MLSP2012 Tutorial:
Manifold Learning: Modeling and Algorithms

Dr. Raviv Raich (presenting)

Behrouz Behmardi

School of Electrical Engineering and Computer Science

Oregon State University, Corvallis, OR 97331-5501

Acknowledgment

- Behrouz Behmardi, PhD candidate, Oregon State University
- Dr. Alfred Hero, Prof. EECS, University of Michigan
- Dr. Kevin Carter, Lincoln Labs
- Dr. Steve Damelin, Prof. math.

Outline

- Motivation
- Mathematical Background
  - Linear models and algorithms
  - Manifolds (terminology)
- Manifold learning approaches
  - Geometric
  - Probabilistic
- New directions

Motivation

- Large volume, high dimensional data
- Dimension reduction for:
  - Visualization: insight into the dataset
  - Compression: storage
  - Denoising: remove redundant dimensions, reduce classifier complexity = improve generalization

Motivation

- **Face image dataset**:
  - Representation: a high dimensional vector where each dimension represents the brightness of one pixel.

20×28

  - Underlying structure parameters: different camera angles, pose and lighting condition, face expression, etc.

Motivation

- **Character recognition**:
  - Representation: a high dimensional vector where each dimension represents the brightness of one pixel.

28×28

  - Underlying structure parameters: orientation, curvature, style (e.g., 2 with/without loops)

Motivation

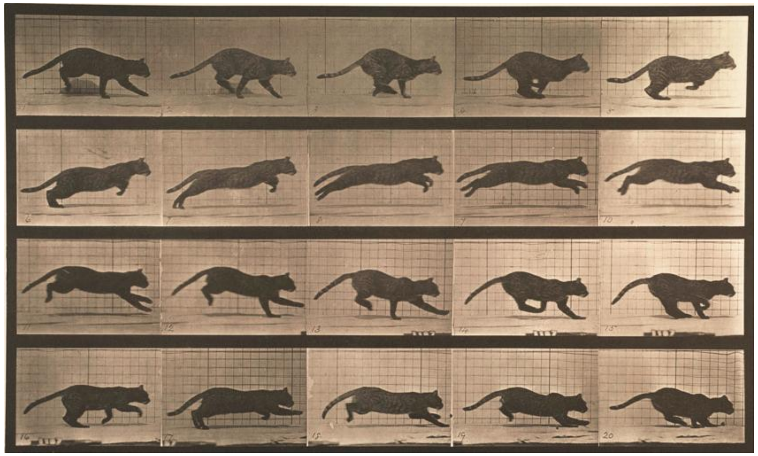
- **Text document**:
  – Representation: vector of term frequency over the dictionary of the word.

Term	D1	D2
game	1	0
decision	0	0
theory	2	0
probability	0	3
analysis	0	2
…		

  – Underlying structure parameter: topic proportions

Motivation

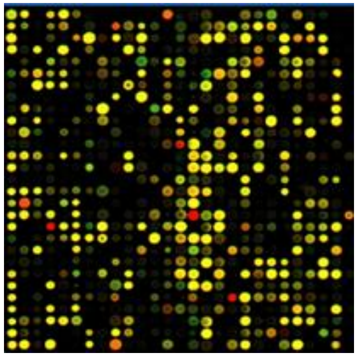
- **Motion capture**:
  - Representation: pose is determined, for example, by the 3D coordinates of multiple points on the body.



  - Underlying structure parameter: pose type
  - Motion can be viewed as a trajectory on the manifold

Motivation

- **Microarray gene expression**:
  - Representation: vector of gene expression values or sequences of such vectors.



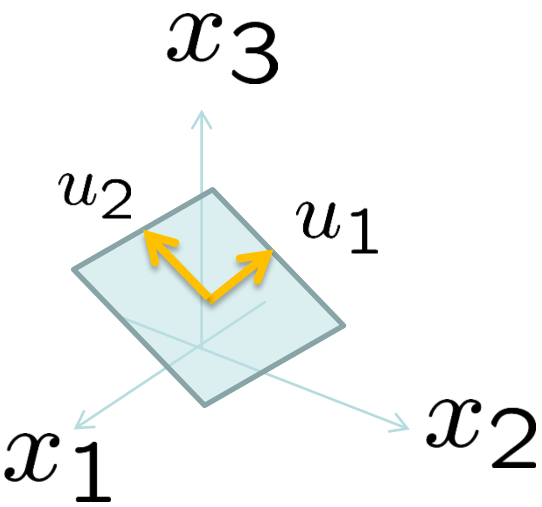
  - Underlying structure parameter: correlated (or dependent) gene groups

Motivation

- Our main goal is to discover the underlying structure of the data given the high dimensional observations.
- Real world datasets are highly nonlinear.
- It is assumed that data lie on or close to a very thin layer of a manifold embedded into the high dimensional space.

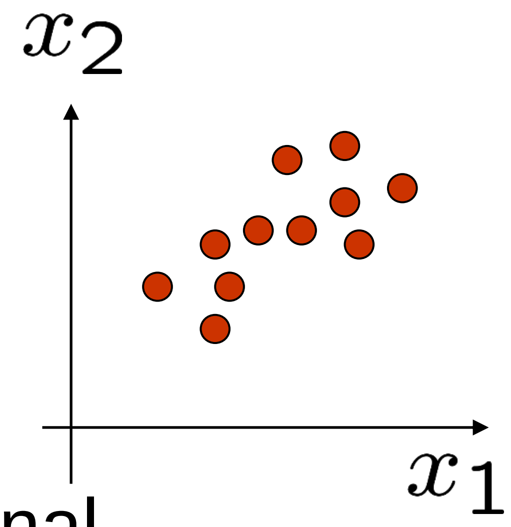
Linear Dimension Reduction

- Common assumption: data points lie on a low-dimensional plane
- Properties:



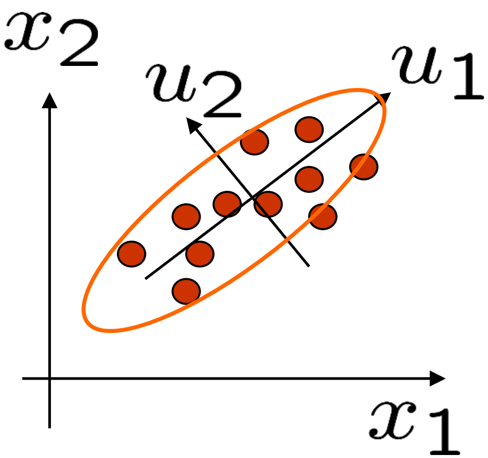
- A point x in the low-dimension plane satisfies: $x - b = \sum_{i=1}^{d} \alpha_i u_i \in \text{span}\{u_1, u_2, \ldots, u_d\}$.
- Any two point on the plane x_1, x_2 satisfy: $x_1 - x_2 \in \text{span}\{u_1, u_2, \ldots, u_d\}$.

Principle Component Analysis (PCA)



- Problem:
  - Given $\{x_1, x_2, \ldots, x_n\}$ in $\mathbb{R}^D$,
  - Find the affine transformation
    $T : \mathbb{R}^D \to \mathbb{R}^d, T(x) = Ax + b$
    that maximizes the low-dimensional transformed data variation:
    $\max_{AA^T=I} \frac{1}{n} \sum_{i=1}^n \|T(x_i) - \overline{T(x_i)}\|^2$
  - or equivalently
    $\max_{AA^T=I} \frac{1}{2n^2} \sum_{i=1}^n \sum_{j=1}^n \|T(x_i) - T(x_j)\|^2$

Principle Component Analysis (PCA)



- Equivalent formulation:
  - $\max_{AA^T=I} AC_xA^T$

    where

$$C_x = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}_i)(x_i - \bar{x}_i)^T$$

- Solution: EigenValue Decomposition (EVD)
  - $C_x = [u_1 \ldots u_D]\text{diag}(\lambda_1, \ldots, \lambda_D)[u_1 \ldots u_D]^T$

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_D$$

$$A = [u_1 \ldots u_d]^T$$

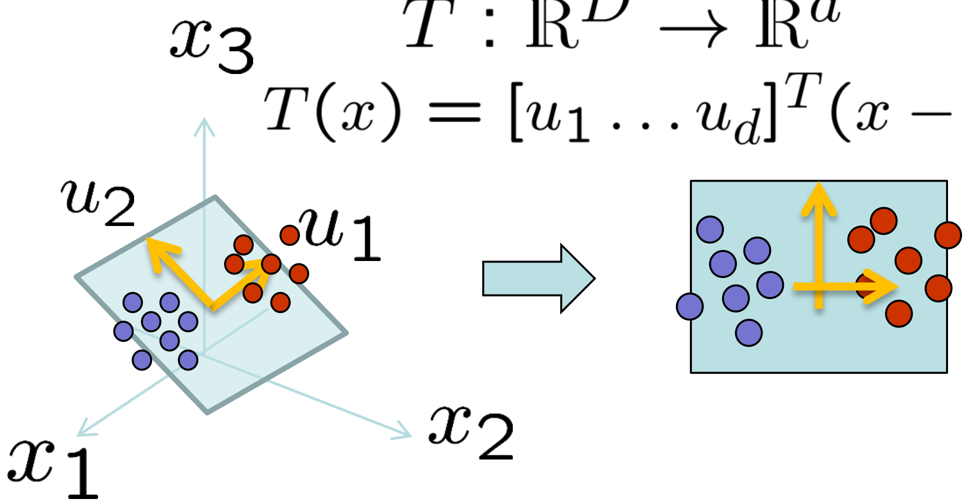
$$T(x) = [u_1 \ldots u_d]^T(x - \bar{x})$$

Principle Component Analysis (PCA)

- PCA produces an affine transformation mapping the high dimensional space into a low dimensional space.

$$T : \mathbb{R}^D \rightarrow \mathbb{R}^d$$

$$T(x) = [u_1 \ldots u_d]^T (x - \bar{x})$$

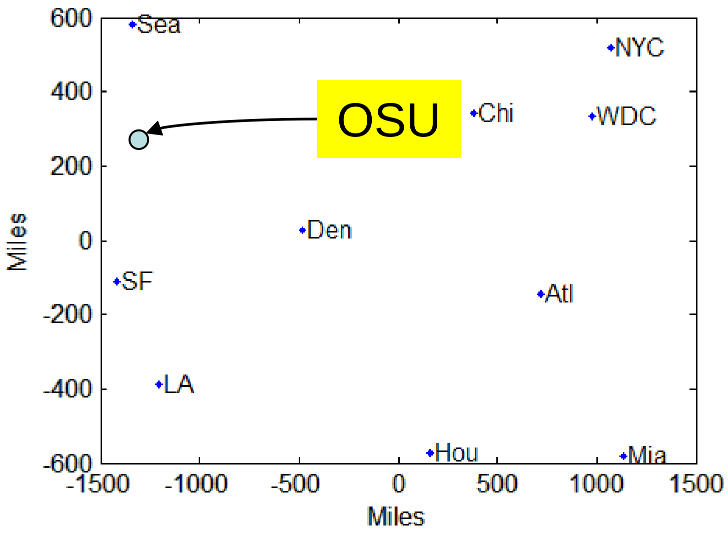


- Distance: $\|T(x_1) - T(x_2)\| \leq \|x_1 - x_2\|$
- Spectral method
- Parametric: easily extends to new point

Multidimensional Scaling (MDS)

- Construct a map of 10 US cities from their relative distances*:

```
cities =
{'Atl','Chi','Den','Hou','LA','Mia','NYC','SF','Sea','WDC'};
D = [    0  587 1212  701 1936  604  748 2139 2182   543;
       587    0  920  940 1745 1188  713 1858 1737   597;
      1212  920    0  879  831 1726 1631  949 1021  1494;
       701  940  879    0 1374  968 1420 1645 1891  1220;
      1936 1745  831 1374    0 2339 2451  347  959  2300;
       604 1188 1726  968 2339    0 1092 2594 2734   923;
       748  713 1631 1420 2451 1092    0 2571 2408   205;
      2139 1858  949 1645  347 2594 2571    0  678  2442;
      2182 1737 1021 1891  959 2734 2408  678    0  2329;
       543  597 1494 1220 2300  923  205 2442 2329     0];
```



- MDS finds the original coordinates up to rotation, translation, and axis reversal.

