Optimizing Eigen-Gaps and Spectral Functions using Iterated SDP

Tony Jebara, Blake Shaw & Andrew Howard
Columbia University

Optimization Tools for ML

Linear Programming

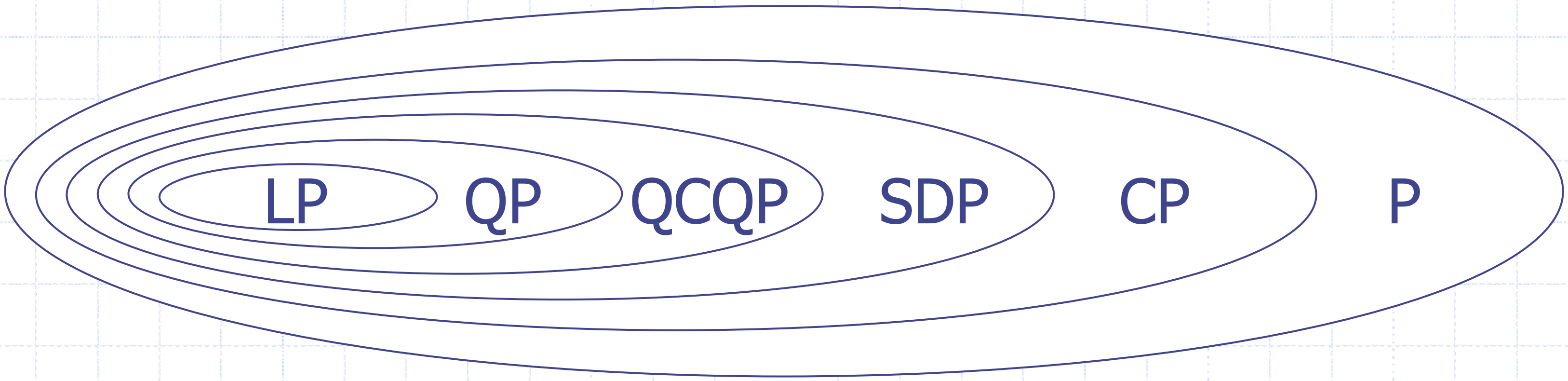
<Quadratic Programming

<Quadratically Constrained Quadratic Programming

<Semidefinite Programming

<Convex Programming

<Polynomial Time Algorithms



Optimization Tools for ML

- LP $\min_{\vec{x}} \vec{b}^T \vec{x} \;\; s.t. \; \vec{c}_i^T \vec{x} \geq \alpha_i \;\; \forall i$ fast O(N^3)
- QP $\min_{\vec{x}} \frac{1}{2} \vec{x}^T H \vec{x} + \vec{b}^T \vec{x} \;\; s.t. \; \vec{c}_i^T \vec{x} \geq \alpha_i \;\; \forall i$
- QCQP $\min_{\vec{x}} \frac{1}{2} \vec{x}^T H \vec{x} + \vec{b}^T \vec{x} \;\; s.t. \; \vec{c}_i^T \vec{x} \geq \alpha_i \;\; \forall i, \;\; \vec{x}^T \vec{x} \leq \eta$
- SDP $\min_K tr(BK) \;\; s.t. \; tr\left(C_i^T K\right) \geq \alpha_i \;\; \forall i, \;\; K \succeq 0$

...other fast SDPs in between?...

- CP $\min_{\vec{x}} f(\vec{x}) \;\; s.t. \; g(\vec{x}) \geq \alpha$ slow O(N^3)

Spectral SDPs

- Spectral function: $f(\lambda_1,\ldots,\lambda_D)$ eigenvalues of a matrix K
- SDP packages use restricted cost functions over hull kappa.

Trace SDPs $$\min_{K\in\kappa} tr(BK) \equiv \min_{K\in\kappa} \pm \sum_i \lambda_i$$

Logdet SDPs $$\min_{K\in\kappa} \sum_i -\log \lambda_i$$

- Consider richer SDPs (assume λ_i in decreasing order)

Linear-Spectral SDPs $$\min_{K\in\kappa} \sum_i \alpha_i \lambda_i$$

- Applications: a tailored alternative for trace in learning
  e.g. lower dimension visualization
  e.g. lower rank matrix factorization

Spectral SDPs

- If alphas are ordered $g(K)=\sum_i \alpha_i \lambda_i$ get a variational SDP problem (a Procrustes problem).

$$\min_{K\in\kappa} g(K)$$

$$= \min_{K\in\kappa} \sum_i \alpha_i \lambda_i \qquad s.t.\ Kv_i = \lambda_i v_i, \lambda_i \geq \lambda_{i+1}, v_i^T v_j = \delta_{ij}$$

$$= \min_{K\in\kappa} \sum_i \alpha_i \lambda_i tr\left(v_i^T v_i\right) \qquad s.t.\ Kv_i = \lambda_i v_i, \lambda_i \geq \lambda_{i+1}, v_i^T v_j = \delta_{ij}$$

$$= \min_{K\in\kappa} \sum_i \alpha_i tr\left(\lambda_i v_i v_i^T\right) \qquad s.t.\ Kv_i = \lambda_i v_i, \lambda_i \geq \lambda_{i+1}, v_i^T v_j = \delta_{ij}$$

$$= \min_{K\in\kappa} \sum_i \alpha_i tr\left(K v_i v_i^T\right) \qquad s.t.\ Kv_i = \lambda_i v_i, \lambda_i \geq \lambda_{i+1}, v_i^T v_j = \delta_{ij}$$

$$= \min_{K\in\kappa} tr\left(K \sum_i \alpha_i v_i v_i^T\right) \qquad s.t.\ Kv_i = \lambda_i v_i, \lambda_i \geq \lambda_{i+1}, v_i^T v_j = \delta_{ij}$$

$$= \begin{cases} \min_{K\in\kappa} \min_V tr\left(K \sum_i \alpha_i v_i v_i^T\right) & s.t.\ v_i^T v_j = \delta_{ij} \quad if\ \alpha_i \leq \alpha_{i+1} \\ \min_{K\in\kappa} \max_V tr\left(K \sum_i \alpha_i v_i v_i^T\right) & s.t.\ v_i^T v_j = \delta_{ij} \quad if\ \alpha_i \geq \alpha_{i+1} \end{cases}$$

If minK minV iterate SDP for K, SVD for V. Monotonic. FAST.

Spectral SDPs, Nondecreasing α

- Theorem: if alpha decreasing the linear-spectral cost $g(K) = \sum_i \alpha_i \lambda_i \quad s.t.\, \alpha_i \leq \alpha_{i+1}$ is concave.

- Proof: Recall from (Overton & Womersley '91) and (Fan '49) and (Bach & Jordan '03)

"Sum of d top eigenvalues of p.d. matrix is convex"

$$f_d(K) = \sum_{i=1}^{d} \lambda_i \Rightarrow \ convex$$

Our linear-spectral cost is a combination of these

$$g(K) = \alpha_D f_D(K) + \sum_{i=D-1}^{1} (\alpha_i - \alpha_{i+1}) f_i(K)$$

$$= \alpha_D tr(K) - \sum_{i=D-1}^{1} |\alpha_i - \alpha_{i+1}| f_i(K)$$

Trace - conic combination of convex fn's is concave.

