Partial-Hessian Strategies for Fast Learning of Nonlinear Embeddings

Max Vladymyrov and **Miguel Á. Carreira-Perpiñán**

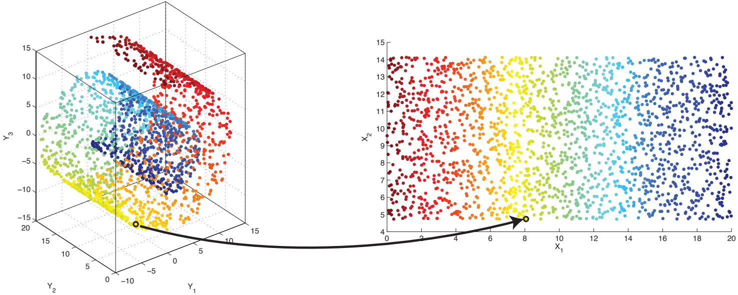
Electrical Engineering and Computer Science
University of California, Merced

`https://eecs.ucmerced.edu`

August 30, 2012

Dimensionality reduction

Given a high-dimensional dataset $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_N) \subset \mathcal{R}^D$ find a low-dimensional representation $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N) \subset \mathcal{R}^d$ where $d \ll D$.



Can be used for:

- Data compression.
- Visualization.
- Detect latent manifold structure.
- Fast search.
- ...

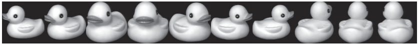
Graph-based dimensionality reduction techniques

- Input: (sparse) affinity matrix $\mathbf{W}$ defined on a set of high-dimensional points $\mathbf{Y}$.
- Objective function: minimization over the latent points $\mathbf{X}$.
- Examples:
  - Spectral methods: Laplacian Eigenmaps (LE), LLE;
    - ✓ closed-form solution;
    - ✗ results can be bad.
  - Nonlinear methods: SNE, t-SNE, elastic embedding (EE);
    - ✓ better results;
    - ✗ slow to train, limited to small data sets.

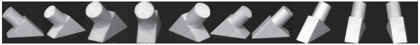
COIL-20 Dataset

Rotations of 10 objects every 5°; input is greyscale images of 128×128.

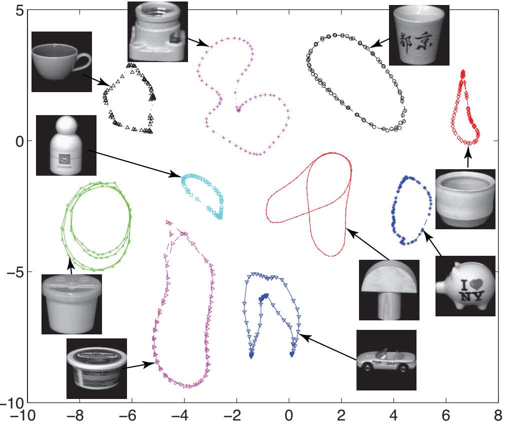
Y :



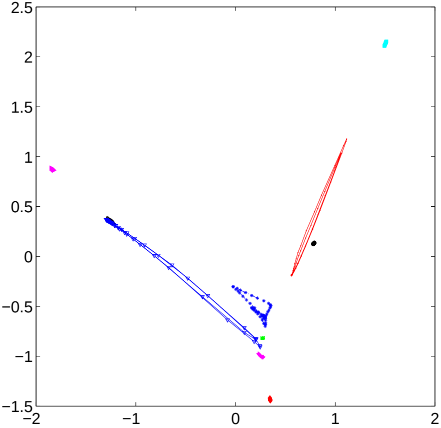
. . .



Elastic Embedding



Laplacian Eigenmaps



General Embedding Formulation (Carreira-Perpiñán 2010)

For $\mathbf{Y} \in \mathcal{R}^{D \times N}$ matrix of high-d points and $\mathbf{X} \in \mathcal{R}^{d \times N}$ low-d points

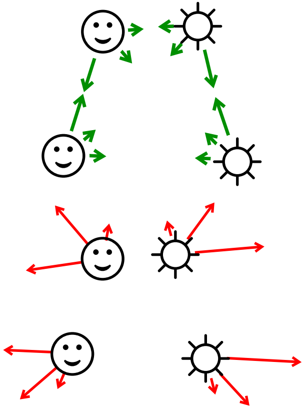
$$E(\mathbf{X}, \lambda) = E^{+}(\mathbf{X}) + \lambda E^{-}(\mathbf{X}) \qquad \lambda \geq 0$$

$E^{+}(\mathbf{X})$ is the *attractive term*:

- often quadratic,
- minimal with coincident points;

$E^{-}(\mathbf{X})$ is the *repulsive term*:

- often very nonlinear,
- minimal with points separated infinitely.



