Kalman filtering, etc.

tSNE, LargeVis, and UMAP all have probabilistic foundations. But they look nothing like well-established LVMs.

Is there an equally tractable LVM for NLDR?
The problem seems to lend itself to one:

Given inputs $\{x_i\}_{i=1}^n$ with $x_i \in \mathbb{R}^D$, infer outputs $\{\mu_i\}_{i=1}^n$ with $\mu_i \in \mathbb{R}^d$ that are neighborhood-preserving.
This talk

1. A tractable LVM for NLDR

NLDR joins the family of canonical problems with tractable LVMs (i.e., those with no need for sampling-based or variational inference).

2. NLDR via sparse least-squares

NLDR joins the family of graph-based problems that can exploit fast Laplacian solvers (Spielman & Teng, 2014) as a core primitive.

3. A scalable framework for Big Data

A coarse-graining procedure breaks the main logjam of NLDR, the large number of pairwise interactions between dissimilar inputs.
And also an effective algorithm ...

So let’s get started ...
Outline

1. Problem and motivation
2. Model and methods
3. Results and discussion
The idea in a nutshell

- **Probabilistic model**
  Map the inputs $\{x_i\}_{i=1}^{n}$ stochastically into a latent space of lower dimensionality.

- **Maximum likelihood estimation**
  Adjust the parameters of this mapping so that, with high probability, only nearby inputs map to nearby outputs.

- **Balancing act**
  Maximize the likelihood by balancing two objectives: pulling nearby inputs together, pushing distant inputs apart.
What we need to do

1. How to define this model?
2. How to estimate this model?
3. How to scale this model?
The probabilistic model

- **Parameterizing the model**

  To each high-dimensional input \( x_i \in \mathbb{R}^D \), we associate two parameters:

  1. a low-dimensional output \( \mu_i \in \mathbb{R}^d \)
  2. a variance \( \sigma_i^2 > 0 \)

- **Embedding the inputs**

  The model maps each input \( x_i \in \mathbb{R}^D \) stochastically to its latent space:

  \[
P(h|x_i) = \frac{1}{(2\pi\sigma_i^2)^{-d/2}} \exp\left(-\frac{1}{2\sigma_i^2} \| h - \mu_i \|^2 \right)
  \]
The similarity graph

- **Meeting the neighbors**

Before we can map similar inputs to nearby outputs, we must specify which inputs to regard as similar.

- **Encoding neighbors by a sparse graph** $S$

Let $\langle i, j \rangle \in S$ if we consider $x_i$ and $x_j$ to be similar. We can regard $S$ as a set of edges or as a sparse matrix $S \in \{0, 1\}^{n \times n}$.

- **Constructing the graph** $S$

  Simplest graph: $\langle i, j \rangle \in S$ if $x_i \in k\text{NN}(x_j)$ or vice versa.
  More robust: $\langle i, j \rangle \in S$ if $x_i$ and $x_j$ are mutual $k\text{NN}$.
  Connected: add edges from minimum weighted spanning tree.

(Carreira-Perpina & Zemel, 2005)
Imagine an experiment in which two similar inputs $x_i$ and $x_j$ (with $S_{ij} = 1$) are stochastically mapped to the latent space:

$$h \sim \mathcal{N}(\mu_i, \sigma_i^2),$$
$$h' \sim \mathcal{N}(\mu_j, \sigma_j^2).$$

With high probability we want $||h - h'|| \ll ||x_i - x_j||$. Let $c_+ \in \{0, 1\}$ denote a binary random variable with

$$P(c_+ = 1|h, h') = 2^{-\frac{||h - h'||^2}{||x_i - x_j||^2}}.$$

Are the inputs $x_i$ and $x_j$ faithfully embedded into the latent space? We can measure the success of this experiment by computing

$$P(c_+ = 1|x_i, x_j) = \int_{h, h'} P(c_+ = 1|h, h') P(h|x_i) P(h'|x_j).$$
Imagine an experiment in which two dissimilar inputs $x_i$ and $x_j$ (with $S_{ij} = 0$) are stochastically mapped to the latent space:

$$h \sim \mathcal{N}(\mu_i, \sigma_i^2),$$
$$h' \sim \mathcal{N}(\mu_j, \sigma_j^2).$$

With high probability we want $\|h - h'\| \gg \max_k \left[ S_{ik}\|x_i - x_k\| \right]$. Let $c_\in \{0, 1\}$ denote a binary random variable with

$$P(c_\in = 1|h, h') = 2^{-\frac{\|h - h'\|^2}{\max_k [S_{ik}\|x_i - x_k\|^2]}}.$$