* numbers taken from Matlab’s website

Multi-Dimensional Scaling (MDS)

- In MDS, the goal is to obtain a set of coordinates

$$\mathcal{X}_n = [x_1, x_2, \ldots, x_n]$$

- given only the square Euclidean distances matrix $\mathcal{D}$:

$$\mathcal{D}_{ij}^2 = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2.$$

- Note that:
  - the classical MDS does not account for noise
  - MDS outputs coordinates (and not a mapping).

Multi-Dimensional Scaling (MDS)

Solution (assume $\mathcal{X}_n 1 = 0$):

- Express $\mathcal{D}$ in a matrix form:

$$\mathcal{D}_{ij}^2 = \|x_i - x_j\|^2 = \|x_i\|^2 + \|x_j\|^2 - 2x_i^T x_j$$

$$\mathcal{D}^2 = \phi 1^T + 1\phi^T - 2\mathcal{X}_n^T \mathcal{X}_n, \;\; \phi = [\|x_1\|^2, \ldots, \|x_n\|^2]^T$$

- Multiplying both sides by $\boldsymbol{P} = \boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$.

$$\Longrightarrow \quad \mathcal{X}_n^T \mathcal{X}_n = -\frac{1}{2}\boldsymbol{P}\mathcal{D}^2\boldsymbol{P}$$

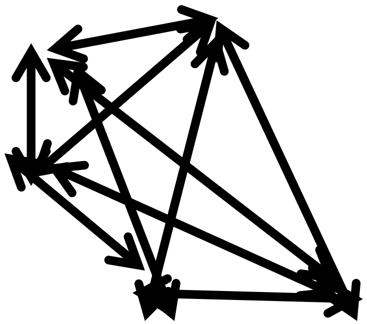
- Given the EVD of the “centered” distance matrix,

$$\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T = -\frac{1}{2}\boldsymbol{P}\mathcal{D}^2\boldsymbol{P}$$

- The resulting coordinate are $\mathcal{X}_n = \boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{U}^T$.

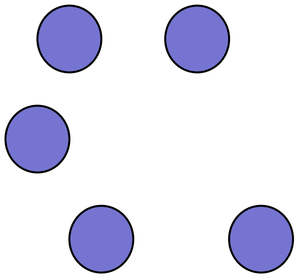
Multi-Dimensional Scaling (MDS)

- Given a set of all distances finds coordinates:



$$\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T = -\tfrac{1}{2}\boldsymbol{P}\mathcal{D}^2\boldsymbol{P}$$

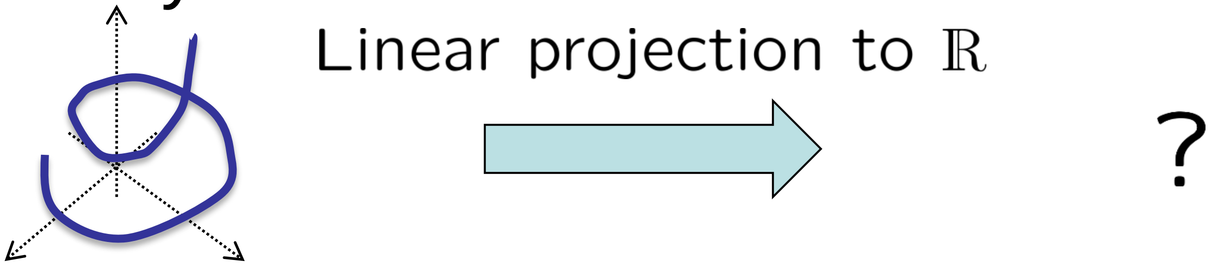
$$\mathcal{X}_n = \boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{U}^T.$$



- Non-parametric
- Requires all distances
- Generalizations:
  - stress minimization (stress majorization)
  - Euclidean distance matrix completion

Linear Dimension Reduction

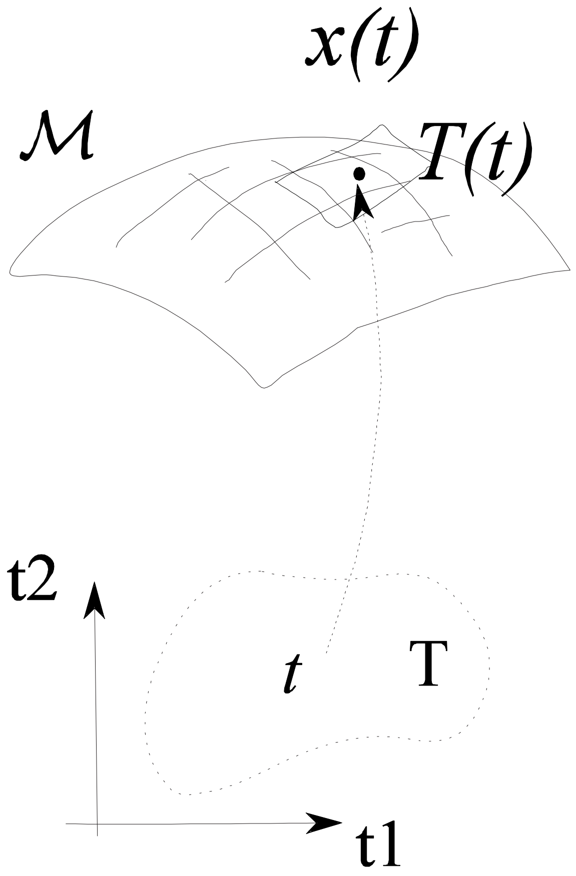
- Advantages:
  - Closed-form solutions
  - Denoising
  - Out-of-sample extension (for some methods)
- Accuracy limitation:



The EVD in PCA will not recognize the 1D structure of the curve

Manifold Learning

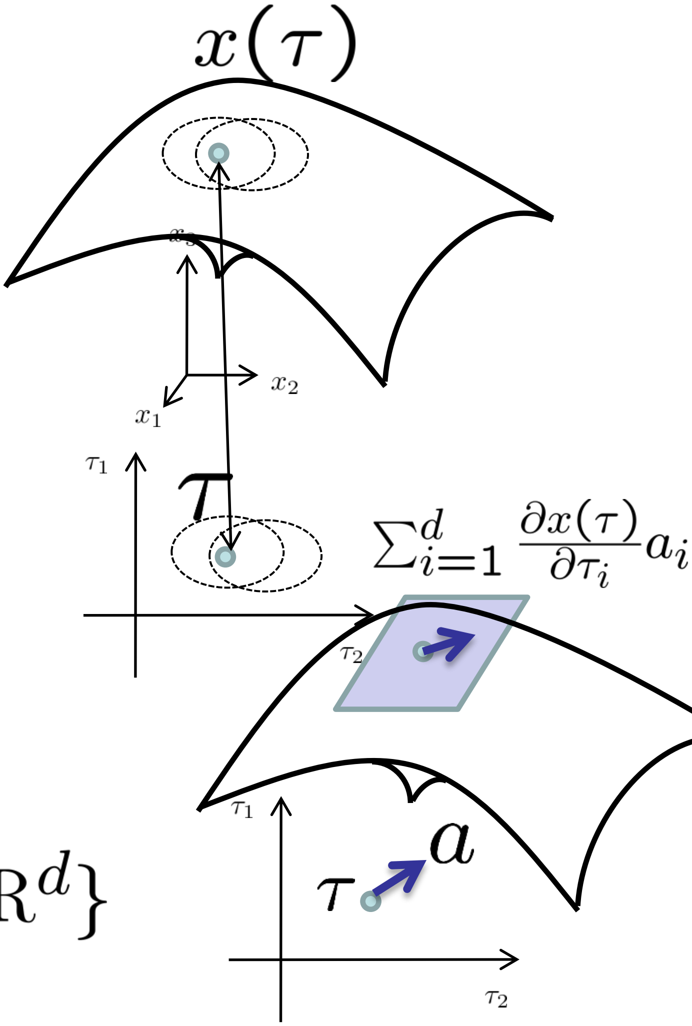
- Nomenclature:
  - Manifold
  - Local Coordinates
  - Global Coordinates
  - Tangent Plane
  - Geodesics



Informal Introduction to Manifolds

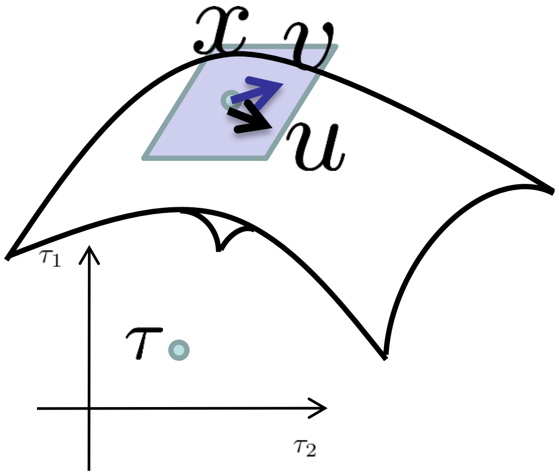
- d-dimensional differentiable manifold:
  - Can be covered with open sets which map (homomorphism) to subsets of d-dimensional Euclidean space
  - Global mapping may not exist
- Tangent space:

$$T_x\mathcal{M} = \{\sum_{i=1}^{d} \frac{\partial x(\tau)}{\partial \tau_i} a_i | a \in \mathbb{R}^d\}$$