Optimal embeddings balance both forces.

General Embedding Formulation: Special Cases

	$E^+(\mathbf{X})$	$E^-(\mathbf{X})$
SNE: (Hinton&Roweis,'03)	$\sum_{n,m=1}^{N} p_{nm} \left\Vert \mathbf{x}_n - \mathbf{x}_m \right\Vert^2$	$\sum_{n=1}^{N} \log \sum_{m=1}^{N} e^{-\left\Vert \mathbf{x}_n - \mathbf{x}_m \right\Vert^2}$
t-SNE: (van der Maaten & Hinton,'08)	$\sum_{n,m=1}^{N} p_{nm} \log \left(1 + \left\Vert \mathbf{x}_n - \mathbf{x}_m \right\Vert^2\right)$	$\log \sum_{n,m=1}^{N} \left(1 + \left\Vert \mathbf{x}_n - \mathbf{x}_m \right\Vert^2\right)^{-1}$
EE: (Carreira-Perpiñán,'10)	$\sum_{n,m=1}^{N} w_{nm}^{+} \left\Vert \mathbf{x}_n - \mathbf{x}_m \right\Vert^2$	$\sum_{n,m=1}^{N} w_{nm}^{-} e^{-\left\Vert \mathbf{x}_n - \mathbf{x}_m \right\Vert^2}$
LE & LLE: (Belkin & Niyogi,'03) (Roweis & Saul,'00)	$\sum_{n,m=1}^{N} w_{nm}^{+} \left\Vert \mathbf{x}_n - \mathbf{x}_m \right\Vert^2$ s.t. constraints	0

w_{nm}^{+} and w_{nm}^{-} are affinity matrices elements

Optimization Strategy

For every iteration k:

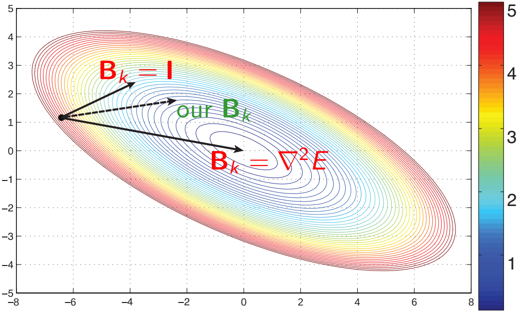
1. Choose positive definite $\mathbf{B}_k$.
2. Solve a linear system $\mathbf{B}_k \mathbf{p}_k = -\mathbf{g}_k$ for a search direction $\mathbf{p}_k$, where $\mathbf{g}_k$ is the gradient.
3. Use line search to find a step size α for the next iteration $\mathbf{X}_{k+1} = \mathbf{X}_k + \alpha \mathbf{p}_k$.(e.g. with backtracking line search).

Convergence is guaranteed! (under mild assumptions)

How to choose good $\mathbf{B}_k$?

Solve linear system $\mathbf{B}_k \mathbf{p}_k = -\mathbf{g}_k$:

$\mathbf{B}_k = \mathbf{I}$ (grad. descent) $\xrightarrow[\text{faster convergence rate}]{\text{more Hessian information}}$ $\mathbf{B}_k = \nabla^2 E$ (Newton's method)



We want $\mathbf{B}_k$:

- contain as much Hessian information as possible;
- positive definite (pd);
- fast to solve the linear system and scale up to larger N.

The Spectral Direction

The Hessian of the generalized embedding formulation is given by:

$$\nabla^2 E \;=\; 4(\mathbf{L}^+ - \lambda \mathbf{L}^-) \otimes \mathbf{I}_d \;+\; 8\mathbf{L}^{xx} \;-\; 16\lambda \operatorname{vec}(\mathbf{X}\mathbf{L}^q) \operatorname{vec}(\mathbf{X}\mathbf{L}^q)^T$$

where $\mathbf{L}^+$, $\mathbf{L}^-$, $\mathbf{L}^{xx}$, $\mathbf{L}^q$ are graph Laplacians.