Are the inputs $x_i$ and $x_j$ faithfully embedded into the latent space? We can measure the success of this experiment by computing

$$P(c_\in = 0|x_i, x_j) = \int_{h,h'} P(c_\in = 0|h, h') P(h|x_i) P(h'|x_j).$$
An objective function for the model

\begin{align*}
\mathcal{L} &= \frac{1}{|S|} \sum_{ij} S_{ij} \log P(c_+ = 1|x_i, x_j) + \frac{1}{|S|} \sum_{ij} \overline{S}_{ij} \log P(c_- = 0|x_i, x_j) \\
&\phantom{=} \text{sum over pairs of similar inputs} + \text{sum over pairs of dissimilar inputs}
\end{align*}

Two interpretations of \( \mathcal{L} \):

1. an objective function for the parameters \( \{\mu_i, \sigma_i^2\}_{i=1}^n \) of a stochastic neighborhood-preserving embedding

2. the normalized log-likelihood of a latent variable model with Gaussian latent variables \( h, h' \in \mathbb{R}^d \):

\begin{align*}
P(c_+ = 1|x_i, x_j) &= \int_{h, h'} P(c_+ = 1|h, h') P(h|x_i) P(h'|x_j), \\
P(c_- = 0|x_i, x_j) &= \int_{h, h'} P(c_- = 0|h, h') P(h|x_i) P(h'|x_j). \\
\end{align*}

But is this model tractable?
Is the model tractable?

Well, there’s good news and bad news ...
Is the model tractable?

This posterior distribution is not multivariate Gaussian:

\[
P(h, h'|c_0 = 0, x_i, x_j)
\]

\[
\propto P(h|x_i) P(h'|x_j) P(c_0 = 0|h, h')
\]

Bayes rule

\[
= e^{-\frac{\|h - \mu_i\|^2}{2\sigma_i^2}} e^{-\frac{\|h' - \mu_j\|^2}{2\sigma_j^2}} \left[1 - e^{-\frac{\|h - h'\|^2}{2\Delta_{ij}^2}}\right]
\]

That typically does not bode well ... but in this case the posterior \( P(h, h'|c_0 = 0, x_i, x_j) \) has a special property:

It is equal to a weighted difference of two multivariate Gaussian distributions.
All of the below can be computed \textit{analytically}.

- For pairs of similar inputs ($S_{ij} = 1$):

  \[ P(c_+ = 1|x_i, x_j) \]  
  \[ E[h | c_+ = 1, x_i, x_j] \]  
  \[ E[h' | c_+ = 1, x_i, x_j] \]  
  \[ E[\|h\|^2 | c_+ = 1, x_i, x_j] \]  
  \[ E[\|h'\|^2 | c_+ = 1, x_i, x_j] \]

- For pairs of dissimilar inputs ($S_{ij} = 0$):

  \[ P(c_- = 0|x_i, x_j) \]  
  \[ E[h | c_- = 0, x_i, x_j] \]  
  \[ E[h' | c_- = 0, x_i, x_j] \]  
  \[ E[\|h\|^2 | c_- = 0, x_i, x_j] \]  
  \[ E[\|h'\|^2 | c_- = 0, x_i, x_j] \]
What we need to do

1. How to define this model?
2. How to estimate this model?
3. How to scale this model?
Some canonical latent variable models (LVMs):

- Factor analysis
- Hidden Markov models
- Gaussian mixture models

How are these models estimated?

Expectation-Maximization (EM) algorithm

<table>
<thead>
<tr>
<th>E-step</th>
<th>Compute statistics of the posterior distribution.</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-step</td>
<td>Use these statistics to update the model parameters.</td>
</tr>
</tbody>
</table>

EM is an alternating iterative procedure for LVMs that converges monotonically in the likelihood.