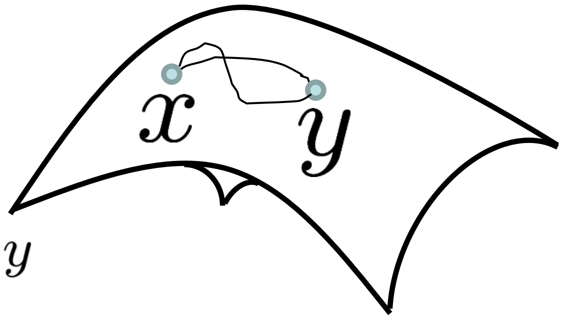


Informal Introduction to Manifolds

- d-dimensional Riemannian manifold:
  - Riemannian metric ('local inner product') is defined for any $x \in \mathcal{M}$ and $u, v \in T_x\mathcal{M}$
    $$g_x(u, v) = \langle u, v \rangle_x$$
  - Euclidean: if $u = \sum a_i \frac{\partial}{\partial x_i}$ and $v = \sum b_i \frac{\partial}{\partial x_i}$
    $$g_x(u, v) = \sum a_i b_i$$



Informal Introduction to Manifolds



- Consider a continuous path on a manifold $x(t), t \in [0,1], x(0)=x, x(1)=y$
- Path length:

$$l(x) = \int_0^1 \sqrt{g_x(\dot{x}(t), \dot{x}(t))}dt$$

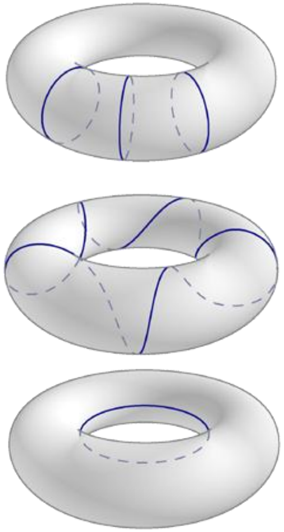
  – Using Euclidean metric

$$l(x) = \int_0^1 \|\dot{x}(t)\| dt$$

  – Geodesic distance:

$$d(x_1, x_2) = \inf_{x(\cdot)} l(x)$$

  – Geodesic: the shortest path (assuming the manifold is geodesically-convex) $\nabla_{\dot{x}} \dot{x} = 0$

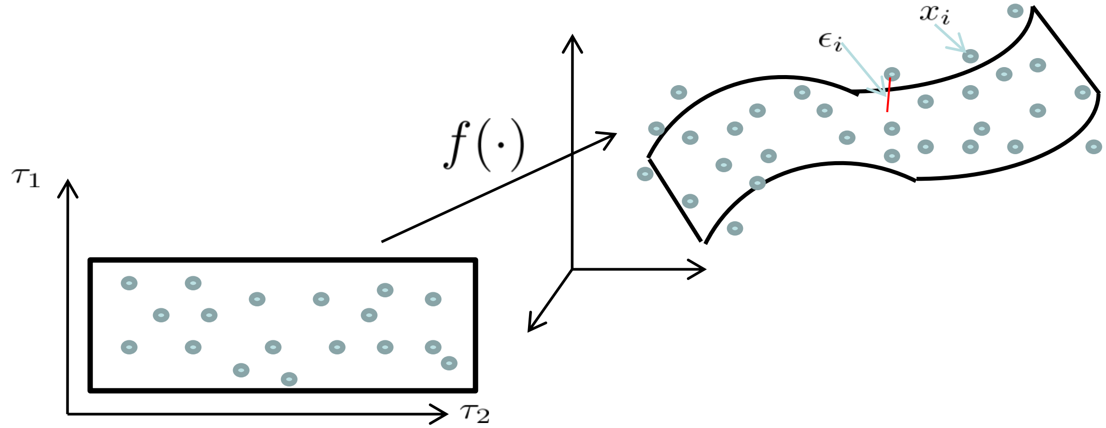


*From Mark Iron's website

What is manifold learning?

- A d dimensional manifold $\mathcal{M}$ is embedded in an m dimensional space, and there is an explicit mapping $f : \mathcal{R}^d \rightarrow \mathcal{R}^m$ where $d \leq m$. We are given samples $x_i \in \mathcal{R}^m$ with noise .

$$x_i = f(\tau_i) + \epsilon_i$$

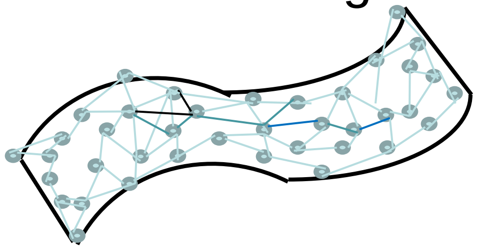


- $f(\cdot)$ is called *embedding function*, m is the *extrinsic dimension*, d is the *intrinsic dimension* or dimension of the latent space.
- Finding either $f(\cdot)$ or from given x_i is called *manifold learning*.
- We don't have any information about the function $f(\cdot)$, distribution of the data in low dimension τ_i , and the distribution of the noise.
- We assume $p(\tau)$ is *smooth*, is distributed *uniformly*, and noise is *small*.

Approaches in manifold learning

Parametric vs. non-parametric

- In the **non-parametric** approach we recover τ_i directly from x_i.
- We construct a *neighborhood graph* of the data, where each vertices of the graph is the data point in the high dimension and each edge indicates the neighborhood relation.
  – k-nearest neighbors (kNN)
  – ϵ - ball



- A neighborhood graph can be seen as a discrete approximation to a smooth manifold.
- Cannot be trivially generalized to the space of the data.

Approaches in manifold learning

Parametric vs. non-parametric

- In the **parametric** approach, we find the explicit mapping $f(\cdot)$ from the given sample x_i .
- Most of the approaches are probabilistic (latent factor modeling).
- We can *generalize to the space* of the data where there is no samples.
- There is no closed form solution for these algorithms and they prone to *local optimum*.
- To have a coherent, single global low dimensional coordinate, we need to take a further step and implement the process of *coordinate alignment*.

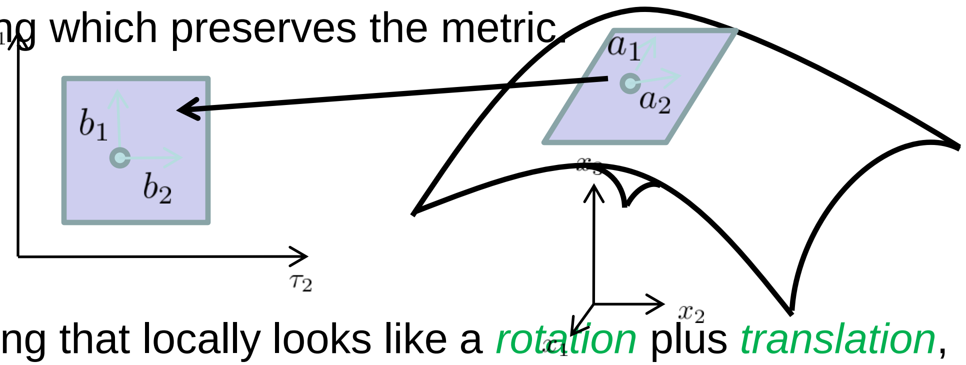
– Mixture of factor analyzers [Ghahramani et al'97].

Approaches in manifold learning

Isometric vs. non-isometric

- **Isometric** embedding is a mapping which preserves the metric.

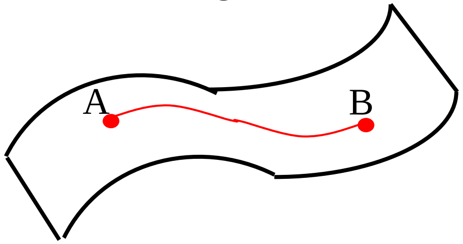
$$\langle a_1, a_2 \rangle = \langle b_1, b_2 \rangle$$



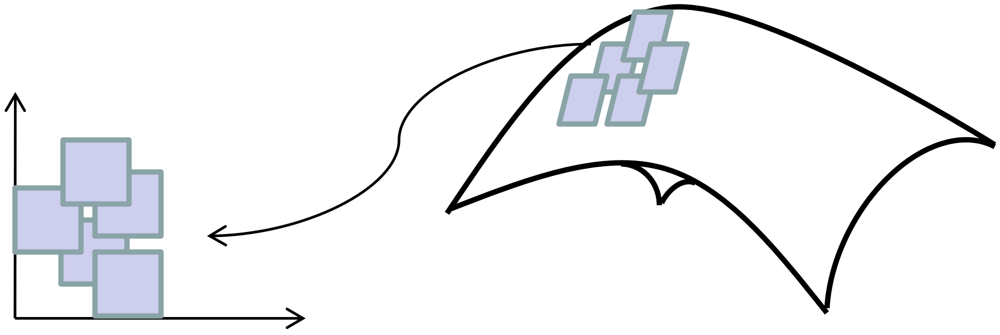
- Intuitively, an isometry is a mapping that locally looks like a *rotation* plus *translation*, thus preserving distances and angles among the vectors.
- ISOMAP [Tenenbaum et al'00], Maximum variance unfolding [Weinberger et al'04].
- **Non-isometric** embedding generally divides into two categories:
  - *Neighborhood preserving mapping* which preserve the neighborhood relations among the data points such as locally linear embedding (LLE), Laplacian eigenmap (LE) [Belkin et al'03].
  - *Conformal mapping* which is a mapping up to rotation, translation, and rescaling. It preserves the angles among the data points as well as neighborhood relations such as conformal ISOMAP [Sha et al'05].

Approaches in manifold learning

- **Global vs. local**
  - In the **global** preserving approaches, we preserve the global geometry properties of the manifold such as geodesic distance (ISOMAP) [Tenenbaum et al'00].



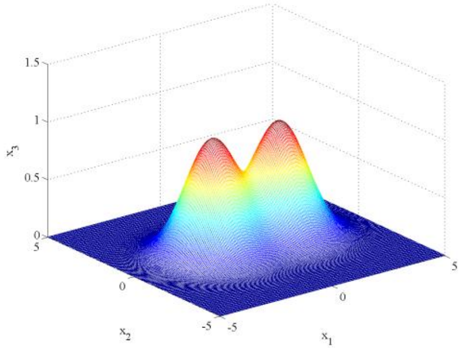
  - **Local** preserving approaches rely on the fact that the surface of any manifold can be locally approximated by its tangent space.



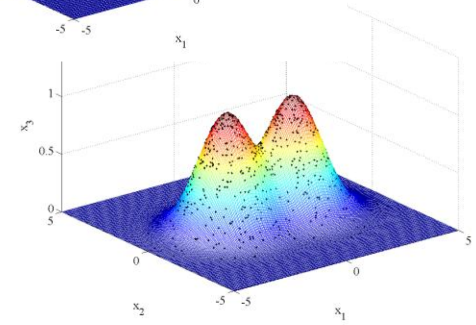
- Overlapping consensus of local geometry information can be used to find a single global low dimensional embedding.

From a Manifold to a Graph

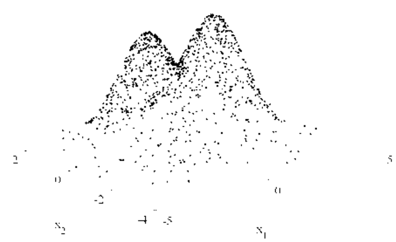
1. Consider Manifold $\mathcal{M}$.