Spectral SDPs, Nonincreasing α

- Theorem: if alpha nonincreasing the linear-spectral cost $g(K) = \sum_i \alpha_i \lambda_i \quad s.t.\, \alpha_i \geq \alpha_{i+1}$ is convex.

- Proof: Recall from (Overton & Womersley '91) and (Fan '49) and (Bach & Jordan '03)

"Sum of d top eigenvalues of p.d. matrix is convex"

$$f_d(K) = \sum_{i=1}^{d} \lambda_i \Rightarrow \; convex$$

Our linear-spectral cost is a combination of these

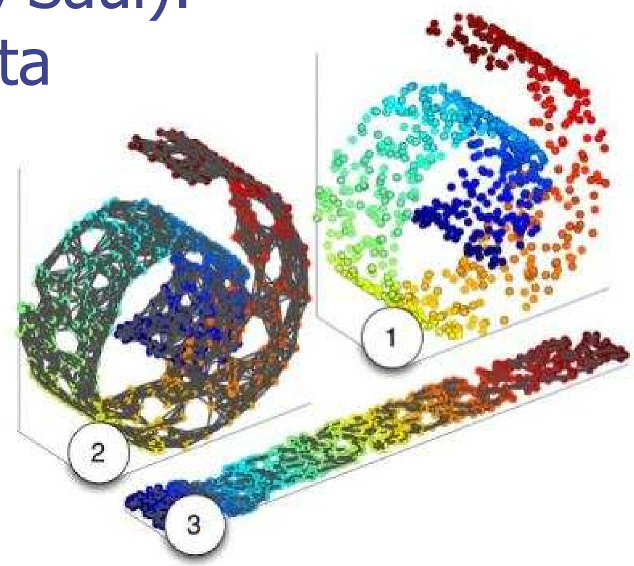
$$g(K) = \alpha_D f_D(K) + \sum_{i=D-1}^{1} (\alpha_i - \alpha_{i+1}) f_i(K)$$

$$= \alpha_D tr(K) + \sum_{i=D-1}^{1} |\alpha_i - \alpha_{i+1}| f_i(K)$$

Trace + conic combination of convex fn's is convex.

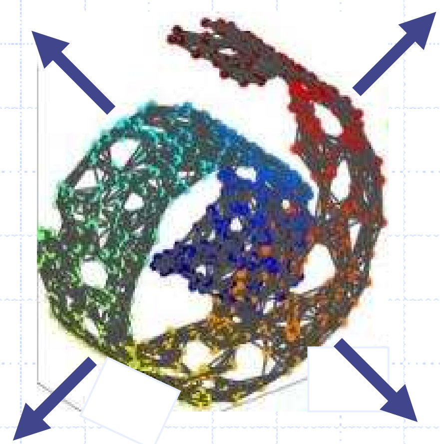
Visualizing kNN Graphs

- To visualize high-dimensional $\{x_1,\ldots,x_N\}$ data:
- PCA and Kernel PCA (Sholkopf et al):
  - -Get matrix A of affinities between pairs $A_{ij}=k(x_i,x_j)$
  - -SVD A & view top projections
- Semidefinite Embedding (Weinberger, Saul):
  - -Get k-nearest neighbors graph of data
  - -Get matrix A
  - -Use max trace SDP to stretch stretch graph A into PD graph K
  - -SVD K & view top projections



Semidefinite Embedding

- SDE unfolds (pulls apart) knn connected graph C but preserves pairwise distances when $C_{ij}=1$



$$\min_K -\sum_i \lambda_i \quad s.t.\ K \in \kappa$$

$$\kappa = \forall K \in \Re^{N\times N}$$

$$s.t.\, K \succeq 0$$

$$s.t. \sum_{ij} K_{ij} = 0$$

$$s.t.\, K_{ii} + K_{jj} - K_{ij} - K_{ji} = A_{ii} + A_{jj} - A_{ij} - A_{ji} \quad if\ C_{ij} = 1$$

- SDE's stretching of graph *improves* the visualization

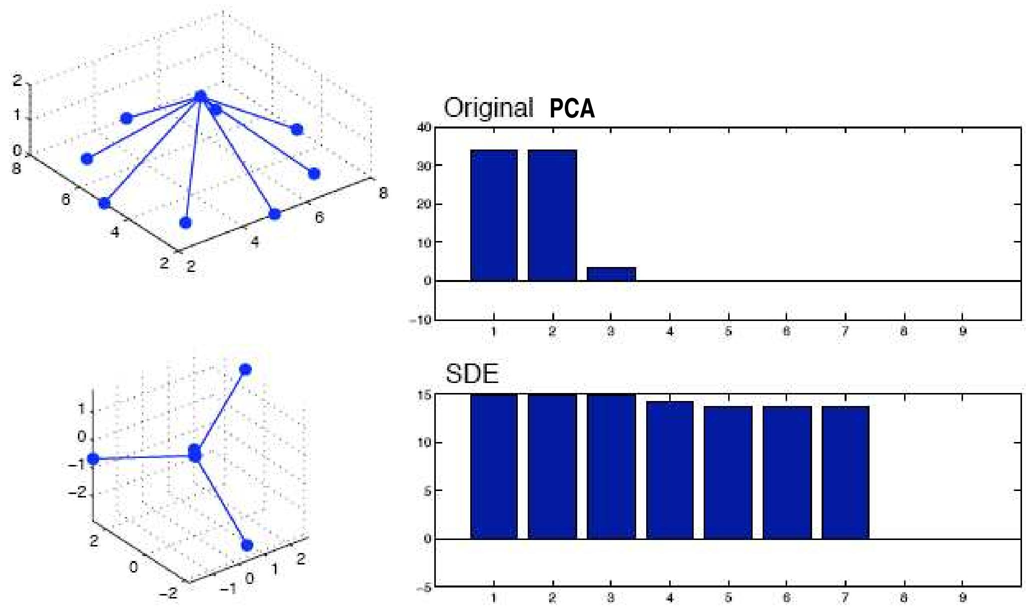
A

K

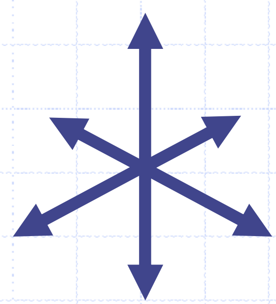
Semidefinite Embedding

- But SDE stretching could *worsen* visualization!

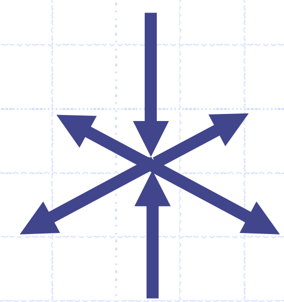
- Spokes Experiment:



- Want to pull apart only in visualized dimensions
- Flatten down remaining ones



vs.