$\mathbf{B} = 4\mathbf{L}^+ \otimes \mathbf{I}_d$ is a convenient Hessian approximation:

- block-diagonal and has d blocks of $N \times N$ graph Laplacian $4\mathbf{L}^+$;
- always psd $\Rightarrow$ global convergence under mild assumptions;
- **constant** for Gaussian kernel. For other kernels we can fix it at some $\mathbf{X}$;
- equal to the Hessian of the spectral methods: $\nabla^2 E^+(\mathbf{X})$;
- "bends" the gradient of the nonlinear E using the curvature of the spectral E^+;

The Spectral Direction (computation)

Solve $\mathbf{B}\mathbf{p}_k = \mathbf{g}_k$ efficiently for every iteration k (naively $\mathcal{O}(N^3 d)$):

- Cache Cholesky factor of $\mathbf{L}^+$ in first iteration.
- (Further) sparsify the weights of $\mathbf{L}^+$ with a κ-NN graph. Runtime is faster and convergence is still guaranteed.

	Cost per iteration
Objective function	$\mathcal{O}(N^2 d)$
Gradient	$\mathcal{O}(N^2 d)$
Spectral direction	$\mathcal{O}(N\kappa d)$

This strategy adds almost no overhead when compared to the objective function and the gradient computation.

Experimental Evaluation: Methods Compared

Now:

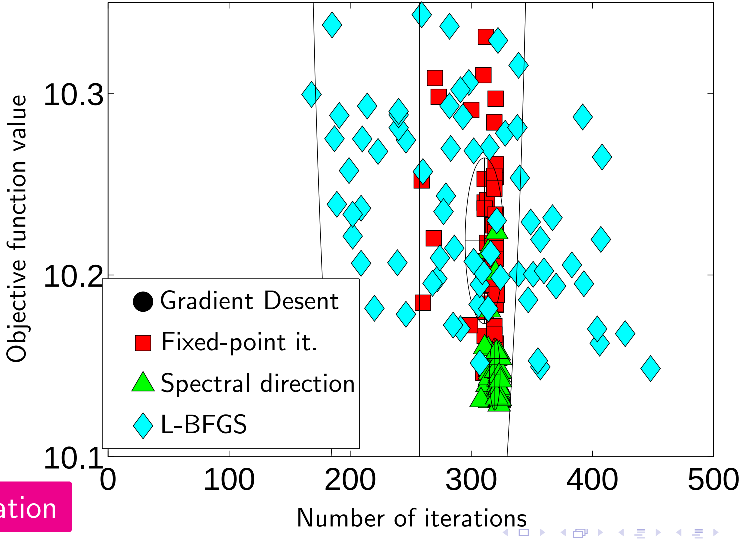
- Gradient descent (GD), $\mathbf{B} = \mathbf{I}$
  (Hinton&Roweis,'03)
- fixed-point iterations (FP), $\mathbf{B} = 4\mathbf{D}^{+} \otimes \mathbf{I}_d$
  (Carreira-Perpiñán,'10)
- Spectral direction (SD), $\mathbf{B} = 4\mathbf{L}^{+} \otimes \mathbf{I}_d$
- L-BFGS.

More experiments and methods at the poster:

- Hessian diagonal update;
- nonlinear Conjugate Gradient;
- some other interesting partial-Hessian update.

COIL-20. Convergence analysis, s-SNE

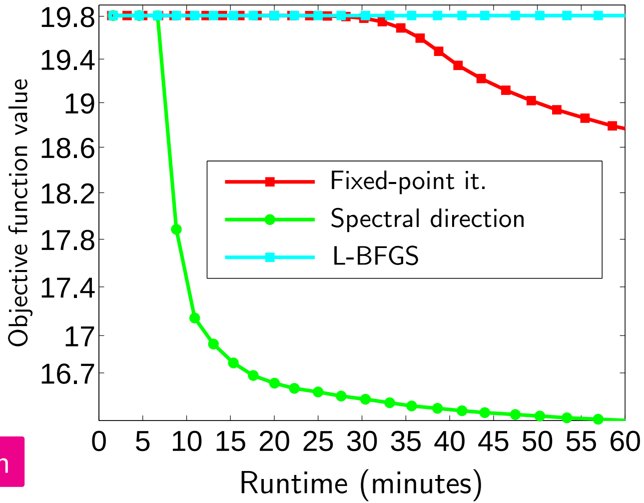
COIL-20 dataset of rotated objects ($N = 720$, $D = 16\,384$, $d = 2$).
Run the algorithms 50 times for 30 seconds each initialized randomly.



Animation

MNIST. t-SNE

- $N = 20\,000$ images of handwritten digits (each a 28×28 pixel grayscale image, $D = 784$).
- One hour of optimization on a modern computer with one CPU.