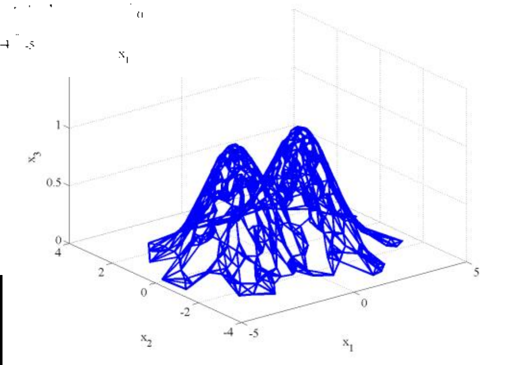
2. Data points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ $(\boldsymbol{x}_i \in \mathcal{M})$ are obtained from $\mathcal{M}$.



3. Given only the data,

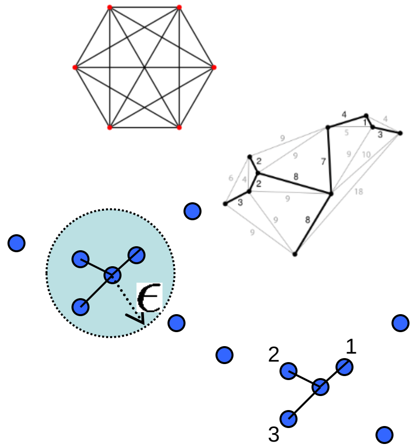


4. Construct a graph $G = (V, E)$ with a vertex set $V = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ and an edge set $E = \{e_1, e_2, \ldots, e_n\}$, where $e_k = (x_i, x_j) \in E$ if x_i and x_j are connected.



Graphs on a Manifold

- Graphs (proximity graphs)
  – Complete graph
  – Minimum spanning tree (MST)
  – $\epsilon-$ball graph
  – K-nearest neighbors graph



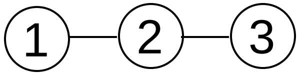
- Why? Proximity graphs offer description of local geometry.
- Global similarity via local similarities.

Unweighted Graphs Representation

- Representation:
  - Vertices: WLOG $\{1, 2, \ldots, n\}$.
  - The edge information (connectivity) is recorded by the ***adjacency matrix***

$$[\boldsymbol{A}]_{i,j} = \begin{cases} 1 & (i,j) \in E \\ 0 & (i,j) \notin E \end{cases}$$

  - The ***degree*** of a vertex is the number of vertices connected to it: $d_i = \sum_{j=1}^{n} A_{ij}$.
  - ***Graph Laplacian***: $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{D} = \mathrm{diag}\{[d_1, d_2, \ldots, d_n]\}$.
  - Normalized graph Laplacian: $\mathcal{L} = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{L} \boldsymbol{D}^{-\frac{1}{2}}$.



$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$[d_1, d_2, d_3] = [1, 2, 1]$$

$$\boldsymbol{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{L} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

$$\mathcal{L} = \begin{bmatrix} 1 & -\frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & 1 & -\frac{1}{\sqrt{2}} \\ 0 & -\frac{1}{\sqrt{2}} & 1 \end{bmatrix}$$

Weighted Graphs

- Weighted graphs: the *adjacency matrix* is given by

$$[\boldsymbol{A}]_{i,j} = \begin{cases} w_{ij} & (i,j) \in E \\ 0 & (i,j) \notin E. \end{cases}$$

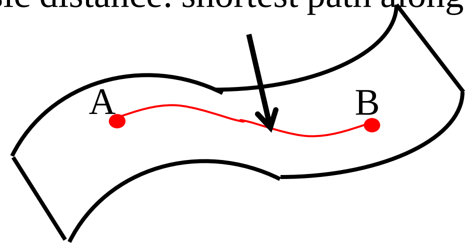
- The weights w_{ij} define the graph.
- For example: Consider the distance matrix whose ij-th element is given by $[\mathcal{D}]_{ij} = d(\boldsymbol{x}_i, \boldsymbol{x}_j)$, e.g., if $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^m$ $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 = \sqrt{\sum_{k=1}^{m} (\boldsymbol{x}_i(k) - \boldsymbol{x}_j(k))^2}$.
- The corresponding, weight matrix could be constructed using a kernel, e.g., $w_{ij} = \exp(-\mathcal{D}_{ij}^2/(2\epsilon))$.

- The weights here satisfy $0 \leq w_{i,j} \leq 1$ (special case $\mathcal{D}_{ij} \in \{0, \infty\}$ - unweighted graph).

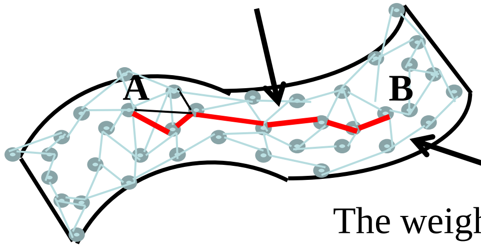
ISOMAP

- **[Tenenbaum et al., 2000]**
  - **General idea**:
    - Approximate the geodesic distances by shortest graph distance.
    - MDS using geodic distances

Geodesic distance: shortest path along the manifold



Graph approximation for geodesic distance.
Shortest path on the graph.



The weights on the edges are
Euclidean distance.

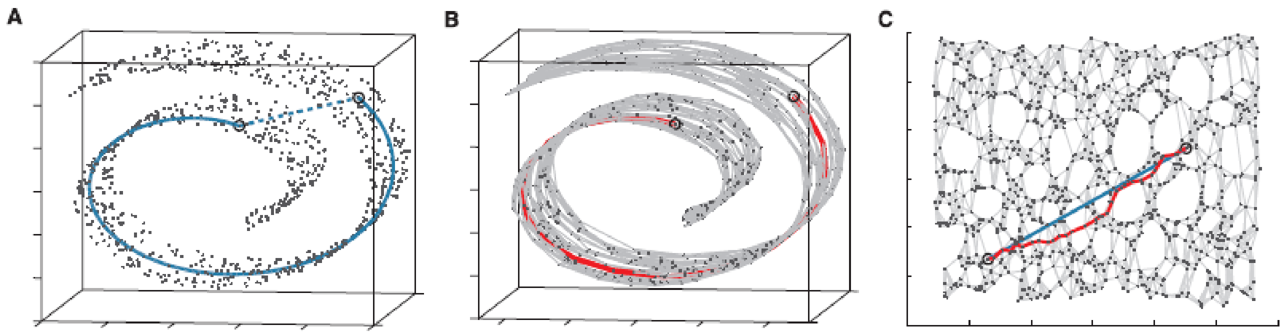
  - ISOMAP provides an *isometric embedding*. *Computational complexity* is high *($O(N^3)$).* It fails for a *non-convex region* dataset because of the convexity properties of the geodesic distance.
  - Variations: *Landmark* ISOMAP, Conformal ISOMAP [Silva et al'03].

ISOMAP

- **[Tenenbaum et al., 2000]**
- Algorithm:
  - Construct a neighborhood graph $w_{ij} \in \{0, 1\}$
  - Construct a distance matrix

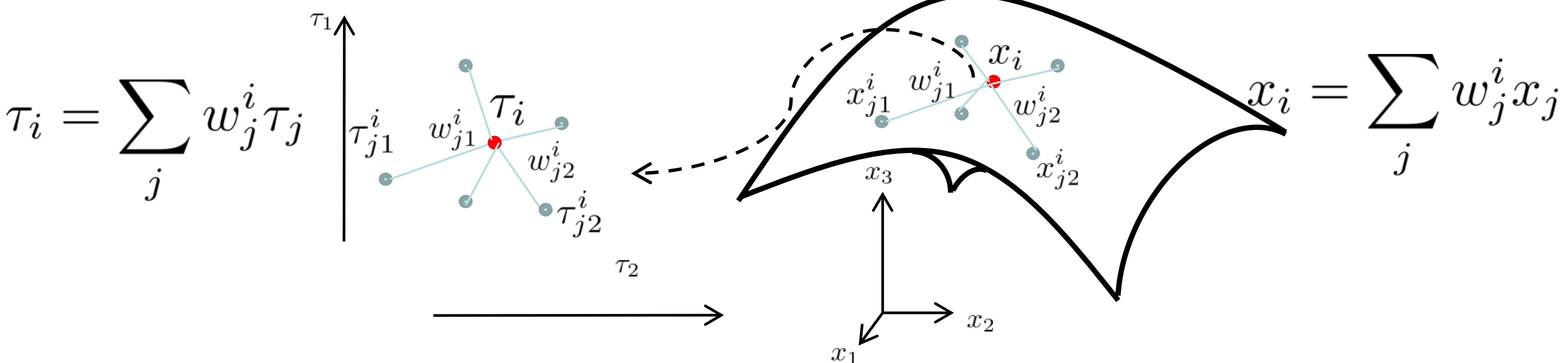
$$d_{ij} = \begin{cases} \|x_i - x_j\| & w_{ij} = 1 \\ \infty & w_{ij} = 0 \end{cases}$$

  - Find the shortest path between every i and j (e.g. using Floyd-Marshall) and construct a new distance matrix such that $\mathcal{D}_{ij}$ is the length of the shortest path between i and j.
  - Apply MDS to matrix to find coordinates



Locally linear embedding (LLE)

- **[Roweis &Saul’00]**
  - **General idea:** represent each point on the local linear subspace of the manifold as a linear combination of its neighbors to *characterize the local neighborhood relations*. Then use the same linear coefficient for embedding to preserve the neighborhood relations in the low dimensional space.



  - Compute the coefficient *w* for each data point by solving a constraint least square problem.
  - It is *easy* to implement and *computationally is efficient* ($O(pN^2)$). It is *unstable* due to the ill-posed condition in solving the least square problem.

Locally Linear Embedding

- Find weight matrix W of linear coefficients:

$$\mathcal{E}(W) = \sum_i \left| \vec{X}_i - \sum_j W_{ij} \vec{X}_j \right|^2$$

$$\min_W \varepsilon(W) \text{ s.t. } \sum_j w_{ij} = 1.$$

- Find low dimensional embedding Y that minimizes the reconstruction error

$$\Phi(Y) = \sum_i \left| \vec{Y}_i - \sum_j W_{ij} \vec{Y}_j \right|^2$$

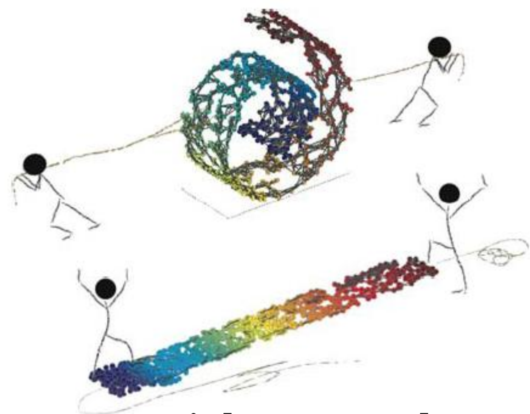
$$\min_Y \Phi(W) \text{ s.t. } YY^T = I.$$

- Solution: Eigendecomposition of

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

Maximum variance unfolding (MVU)

- **Weinberger et al., 2004**
  - **General idea**: maximize the spread of the data in the low dimensional space while preserve the distance among all the data points locally.
  - Intuitively, we connect the neighborhoods by rigid rods that fix angles and distance and then pull it as far apart as possible.