Minimum Volume Embedding

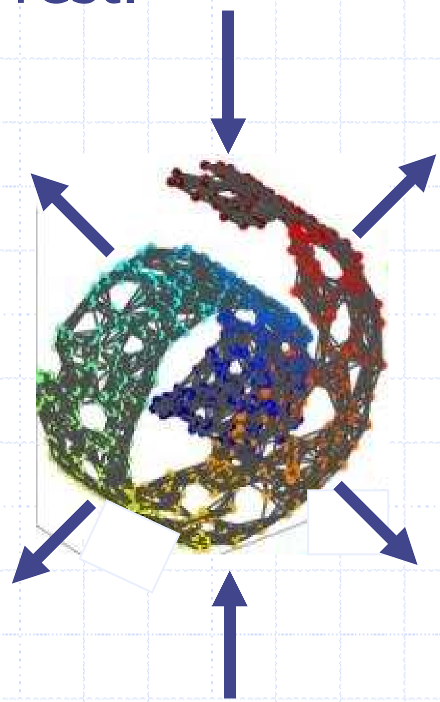
- Stretch in d<D top dimensions and squash rest.

$$\min_K -\sum_{i=1}^{d} \lambda_i + \sum_{i=d+1}^{D} \lambda_i \quad s.t.\ K \in \kappa$$

- Simplest Linear-Spectral SDP…

$$\vec{\alpha} = \begin{bmatrix} \alpha_1 & \cdots & \alpha_d & \alpha_{d+1} & \cdots & \alpha_D \end{bmatrix}$$
$$= \begin{bmatrix} -1 & \cdots & -1 & 1 & \cdots & 1 \end{bmatrix}$$

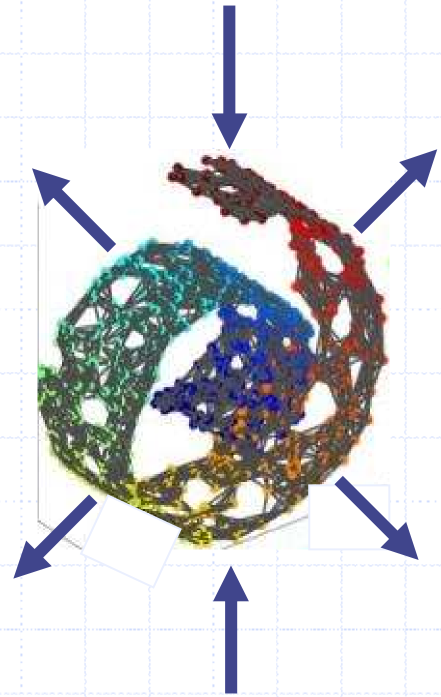
- Effectively maximizes Eigengap between d'th and d+1'th λ



Minimum Volume Embedding

- Stretch in d<D top dimensions and squash rest.

$$\min_K - \sum_{i=1}^{d} \lambda_i + \sum_{i=d+1}^{D} \lambda_i \quad s.t.\ K \in \kappa$$



- Simplest Linear-Spectral SDP…

$$\vec{\alpha} = \begin{bmatrix} \alpha_1 & \cdots & \alpha_d & \alpha_{d+1} & \cdots & \alpha_D \end{bmatrix}$$
$$= \begin{bmatrix} -1 & \cdots & -1 & 1 & \cdots & 1 \end{bmatrix}$$

- Effectively maximizes Eigengap between d'th and d+1'th λ

- Variational bound on cost à Iterated Monotonic SDP
- Lock V and solve SDP K. Lock K and solve SVD for V.

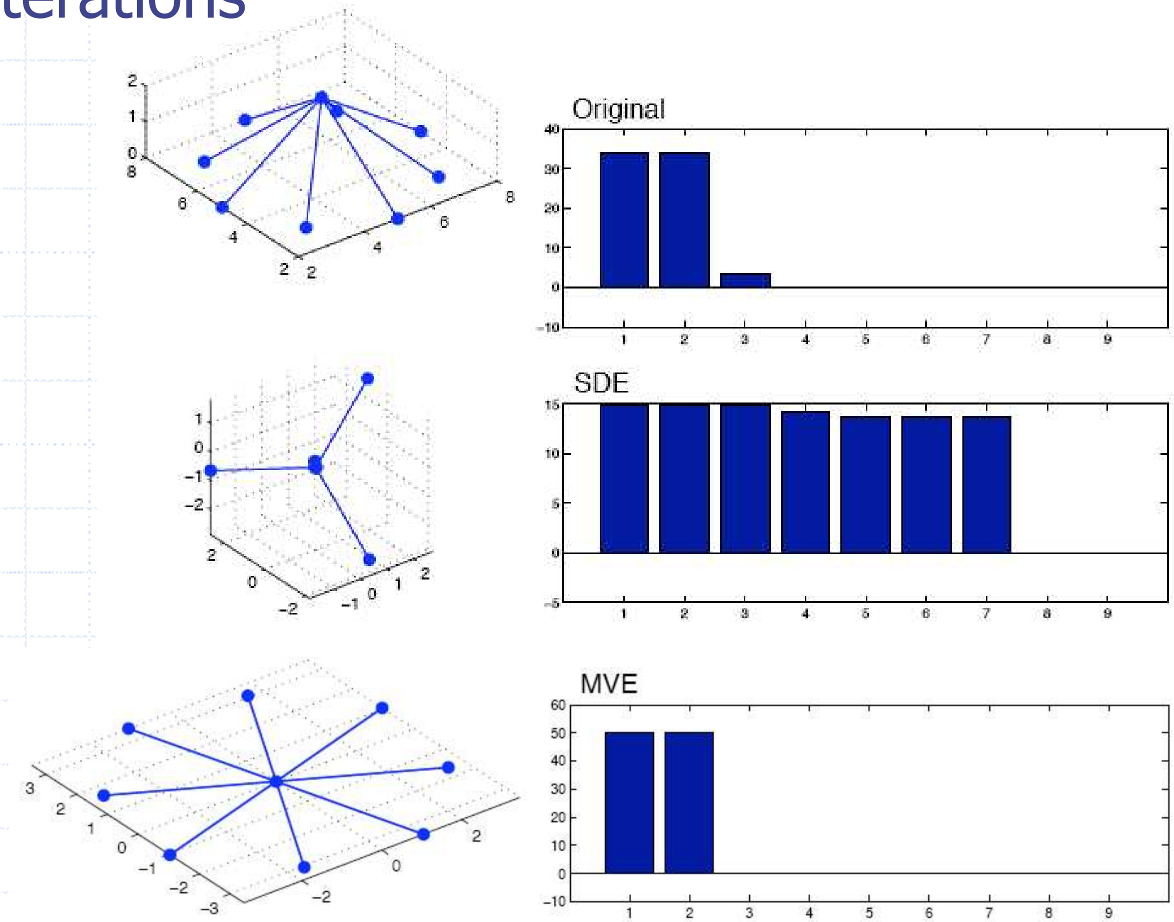
Minimum Volume Embedding

- Overall MVE algorithm:

Input	$(\vec{x}_i)_{i=1}^N$, kernel κ, and parameters d, k.
Step 1	Form affinity matrix $A \in \Re^{N \times N}$ with pairwise entries $A_{ij} = \kappa(\vec{x}_i, \vec{x}_j)$.
Step 2	Use A to find a binary connectivity matrix C via k-nearest neighbors.
Step 3	Initialize $K = A$.
Step 4	Solve for the eigenvectors $\vec{v}_1, \ldots, \vec{v}_N$ and eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$ of K.
Step 5	Set $B = -\sum_{i=1}^{d} \vec{v}_i \vec{v}_i^T + \sum_{i=d+1}^{N} \vec{v}_i \vec{v}_i^T$.
Step 6	Using SDP find $\hat{K} = \arg\min_{K \in \mathcal{K}} tr(KB)$.
Step 7	If $\|K - \hat{K}\| \geq \epsilon$ set $K = \hat{K}$, go to Step 4.
Step 8	Perform kernel PCA on K^* to get d-dimensional output vectors $\vec{y}_1, \ldots, \vec{y}_N$.

Minimum Volume Embedding

- Spokes experiment visualization and spectra
- Converges in ~5 iterations



Minimum Volume Embedding

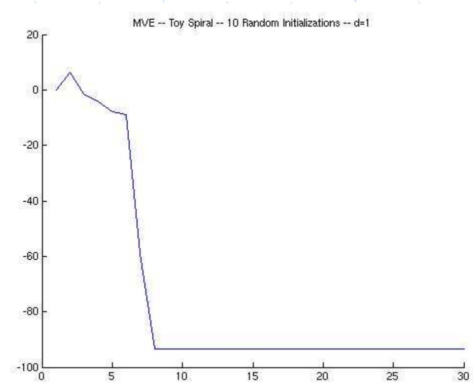
- Swissroll Visualization
  (Connectivity via knn)
  (d is set to 1)

PCA

SDE

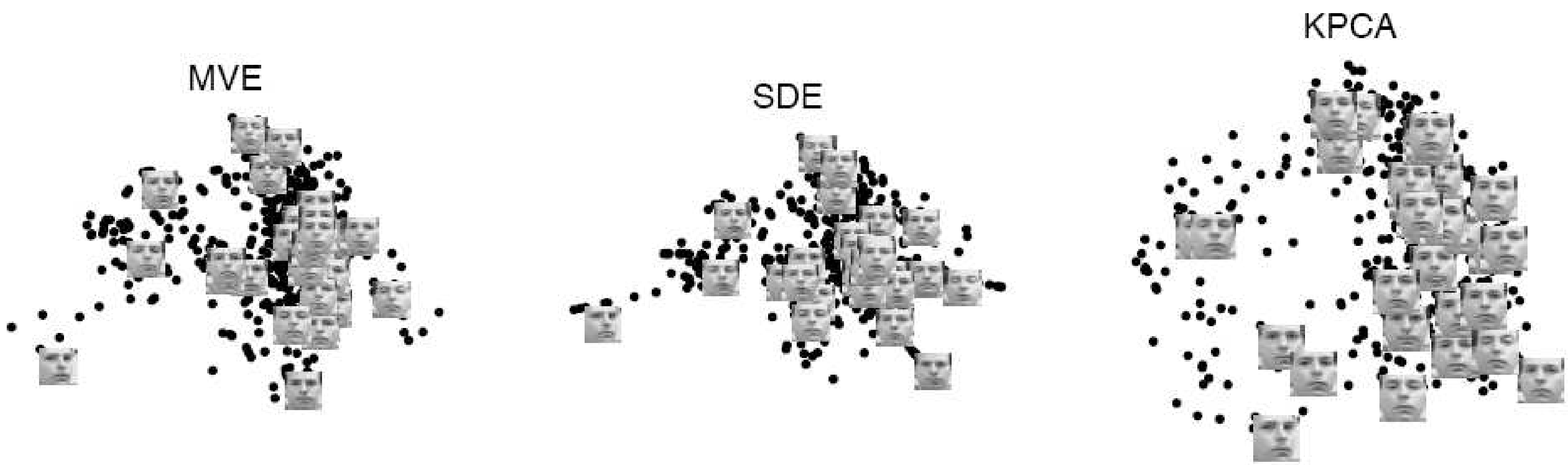
MVE

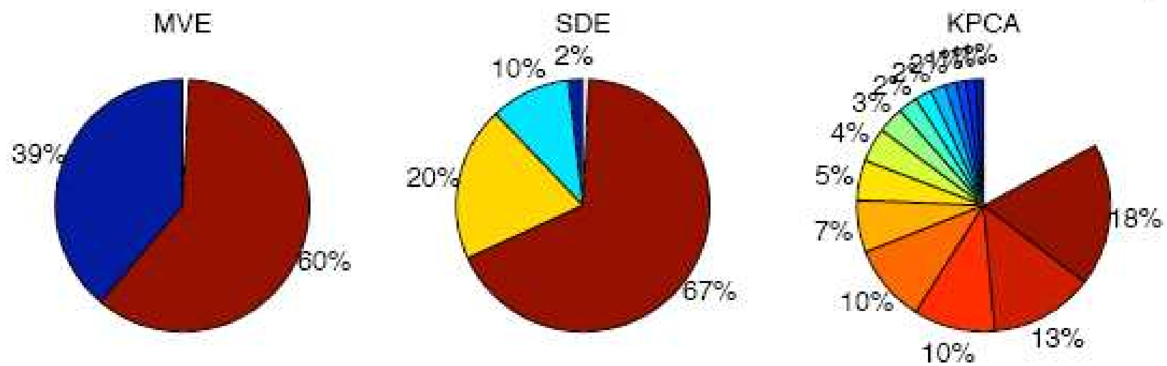
- Same convergence
  under random initialization
  or K=A…



Minimum Volume Embedding

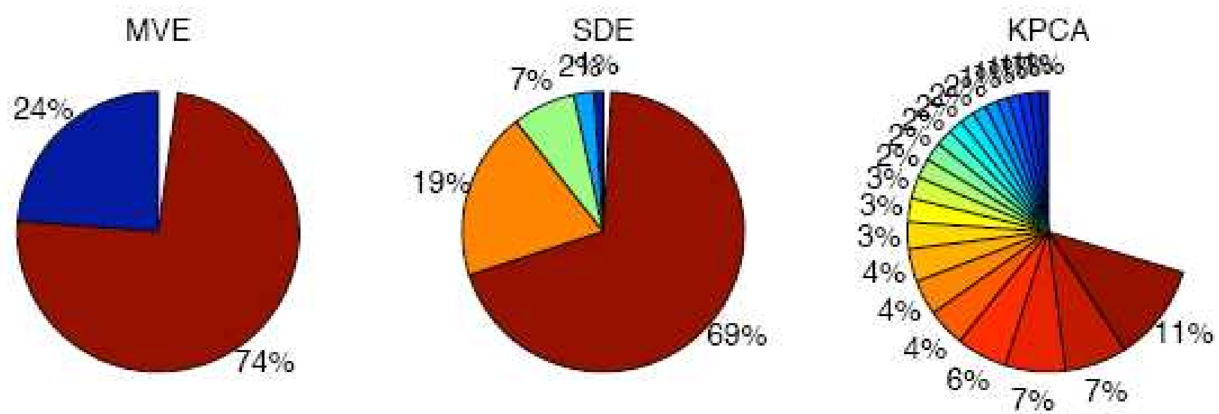
- Face Images Visualization and Spectra
  (Connectivity via knn, d=2)