Animation

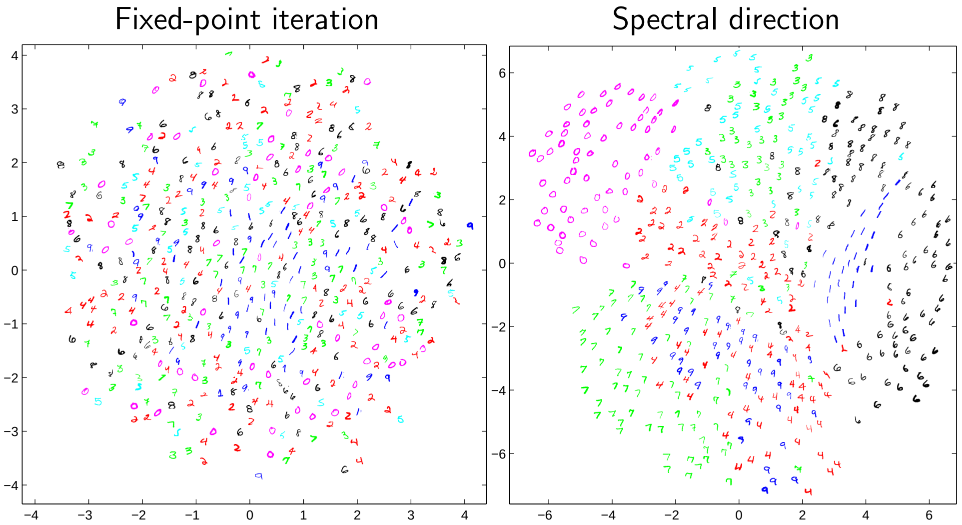
Conclusions

- We presented a common framework for many well-known dimensionality reduction techniques.
- We presented the **spectral direction**: a new simple, generic and scalable optimization strategy that runs one to two orders of magnitude faster compared to traditional methods.
- Matlab code: `http://eecs.ucmerced.edu/`.

Ongoing work:

- The evaluation of E and ∇E remains the bottleneck ($\mathcal{O}(N^2 d)$). We can use Fast Multipole Methods to speed up the runtime.
- Avoid line search, use constant, near-optimal step sizes.

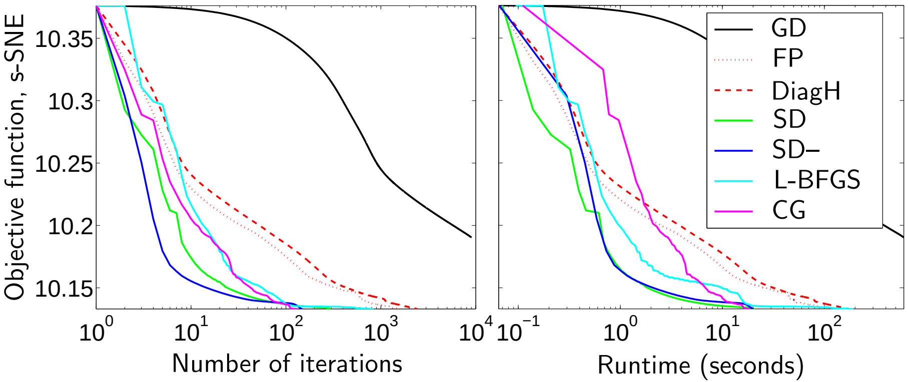
MNIST. Embedding after 20 min of EE optimization



Animation

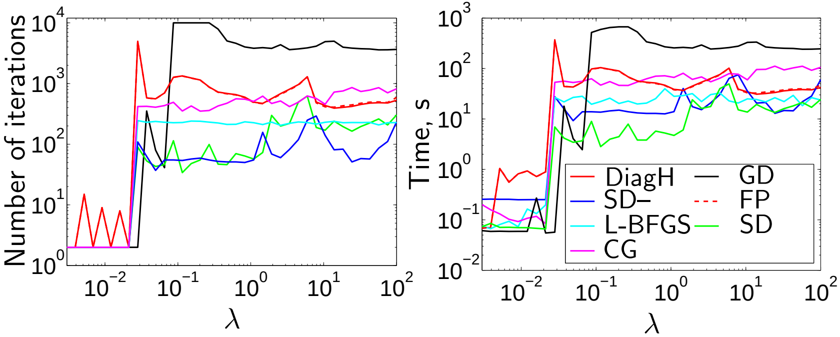
COIL-20. Convergence to the same minimum, s-SNE

We initialized $\mathbf{X}_0$ close enough to $\mathbf{X}_\infty$ so that all methods have the same initial and final points.



COIL-20: Homotopy optimization for EE

Start with small λ where E is convex and follow the path of minima to desired λ by minimizing over $\mathbf{X}$ as λ increases. We used 50 log-spaced values from 10^{-4} to 10^{2}.



Animation