Weinberger et al., 2004

$$\max \sum_{i,j} \| \tau_i - \tau_j \|^2$$

$$\text{s.t. } \| \tau_i - \tau_j \|^2 = \| x_i - x_j \|^2,$$ j is in the nbhd of i

$$\sum_i \tau_i = 0$$

  - This is a non-convex optimization problem.
  - Formulate the problem as a convex semidefinite program.
  - This is an *isometric embedding* approach. Computationally is *complex* $O((kN)^3)$.
  - Variation: *landmark* MVU [Weinberger et al'04]

Maximum variance unfolding (MVU)

- Solution:
  - Construct a nbhd graph
  - Let K be the Gram matrix: $K_{ij} = \tau_i^T \tau_j$

$$\max \operatorname{tr}(K)$$

s.t.

$$K_{ii} + K_{jj} - K_{ij} - K_{ji} = \|x_i - x_k\|^2 \text{ for all } j \text{ in nbhd } i.$$

$$K \succeq 0$$

$$1^T K 1 = 0$$

- Use semi-definite programing to find K.
- EVD to find the τ_i's.

Laplacian eigenmaps (LE)

- **Belkin et al., 2003**
  - **General idea: minimize the norm of Laplace-Beltrami operator on the manifold**

$$\min \int_{\mathcal{M}} \|\nabla f\|^2 \text{ s.t. } \|f\|^2_{\mathcal{L}(\mathcal{M})} = 1,\ f \perp 1.$$

  - $\int_{\mathcal{M}} \|\nabla f\|^2$ measures how far apart maps nearby points.
  - Avoid the trivial solution of f = const.
  - The Laplacian-Beltrami operator can be approximated by Laplacian of the neighborhood graph with appropriate weights.
  - Construct the Laplacian matrix L=D-W.
  - $\int_{\mathcal{M}} \|\nabla f\|^2$ can be approximated by its discrete equivalent:

$$\sum_{ij} w_{ij} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2.$$

Laplacian Eigenmaps [Belkin& Niyogi'03]

- Construct a neighborhood graph (e.g., epsilon-ball, k-nearest neighbors).
- Construct an adjacency matrix with the following weights $w_{ij} = \exp(-\mathcal{D}_{ij}^2/(2\epsilon))$.
- Minimize $\sum_{ij} w_{ij} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2$.
- The generalized eigendecomposition of the graph Laplacian is $\boldsymbol{L}\boldsymbol{u}_k = \lambda_k \boldsymbol{D}\boldsymbol{u}_k$.
- Spectral embedding of the Laplacian $\mathcal{M} \to \mathbb{R}^d$:
$$\boldsymbol{x}_i \mapsto \boldsymbol{y}_i = [\boldsymbol{u}_2(i), \boldsymbol{u}_3(i), \ldots, \boldsymbol{u}_{d+1}(i)]^T.$$
- The first eigenvector is trivial (the all one vector).

Hessian eigenmaps (HLLE)

- **Dohono et al., 2003**
  - **General idea:** Substitute the Laplace-Beltrami operator with the Hessian of .

$$min_f \int \| H_f(x) \|^2$$

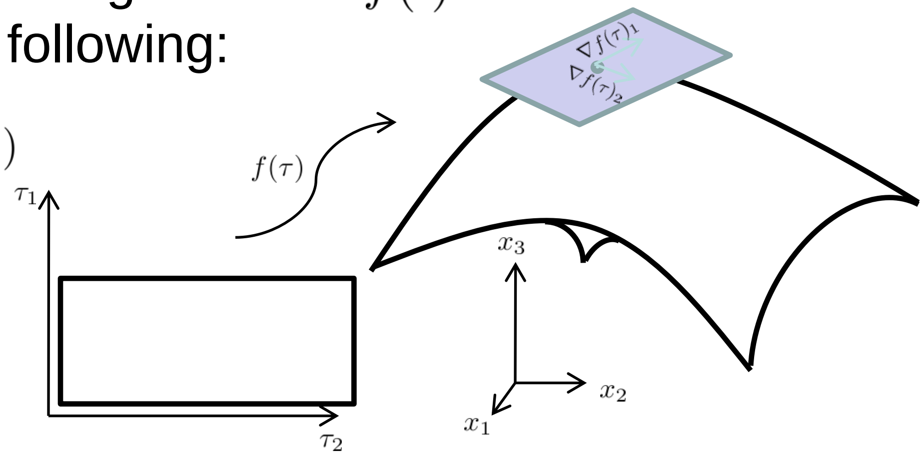
  - The null space of the Hessian matrix is a set of functions with everywhere vanishing Hessian which span the tangent space of the manifold. Therefore, the low dimensional can be recovered from the null space of the Hessian matrix.
  - HLLE is a modification of LE. A function is linear *iff* it has a vanishing Hessian everywhere but it is not true for the Laplacian.

Local tangent space alignment

- Every *smooth manifold* can be constructed locally by its *tangent plane*.
- Taylor series expansion of the embedding function $f(\cdot)$ in the local neighborhood of τ^* can be given as following:

$$f(\tau) = f(\tau^*) + \nabla f(\tau^*)^T(\tau - \tau^*) + O(\| \tau - \tau^* \|^2)$$

$$\xRightarrow[\tau \to \tau^*]{} \quad f(\tau) \approx f(\tau_*) + \nabla f(\tau^*)^T(\tau - \tau^*)$$

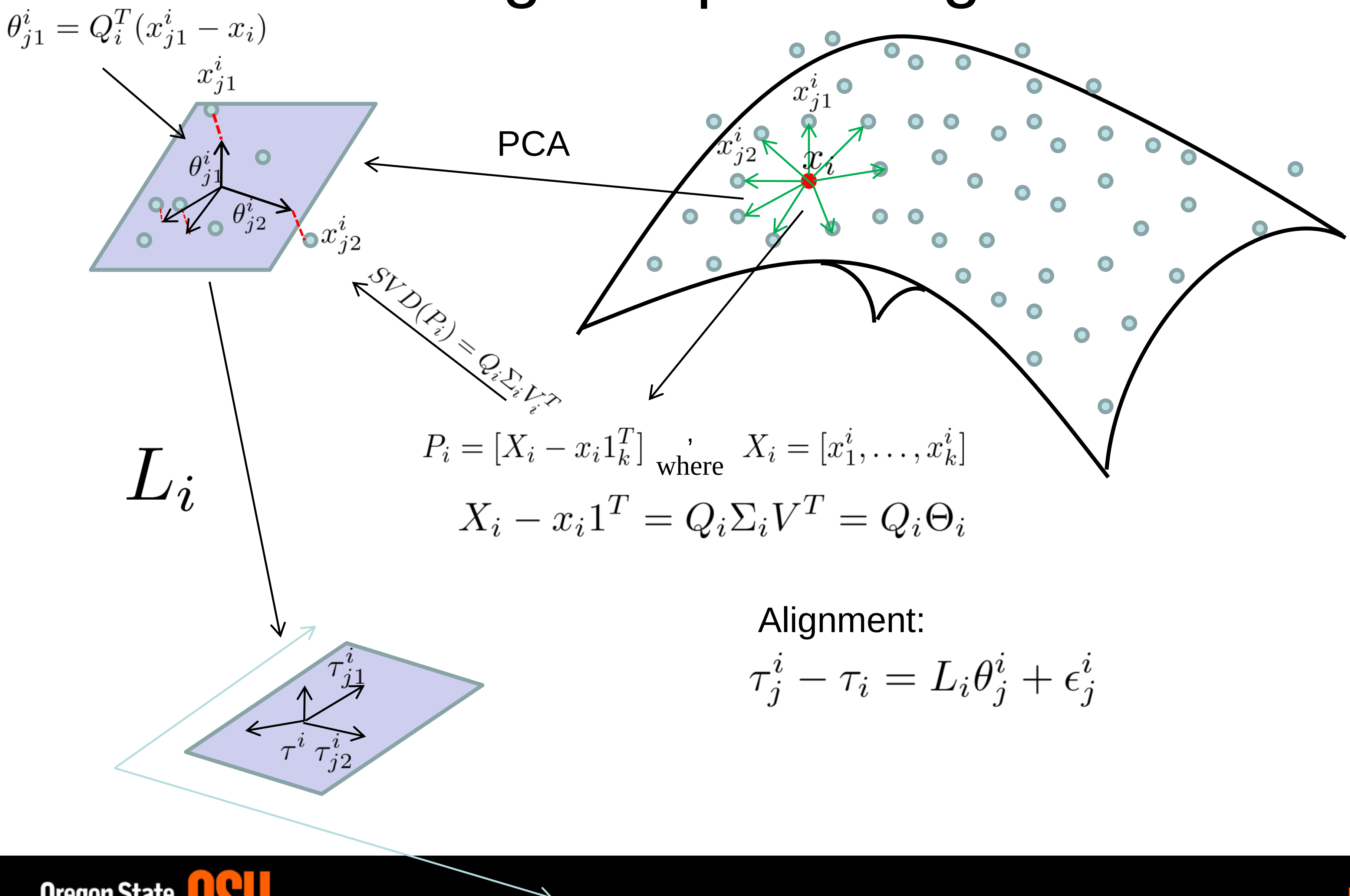


- We are given samples $x_1, \ldots, x_n$ from the embedded manifold with noise therefore, $x_i = f(\tau_i) + \epsilon_i$
- For an arbitrary point x_i and its local neighbor x^i_{j1} and in the absence of the noise, $(\epsilon_i = 0)$ we can write:

$$x_i \approx x^i_{j1} + \nabla f(\tau^i_{j1})^T(\tau_i - \tau^i_{j1}) \quad \Longrightarrow \quad x^i_{j1} - x_i \approx \nabla f(\tau^i_{j1})^T(\tau^i_{j1} - \tau_i)$$

- If we have the *explicit mapping* $f(.)$ therefore we can discover τ_i from the given x_i .

Local tangent space alignment



$P_i = [X_i - x_i 1_k^T]$, where $X_i = [x_1^i, \ldots, x_k^i]$

$$X_i - x_i 1^T = Q_i \Sigma_i V^T = Q_i \Theta_i$$

Alignment:

$$\tau_j^i - \tau_i = L_i \theta_j^i + \epsilon_j^i$$

Local tangent space alignment

- Solve $\min_{\{L_i\},\mathcal{T}} \sum_i \|\mathcal{T} s_i - L_i\theta_i\|^2$
  where s_i is the i-th nbhd-membership vector.
- The optimal alignment (using LS): $L_i = \mathcal{T} s_i \theta_i^\dagger$
- Substituting Li into the objective:
  $$\min_{\mathcal{T}} \|\mathcal{T} S W\|_F^2 \text{ s.t. } \mathcal{T}\mathcal{T}^T = I$$
  where S=[s1,…,sn], W=diag(W_1,…,W_n), and $W_i = (I - 11^T/k)(I - \theta_i\theta_i^\dagger)$
- Solve using an EVD.