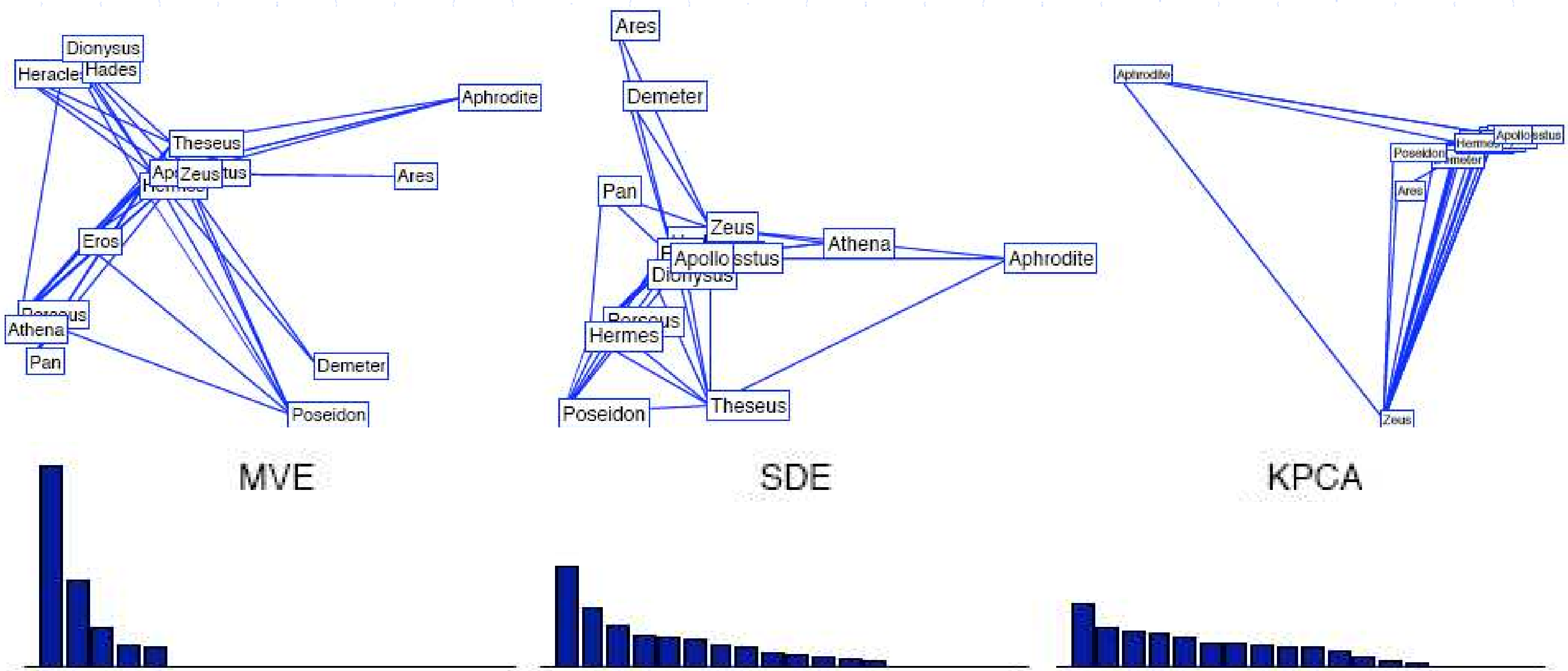
Minimum Volume Embedding

- Digit Images Visualizations and Spectra
  (Connectivity via knn, d=2)



Minimum Volume Embedding

- Social Network Visualization and Spectra
  (Connectivity from friendship links, d=2)



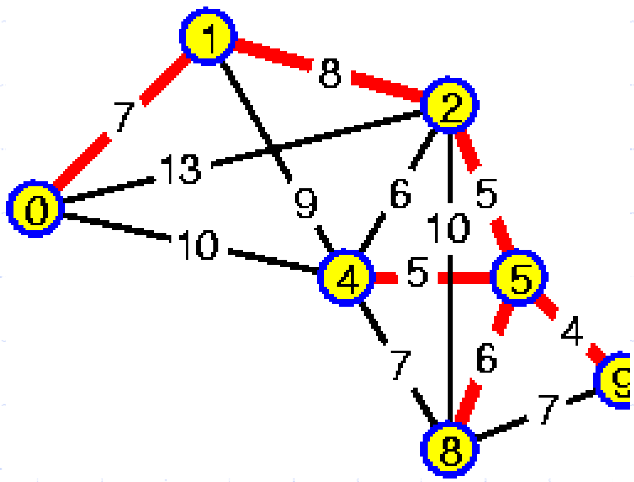
Minimum Volume Embedding

Percentage of eigenvalue energy captured in 2D

	MVE	SDE	KPCA
Hubs and Spokes	100%	29.9%	95.0%
Spiral (% in 1D)	99.9%	99.9%	45.8%
Twos	97.8%	88.4%	18.4%
Faces	99.2%	83.6%	31.4%
Social Networks	77.5%	41.7%	29.3%