(Dempster, Laird, & Rubin, 1977)
Compute the following posterior statistics:

\[
\phi_{ij} = \mathbb{E} \left[ \| h - \mu_i \|^2 \mid c_+ = 1, x_i, x_j \right]
\]

\[
\phi'_{ij} = \mathbb{E} \left[ \| h' - \mu_j \|^2 \mid c_+ = 1, x_i, x_j \right]
\]

for pairs of similar inputs \((S_{ij} = 1)\)

\[
\bar{\phi}_{ij} = \mathbb{E} \left[ h \mid c_- = 0, x_i, x_j \right]
\]

\[
\bar{\phi'}_{ij} = \mathbb{E} \left[ h' \mid c_- = 0, x_i, x_j \right]
\]

for pairs of dissimilar inputs \((S_{ij} = 0)\)

\[
\psi_{ij} = \mathbb{E} \left[ \| h - \mu_i \|^2 \mid c_- = 0, x_i, x_j \right]
\]

\[
\psi'_{ij} = \mathbb{E} \left[ \| h' - \mu_j \|^2 \mid c_- = 0, x_i, x_j \right]
\]

All of these statistics can be computed analytically.
(We’ll address the issue of scaling later …)
M-step for NLDR

The updates take a simple form in terms of these statistics. As shorthand, let $s_{ij} = S_{ij}/|S|$ and $\tilde{s}_{ij} = \tilde{S}_{ij}/|\tilde{S}|$.

- **Update for variances**

$$\sigma_i^2 \leftarrow \frac{1}{d} \cdot \frac{1}{\sum_j \left[ s_{ij} \phi_{ij} + s_{ji} \phi'_ji + \tilde{s}_{ij} \psi_{ij} + \tilde{s}_{ji} \psi'_ji \right]} \cdot \sum_j [s_{ij} + s_{ji} + \tilde{s}_{ij} + \tilde{s}_{ji}]$$

- **Update for outputs**

$$\mu_i \leftarrow A^{-1} \sum_j \left[ \tilde{s}_{ij} \bar{h}_{ij} + \tilde{s}_{ji} \bar{h}'_{ji} \right]$$

where $A$ is a symmetric diagonally dominant (SDD) matrix with the same off-diagonal zeros as the similarity graph $S$. 
Properties of these updates

- **Monotonic convergence**

  These updates are guaranteed to improve the model's log-likelihood (with no pesky learning rates):

  \[
  \mathcal{L} = \sum_{ij} \left[ s_{ij} \log P(c_+ = 1|x_i, x_j) + \bar{s}_{ij} \log P(c_- = 0|x_i, x_j) \right]
  \]

- **Nearly linear-time solvers**

  \[
  \mu_i \leftarrow \mathbf{A}^{-1} \sum_{j} \left[ \bar{s}_{ij} \bar{h}_{ij} + \bar{s}_{ji} \bar{h}'_{ji} \right]
  \]

  There are extremely efficient solvers for SDD linear systems. This extends the so-called *Laplacian paradigm* to NLDR.

  (Koutis, Miller, & Peng, 2012; Spielman & Teng, 2014)
Scaling the model

- **Original log-likelihood**

\[
\mathcal{L} = \sum_{ij} \left[ s_{ij} \log P(c_+ = 1|\mathbf{x}_i, \mathbf{x}_j) + \bar{s}_{ij} \log P(c_- = 0|\mathbf{x}_i, \mathbf{x}_j) \right]
\]

How to manage the \(O(n^2)\) terms for pairs of dissimilar inputs? (UMAP and LargeVis use a random sampling of these pairs.)

- **Coarse-grained log-likelihood**

We can construct a recursive approximation to this log-likelihood, trading off accuracy (on shorter length scales) for speed:

<table>
<thead>
<tr>
<th>levels of recursion ((\ell))</th>
<th># of terms in (\mathcal{L})</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(O(n^2))</td>
</tr>
<tr>
<td>1</td>
<td>(O(n^{4/3}))</td>
</tr>
<tr>
<td>2</td>
<td>(O(n^{10/9}))</td>
</tr>
</tbody>
</table>
Coarse-graining: before versus after

BEFORE

AFTER

Randomly designate $n_\ell$ inputs as landmarks.
Delete dissimilarity edges between non-landmarks.
Add similarity edges from inputs to closest landmarks.
Upweight dissimilarity edges between landmarks.

Non-landmarks will be separated by the triangle inequality.
How many landmarks to choose?

Assume that $n_{\ell}$ landmarks are evenly distributed among $n$ inputs. How many pairs of dissimilar inputs remain in the likelihood?

BEFORE

After coarse-graining:

$O \left( n^{2} \right)$

After coarse-graining:

$O \left( n_{\ell}^{2} + n_{\ell}(n/n_{\ell})^{2} \right)$

By choosing $n_{\ell} \sim n^{2/3}$, we reduce the number of pairs to $O \left( n^{4/3} \right)$. 
Recursion for massive data sets

What if $O(n^{4/3})$ is still prohibitively large?