Other Nonlinear Methods

- Kohonen Self-Organizing Map [Kohonen1990]
- Kernel PCA [Mika et. Al.’99]
- Neural nets

Probabilistic Approaches

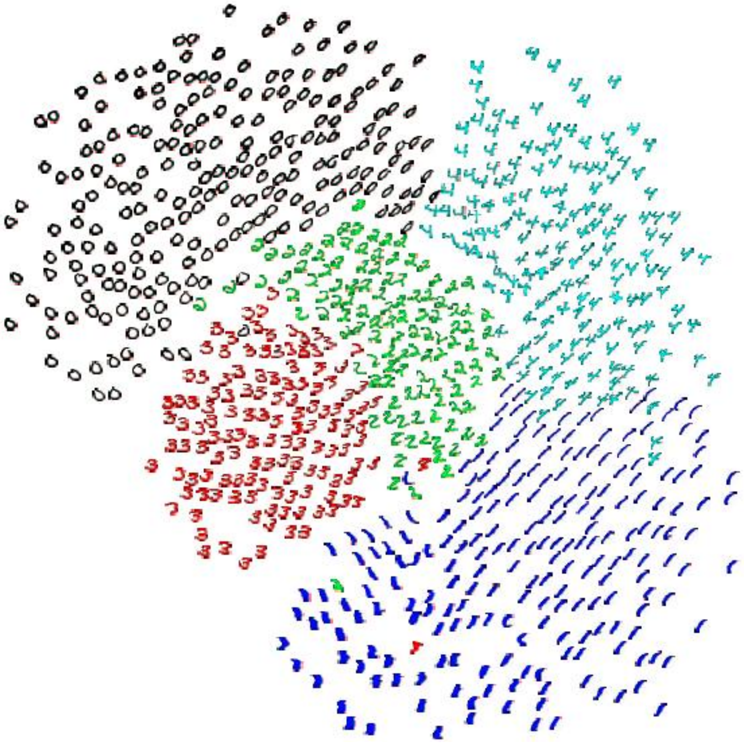
- Based on a probabilistic model relating the high dimensional data and the low dimensional data.
- Examples: SNE, Probabilistic PCA, MFA

Stochastic Neighbor Embedding [Hinton&Roweis'02]

- Construct the probability that will choose j as its neighbor p(j|i):

$$p_{ij} = \frac{\exp(-d_{ij}^2)}{\sum_{k \neq i} \exp(-d_{ik}^2)}$$

$$d_{ij} = \|x_i - x_j\|^2 / (2\sigma_i^2)$$



- For the low-dimensional embedding define:

$$q_{ij} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_i - y_j\|^2)}$$

Stochastic Neighbor Embedding [Hinton&Roweis'02]

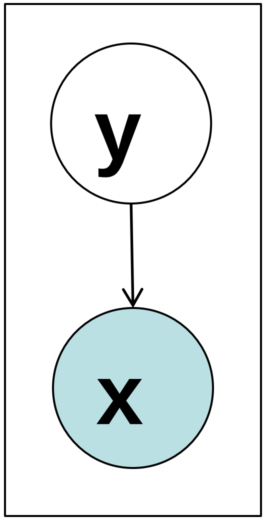
- For each i, find the neighborhood size \sigma_i by $H(p_{i\cdot}) = -\sum_{j\neq i} p_{ij} \log p_{ij} = k$ to produce effective number of neighbors k.
- To find the low dimensional coordinates solve:

$$\min_Y \sum_i KL(p_{i\cdot} \| q_{i\cdot}) = \sum_{i\neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

- Non-convex problem
- Use gradient descent:

$$\nabla_{y_i} = 2\sum_j (y_i - y_j)(p_{ij} - q_{ij} + p_{ji} - q_{ji})$$

Probabilistic PCA [Tipping&Bishop'99]



- Model:
  - Prior: $y \sim \mathcal{N}(0, I)$
  - Conditional: $x|y \sim \mathcal{N}(Wy + \mu, \sigma^2 I)$
  - Marginal: $x \sim \mathcal{N}(\mu, WW^T + \sigma^2 I)$
- Approach: To find the latent low-dimensional embedding y:
  1. Estimate W, μ, and σ^2 using MML.
  2. Estimate y|x using the posterior mean.

Probabilistic PCA [Tipping&Bishop'99]

- Marginal Maximum Likelihood (MML):

$$\min_{\mu,\sigma,W} \log\det C_x + \mathrm{tr}(C_x^{-1} S)$$

$$S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T$$

$$C_x = WW^T + \sigma^2 I$$

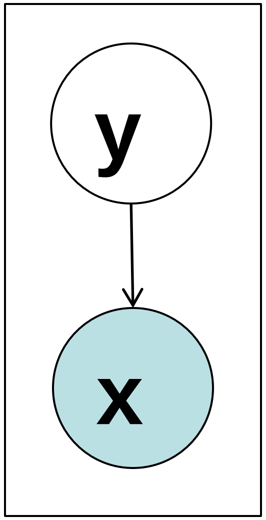
  – Solution in closed-form:

$$\mu_{ML} = \bar{x} \quad S = U\Lambda U^T$$

$$\sigma^2_{ML} = \frac{1}{D-d}\sum_{i=d+1}^{D}\lambda_i$$

$$W = U_d(\Lambda_d - \sigma^2 I)^{1/2}$$

  – Note: as with PCA, PPCA requires the first d eigenvectors of the data covariance matrix.



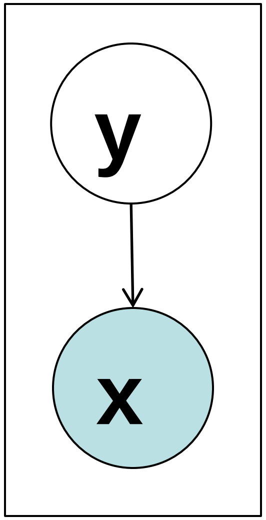
Probabilistic PCA [Tipping&Bishop’99]

- Posterior mean for y|x:

$$E[y_i|x_i] = W^{\dagger}(x_i - \bar{x})$$

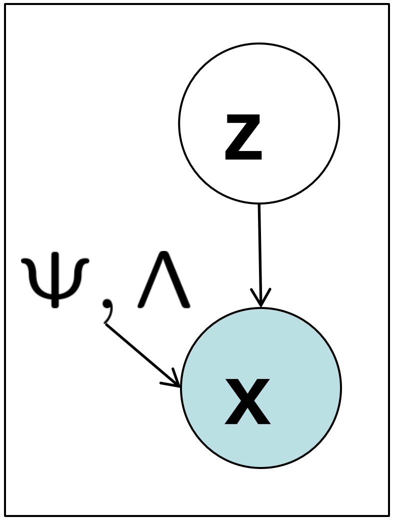
  - Linear Projection.

- Advantages:
  - Natural extension to missing features
  - Natural extension to mixtures of PPCA



Mixture of Factor Analyzers [Ghahramani&Hinton'97]

- Basic factor analyzer model:
  - Prior: $y \sim \mathcal{N}(0, I)$
  - Conditional: $x|z \sim \mathcal{N}(\Lambda z, \Psi)$ diagonal
  - Marginal: $x \sim \mathcal{N}(0, \Lambda\Lambda^T + \Psi)$



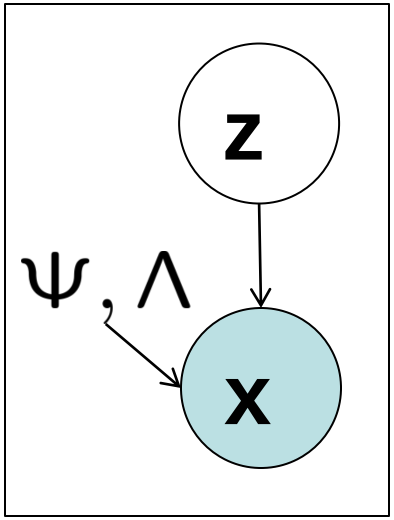
- Approach: To find the latent low-dimensional embedding y:
  1. Estimate Λ and Ψ using MML (EM).
  2. Estimate z|x using the posterior mean.

Mixture of Factor Analyzers [Ghahramani&Hinton'97]

- EM iterations:

$$\Lambda_{new} = \Big(\sum_{i=1}^{n} x_i E[z|x_i]'\Big)\Big(\sum_{i=1}^{n} E[zz|x_i]\Big)^{-1}$$

$$\Psi_{new} = \frac{1}{n}\text{diag}\{\sum_{i=1}^{n} x_i x_i' - \Lambda_{new} E[z|x_i] x_i'\}$$

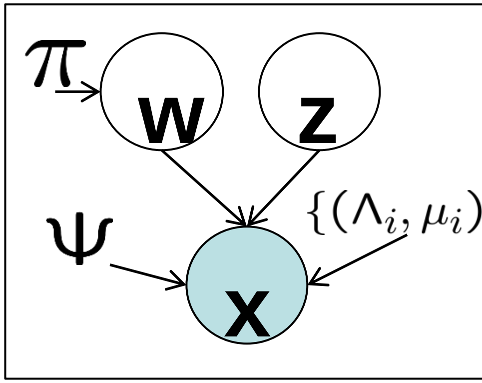


- Posterior mean:

$$E[z|x] = \Lambda'(\Psi + \Lambda\Lambda')^{-1} x$$

Mixture of Factor Analyzers [Ghahramani&Hinton'97]

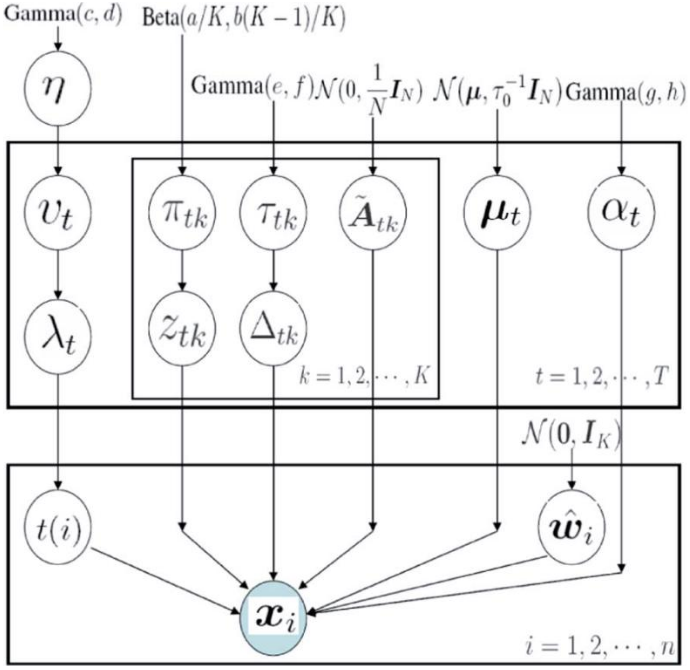
- Mixture of factor analyzers model:
  - Prior: $y \sim \mathcal{N}(0, I),\ w \sim discrete(\pi)$
  - Conditional: $x|z, w \sim \mathcal{N}(\Lambda_w z + \mu_w, \Psi)$
  - Marginal: $x \sim \sum_w \pi_w \mathcal{N}(\mu_w, \Lambda_w \Lambda_w^T + \Psi)$



- Approach: To find the latent low-dimensional embedding y:
  1. Estimate $\{(\Lambda_i, \mu_i)\}$, π, and Ψ using EM.
  2. Estimate z|x,w using the posterior mean.
- Multiple local mappings!