Visualizing Spanning Trees

- Instead of kNN, use maximum weight spanning tree to connect points



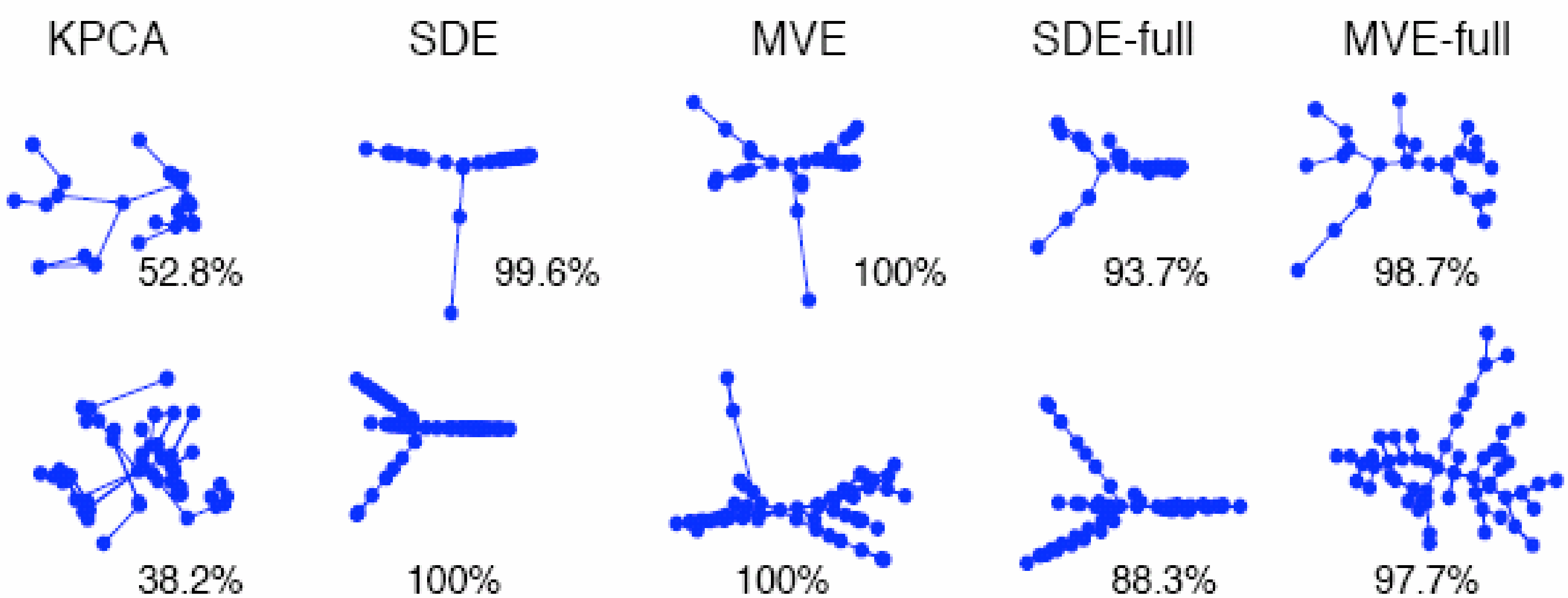
- Tree connectivity can fold over under SDE or MVE.
- Add constraints on all pairs (FULL)
- Keep all distances from shrinking

$$K \in original\,\kappa$$

$$and\;\; K_{ii} + K_{jj} - K_{ij} - K_{ji} \geq A_{ii} + A_{jj} - A_{ij} - A_{ji}$$

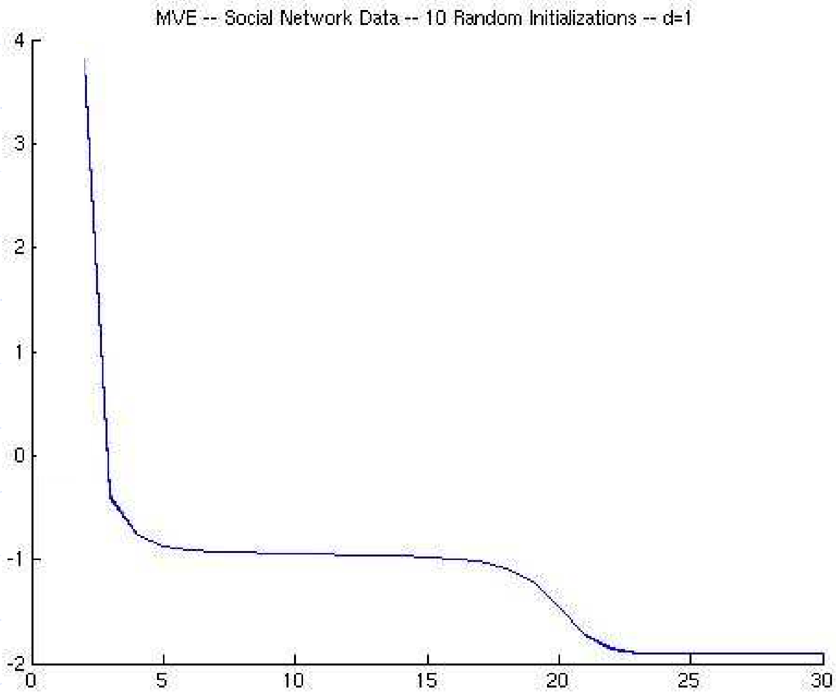
Visualizing Spanning Trees

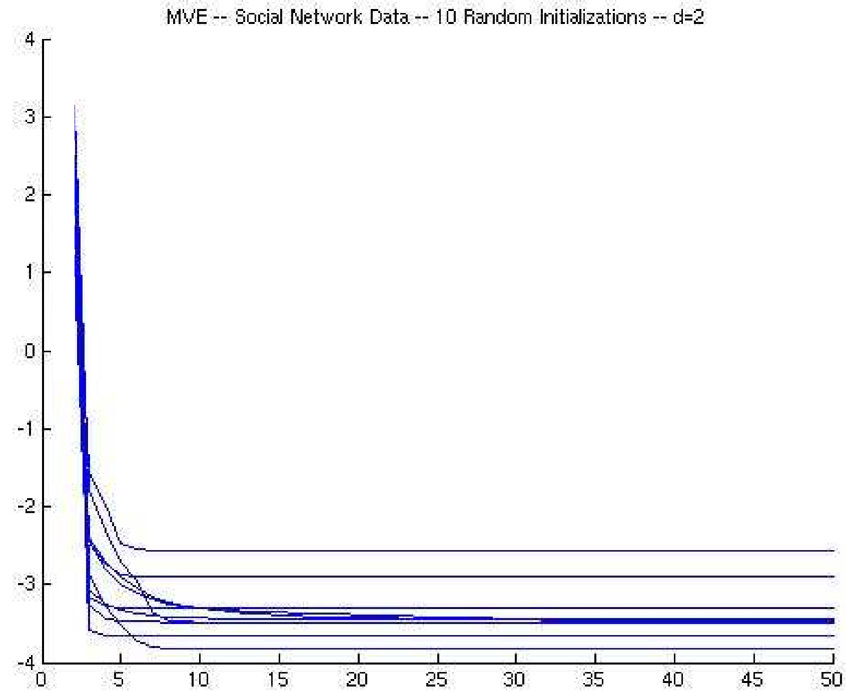
- Tree connectivity with degree=2.
- Top: phylo tree 30 species of salamanders
- Bottom: phylo tree 56 species of crustaceans