- **Super-landmarks**
  Sample $n^{2/3}_\ell$ of the $n_\ell$ landmarks as *super-landmarks*, and model their separation at even longer length scales.

- **Sub-landmarks**
  Sample $n_\ell(n/n_\ell)^{2/3}$ of inputs as *sub-landmarks*, and model their separation at shorter length scales.

- **In practice**
  With two levels of recursion, $n \approx 200000$, and $d = 10$, each iteration of EM takes $< 4$ sec on a modest laptop.

- **And so on ...**
Outline

1. Problem and motivation
2. Model and methods
3. Results and discussion
Handwritten digits

\( n = 70K \) grayscale images

\( D = 50 \) reduced by PCA from \( 28 \times 28 \)

\( d = 2 \) embedding after 400 updates of EM

\( \ell = 1 \) one round of coarse-graining
Handwritten digits

(n = 70K) grayscale images

(D = 50) reduced by PCA from 28 × 28

dl = 2 embedding after 400 updates of EM

(ℓ = 1) one round of coarse-graining
Handwritten digits

\[ n = 70K \] grayscale images

\[ D = 50 \] reduced by PCA from \( 28 \times 28 \)

\[ d = 2 \] embedding after 400 updates of EM

\[ \ell = 1 \] one round of coarse-graining
Handwritten digits

\[ n = 70K \] grayscale images

\[ D = 50 \] reduced by PCA from \( 28 \times 28 \)

\[ d = 2 \] embedding after 400 updates of EM

\[ \ell = 1 \] one round of coarse-graining
inputs  \( n \approx 207K \) normalized vectors of dimensionality \( D = 300 \) from a word2vec model trained on the GoogleNews corpus

outputs \( d = 10 \) embedding after 400 iterations of EM, with \( \ell = 2 \) rounds of coarse-graining

Nearly 40% of words retain the same 1NN in the \( d = 10 \) embedding. (PCA requires \( d \approx 75 \) dimensions for this same degree of recall.)

<table>
<thead>
<tr>
<th>word</th>
<th>4nn (before)</th>
<th>4nn (after)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tractable</td>
<td>willer, tractability,</td>
<td>tractability, Rube_Goldberg, effervescence,</td>
</tr>
<tr>
<td></td>
<td>intractable, impulsion</td>
<td>ingenious</td>
</tr>
<tr>
<td>latent</td>
<td>unexpressed, unarticulated, untapped, Latent</td>
<td>unexpressed, unvoiced, unarticulated, inchoate</td>
</tr>
<tr>
<td>variable</td>
<td>Variable, adjustable,</td>
<td>Variable, Fixed, fixed, assignable</td>
</tr>
<tr>
<td></td>
<td>quantize, variability</td>
<td></td>
</tr>
<tr>
<td>model</td>
<td>models, Model, concept,</td>
<td>Model, models, Models, Modeling</td>
</tr>
<tr>
<td></td>
<td>paradigm</td>
<td></td>
</tr>
<tr>
<td>nonlinear</td>
<td>linear, oscillatory,</td>
<td>linear, lineal, gravitational_attraction,</td>
</tr>
<tr>
<td></td>
<td>diffusive, fractal</td>
<td>gravitation</td>
</tr>
<tr>
<td>dimensionality</td>
<td>dimensional, translucency, tonal, spatiality</td>
<td>Dimensional, Dimensional, Dimension, translucency</td>
</tr>
<tr>
<td>reduction</td>
<td>reductions, decrease,</td>
<td>increases, decreases, increase, reductions</td>
</tr>
<tr>
<td></td>
<td>increase, increases</td>
<td></td>
</tr>
</tbody>
</table>
Discussion

- **Relation to previous work**
  
  UMAP & LargeVis optimize a similar objective, but rely on gradient ascent and negative sampling (versus EM and coarse-graining).

  tSNE optimizes a different objective, which is mainly useful for low-dimensional embeddings ($d = 2$ or $d = 3$).

- **Directions for further work**
  
  - Laplacian least-squares updates
  - Variance-based outlier detection
  - Speedups from coarse-graining
  - Additional latent variables

  Faster solvers?

  More robust embeddings?

  Larger data sets?

  Richer models?
Where to find more

- paper: https://www.pnas.org/content/117/27/15403
- code: https://osf.io/wy793/

Thank you!