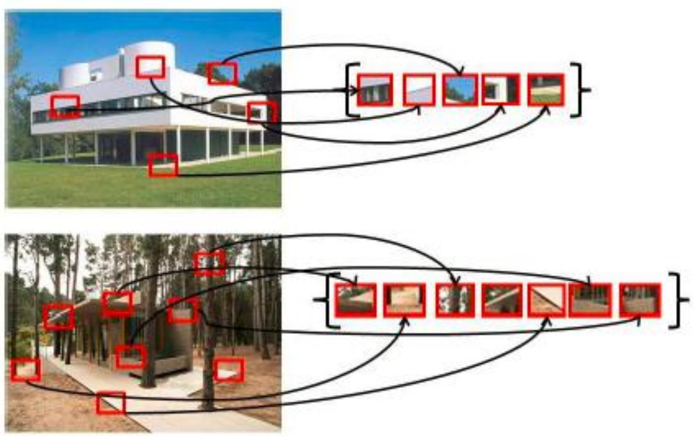
Infinite Mixture of Factor Analyzers [Chen et al'10]

- Uses a non-parametric Bayesian approach – every unknown is a random variable.
- Dirichlet process to facilitate infinite mixture of FAs.
- Use Gibbs sampling to perform inference.

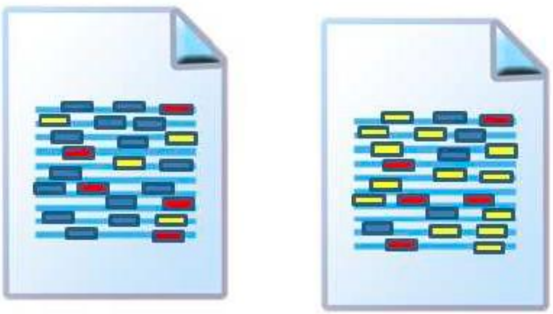


Manifold Learning for Multi-instance Data

- Multiple-instance data



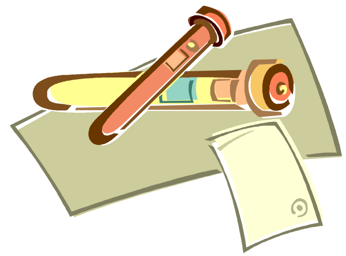
Images

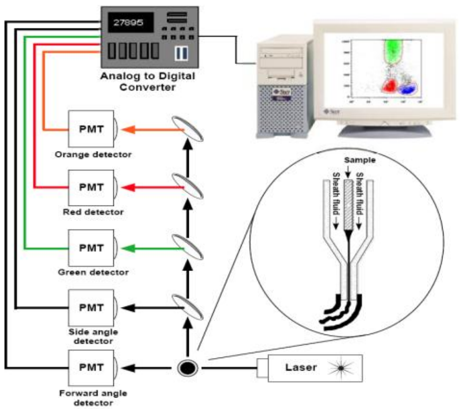


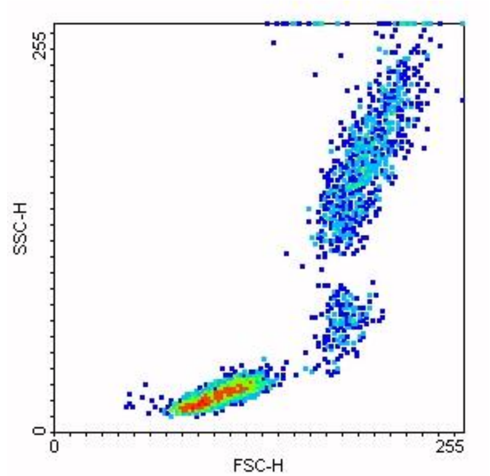
Text documents

- Each example is represented as a collection of feature vectors $X_i=\{x_{1i},x_{2i},\ldots,x_{n_i i}\}$

Application to Flow Cytometry

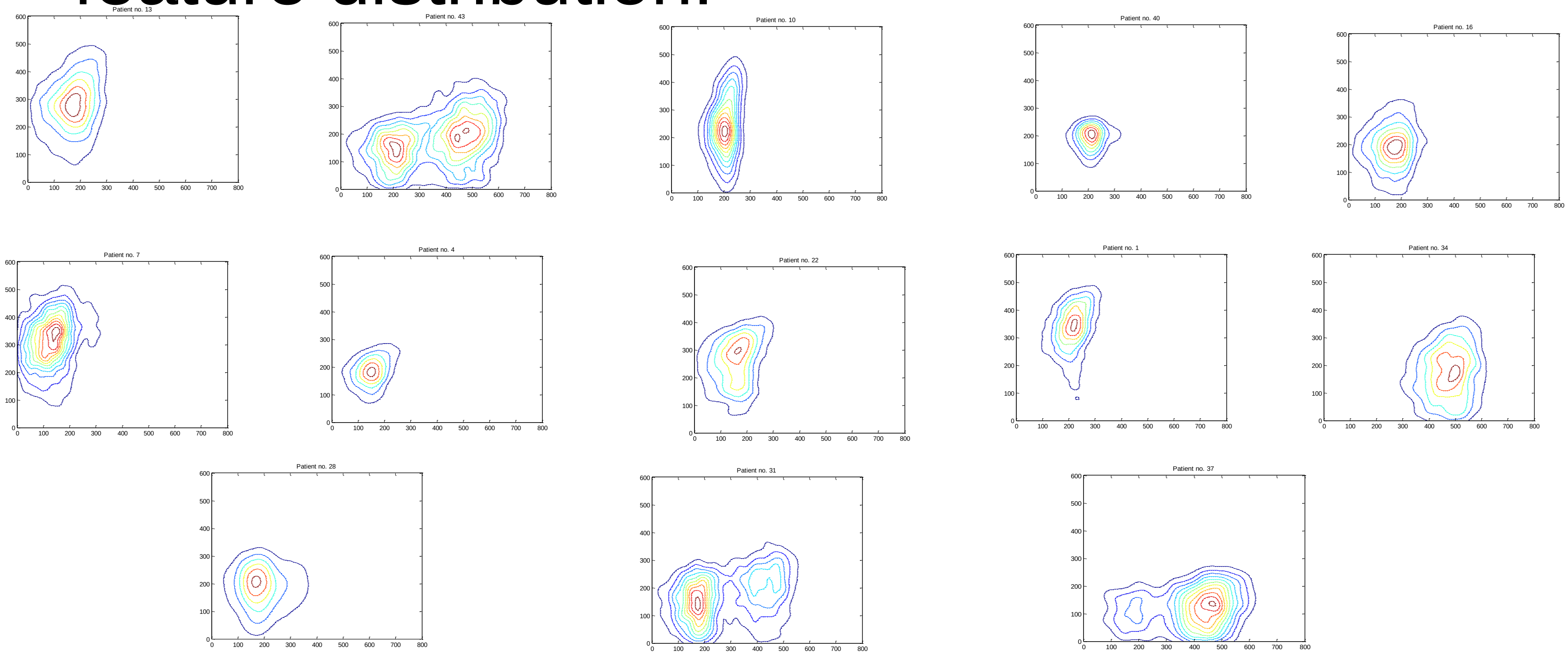






Application to Flow Cytometry

- Each patient is characterize by a cell feature distribution:



Manifold Learning for Multi-instance Data

- How can manifold learning be extended to learning embedding for objects that are not represented as vectors?
- To determine neighborhood graphs, a distance is required. $D(X_i,X_j)=?$
- How can we construct tangent planes?
- Approach: treat the i-th ‘bag’ an iid draw from a generative model $f(x|\theta_i)$

Information Geometry

- Consider the manifold of densities M:

$$\left\{ f(\boldsymbol{y}|\boldsymbol{\theta}) \;\middle|\; \boldsymbol{\theta} \in \mathbb{R}^m, \;\; \int f(\boldsymbol{y}|\boldsymbol{\theta}) = 1, \; f(\boldsymbol{y}|\boldsymbol{\theta}) \geq 0 \right\}.$$

- Use the Fisher information metric as a Riemannian metric for the manifold:

$$\mathcal{I}_{ij} = \int \frac{\partial \log f(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_i} \frac{\partial \log f(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_j} f(\boldsymbol{y}|\boldsymbol{\theta}) d\boldsymbol{y}.$$

The metric defines an inner product, which allows us to compute distances.

Information Geometry

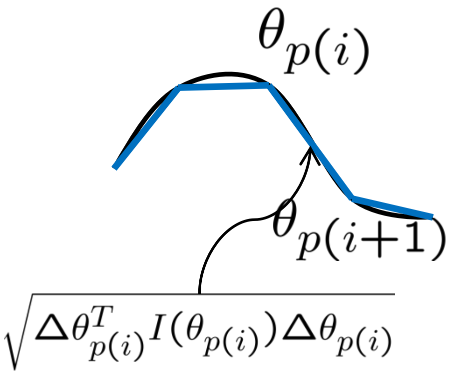
- Geodesic distance:

$$D_F(\theta_1, \theta_2) = \min_{\substack{\theta(\cdot): \\ \theta(0)=\theta_1 \\ \theta(1)=\theta_2}} \int_0^1 \sqrt{\left(\frac{d\theta}{d\beta}\right)^T \mathcal{I}(\theta) \left(\frac{d\theta}{d\beta}\right)} d\beta$$

- Let p(i), i=1,2,…,P denote vertex squence.
- Path graph approximation :

$$l \approx \sum_i \sqrt{\Delta\theta_{p(i)}^T I(\theta_{p(i)}) \Delta\theta_{p(i)}}$$

$$\Delta\theta_{p(i)} = (\theta_{p(i+1)} - \theta_{p(i)})$$



- How to approximate the FIM?