Iterated SDP Convergence

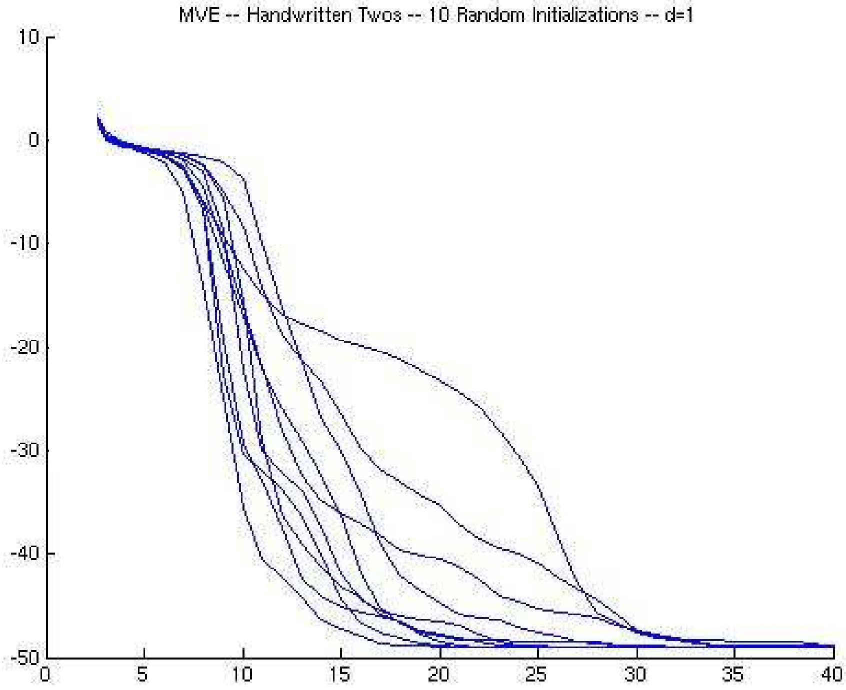
- From rand inits, iterated MVE on Social Network
  global optimum for d=1
  local optima for d=2

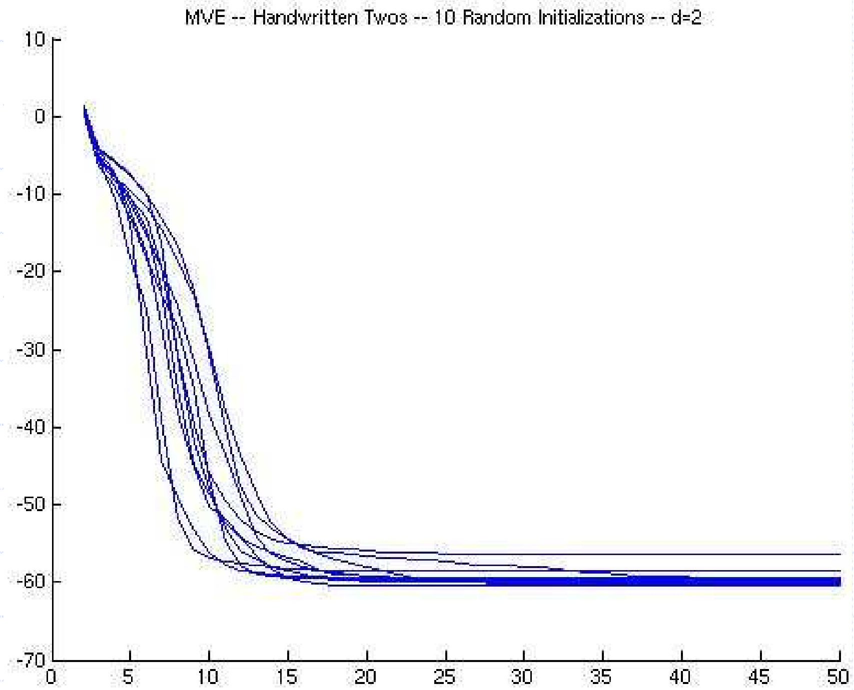




Iterated SDP Convergence

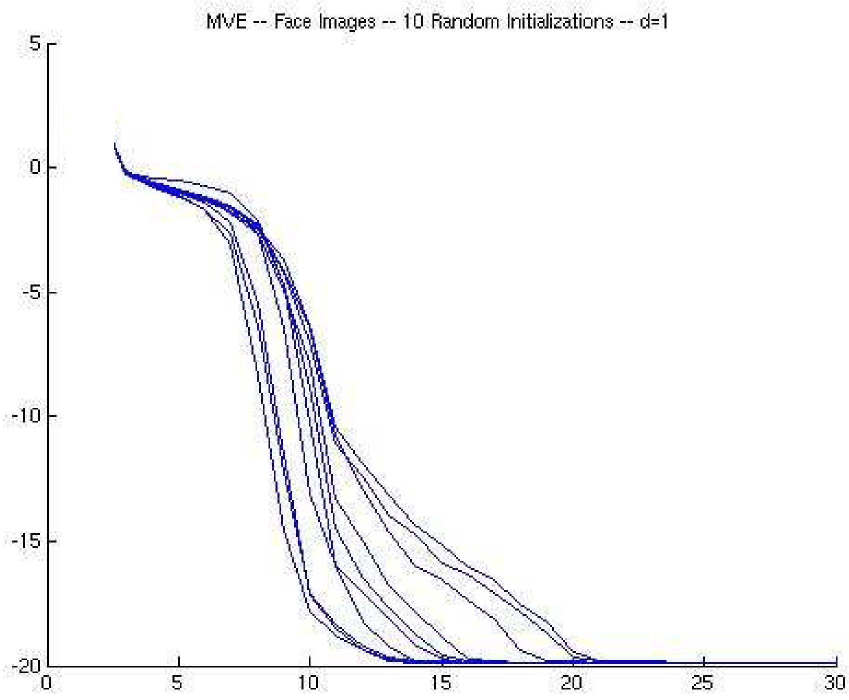
- From rand inits, iterated MVE on Digits
  global optimum for d=1
  local optima for d=2

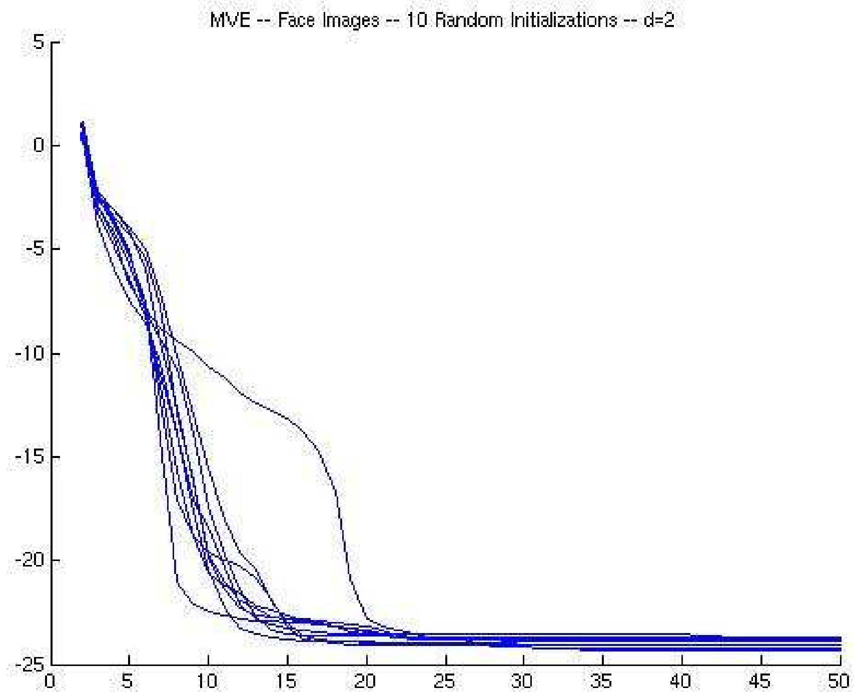




Iterated SDP Convergence

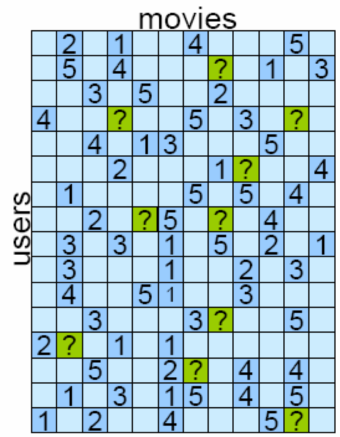
- From rand inits, iterated MVE on Face Images
  global optimum for d=1
  local optima for d=2





Max Margin Matrix Factorization

- Another ML method using trace SDP (Srebro & Jaakkola)
- Matrix completion for collaborative filtering

X = 

- Min rank of X subject to constraints χ: $\min_{X \in \chi} rank(X)$
- Approximated by envelope SDP (Fazel & Boyd)

$$\min_{Z,Y,X \in \chi} tr(K) = tr(Z) + tr(Y)$$

$$s.t.\, K = \begin{bmatrix} Y & X \\ X^T & Z \end{bmatrix} \succeq 0$$

- Also slacken constraints X with linear cost C

Spectral Matrix Factorization

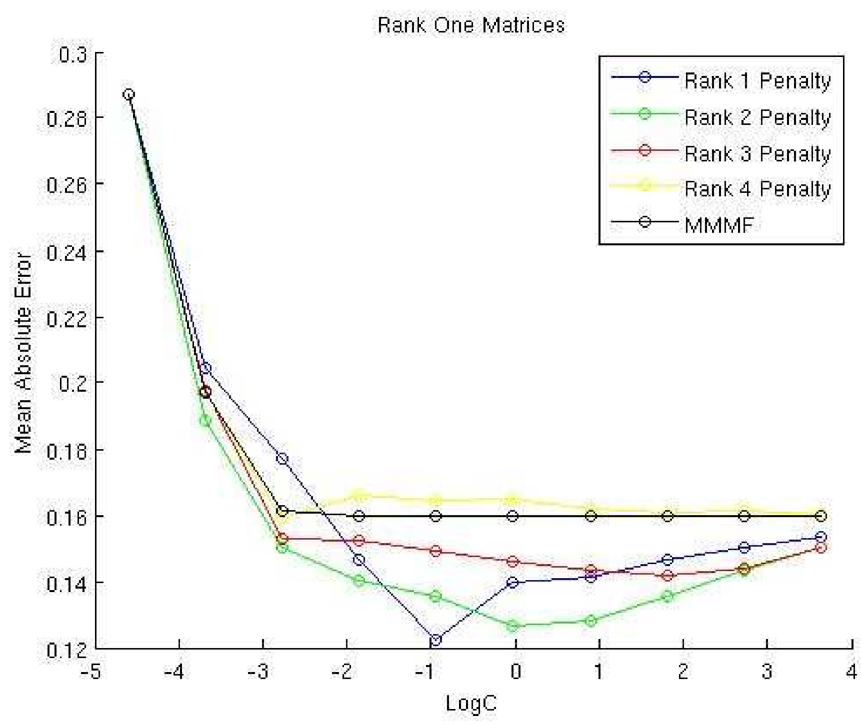
- Instead of min trace try using a linear spectral function

$$\min_{Z,Y,X\in\chi} -\sum_{i=1}^{d}\lambda_i(K)+\sum_{i=d+1}^{D}\lambda_i(K)$$

$$s.t.\, K=\begin{bmatrix} Y & X \\ X^T & Z \end{bmatrix}\succeq 0$$

- Iterated SDP

- Better reconstruction of random synthetic rank 1 matrices



Spectral Matrix Factorization

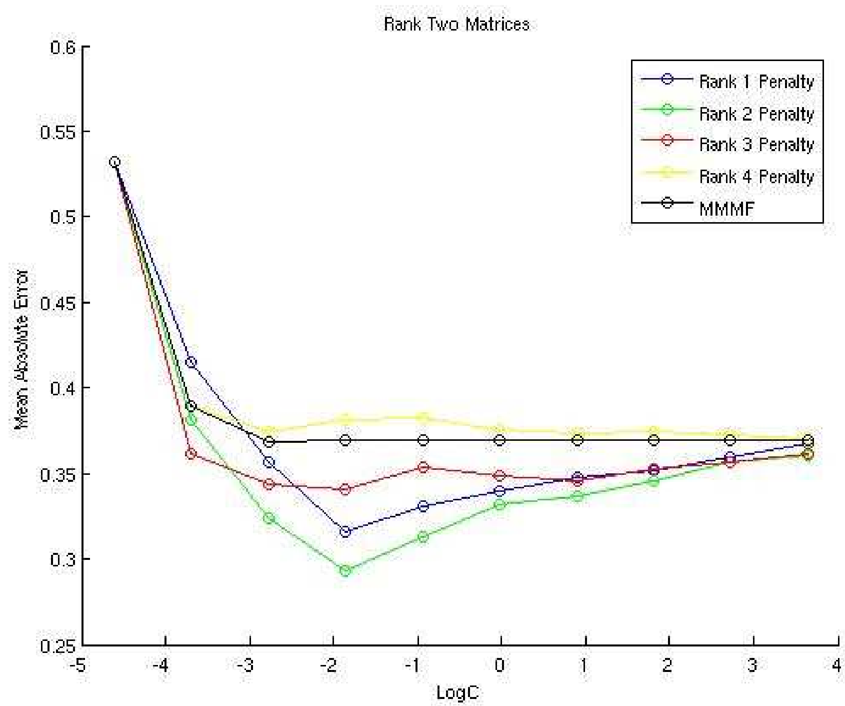
- Instead of min trace try using a linear spectral function

$$\min_{Z,Y,X\in\chi} -\sum_{i=1}^{d}\lambda_i(K)+\sum_{i=d+1}^{D}\lambda_i(K)$$

$$s.t.\, K=\begin{bmatrix} Y & X \\ X^T & Z \end{bmatrix}\succeq 0$$

- Iterated SDP

- Better reconstruction of random synthetic rank 2 matrices



Conclusions

- More general spectral cost functions in SDP
- Tailor various weights on eigenvalues

$$\min_{K \in \kappa} \sum_i \alpha_i \lambda_i$$

- Fast iterative SDP-SVD nondecreasing alpha
- Concave minimization for nondecreasing alpha
- Convex minimization for nonincreasing alpha

- Useful for optimizing (max or min) eigen-gaps
  aggressively driving dimension down
  minimum volume embedding AISTATS '07
  improving MMMF matrix factorization