Information Geometry

- Approximating the FIM:
  - Using **KL divergence**
    $$\log f(x|\theta+d\theta)-\log f(x|\theta) \approx \frac{d\log f(x|\theta)}{d\theta}d\theta+\frac{1}{2}d\theta^T\frac{d^2\log f(x|\theta)}{d\theta d\theta^T}d\theta$$
    Integrate both sides $-\int \cdot f(x|\theta)dx$ $\Longrightarrow$ $\boxed{D_{KL}(\theta||\theta+d\theta) \approx \frac{1}{2}d\theta^T I(\theta)d\theta}$
  - Can also show: $D_{KL}(\theta+d\theta||\theta) \approx \frac{1}{2}d\theta^T I(\theta)d\theta$
  - Symmetrized KL: $D^s_{KL}(\theta+d\theta,\theta) = D_{KL}(\theta+d\theta||\theta)+D_{KL}(\theta||\theta+d\theta) \approx d\theta^T I(\theta)d\theta$
  - Using **Hellinger distance**:
    $$\sqrt{f(x|\theta+d\theta)}-\sqrt{f(x|\theta)} \approx \frac{d\sqrt{f}(x|\theta)}{d\theta}d\theta = \sqrt{f}(x|\theta)\frac{d\log f(x|\theta)}{d\theta}d\theta$$
    square and integrate both sides $\int \cdot dx$:
    $$\boxed{D^2_H(\theta+d\theta||\theta) \approx d\theta^T I(\theta)d\theta}$$
- $$\boxed{\Delta\theta^T_{p(i)}I(\theta_{p(i)})\Delta\theta_{p(i)} \approx D^s_{KL}(\theta_{p(i+1)}||\theta_{p(i)}) \approx D^2_H(\theta_{p(i+1)}||\theta_{p(i)})}$$

Information Geometry

- Approximation to the length of a path:

  $$l \approx \sum_i D_H(\theta_{p(i+1)} || \theta_{p(i)})$$

  or

  $$l \approx \sum_i \sqrt{D^s_{KL}(\theta_{p(i+1)} || \theta_{p(i)})}$$

- The *Hellinger distance* plays the same role as the *Euclidean distance* in manifolds that are based on a Euclidean metric.
- Similar approximations can be obtained to tangent vectors and tangent planes using the Taylor series expansion.

Experimental Setting

- Unsupervised learning – clustering.
- 43 Patients: 23 CLL patients and 20 MCL patients.
- For both diseases, analysis is of just the lymphocytes.
- Varying number of cells (around 5000-6000) per patient are recorded.
- Testing ten different six-dimensional marker combination data samples.

Experimental Setting

- Use kernel density estimation:

$$\widehat{f}_i(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{N_i} K_h(\boldsymbol{x}, \boldsymbol{x}_i).$$

  with a Gaussian kernel.
- Use the Kullback-Leibler divergence

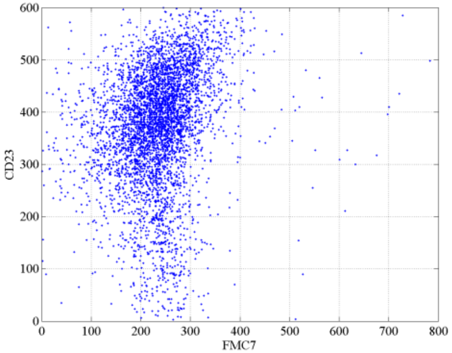
$$D_{KL}(f_i \| f_j) = \int \log\left(\frac{f_i(\boldsymbol{x})}{f_j(\boldsymbol{x})}\right) f_i(\boldsymbol{x}) d\boldsymbol{x}$$

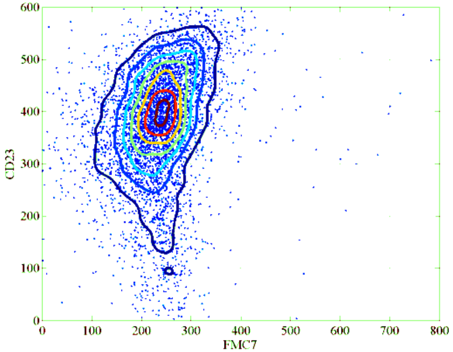
  to form the distance matrix:

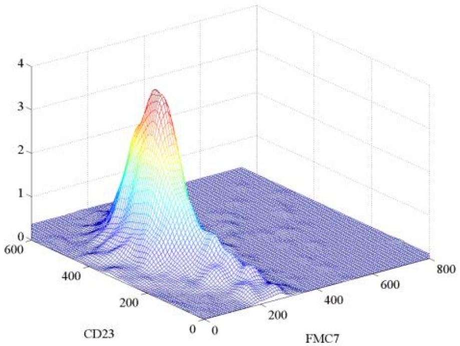
$$\mathcal{D}_{ij} = D_{KL}(f_i \| f_j) + D_{KL}(f_j \| f_i).$$

- Use multidimensional scaling (MDS) to find a two-dimensional embedding.

Obtaining the PDFs



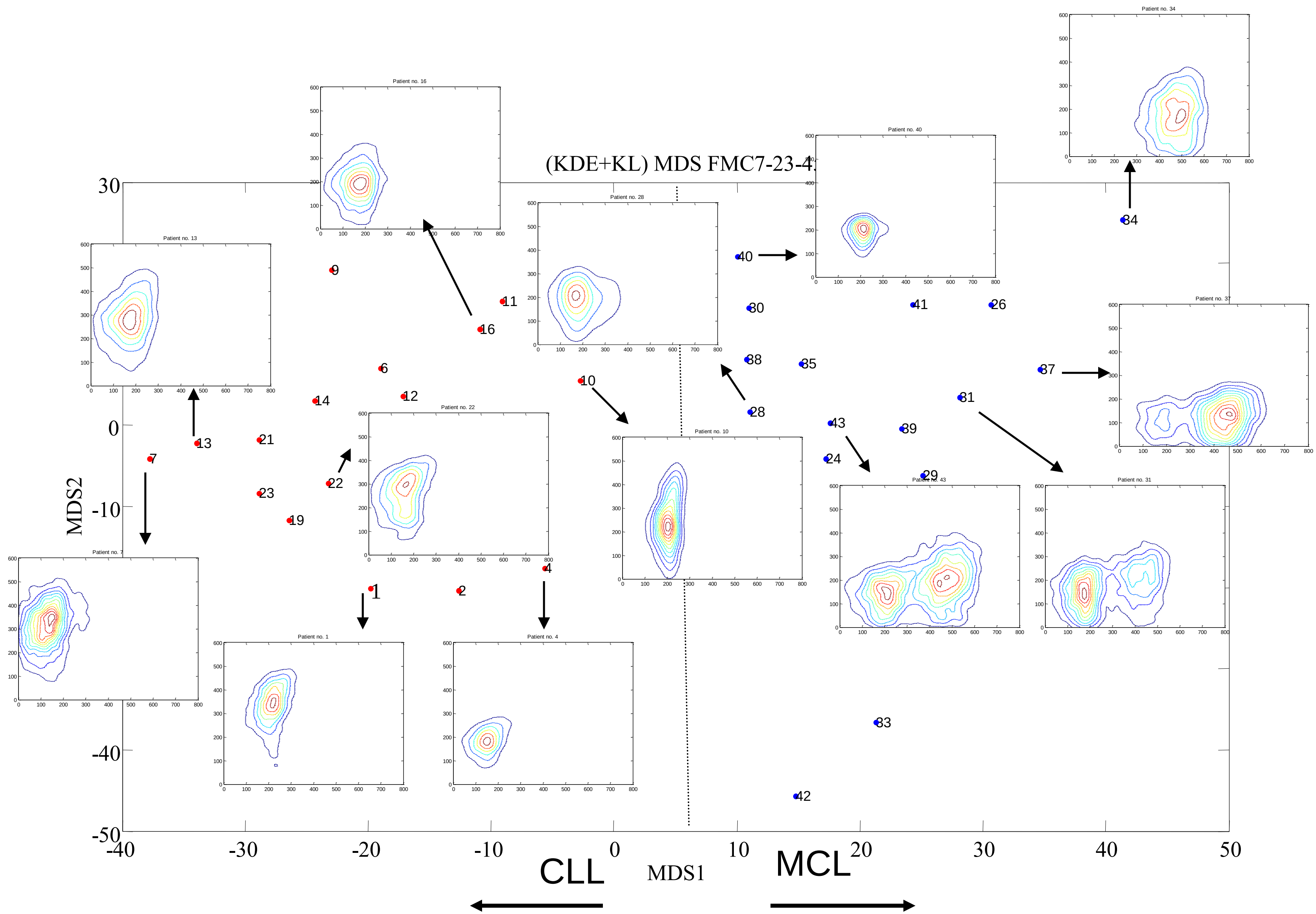




Patient 1: Scatter plot of FMC7 vs. CD23

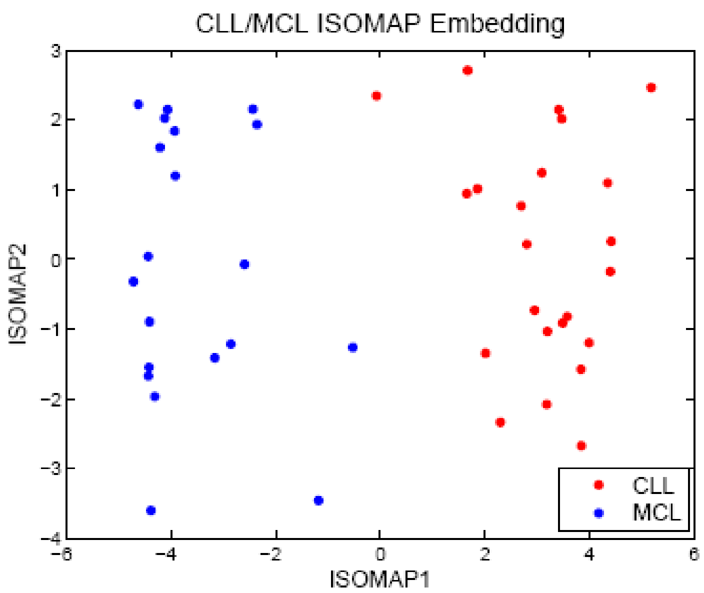
Patient 1: density estimate

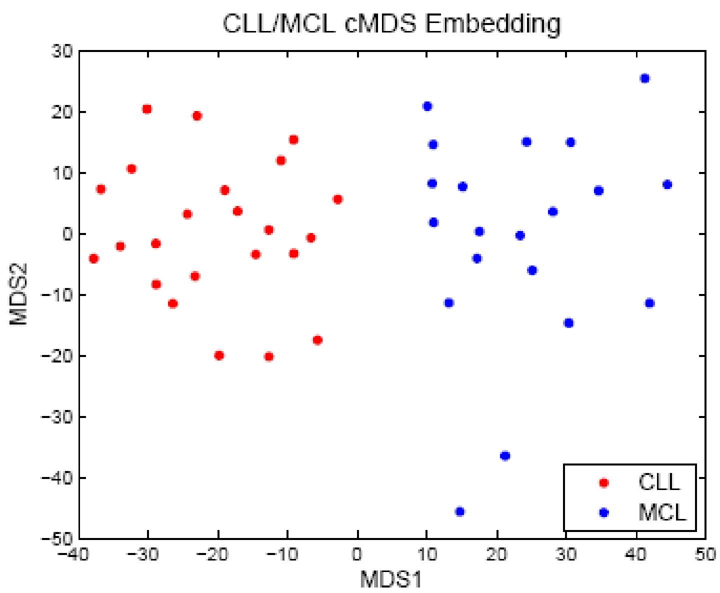
• Actual density estimation was performed for the six-variate density.



(KDE+KL) MDS FMC7-23-4
MDS2
MDS1
CLL
MCL
Patient no. 16
Patient no. 13
Patient no. 28
Patient no. 40
Patient no. 34
Patient no. 37
Patient no. 22
Patient no. 10
Patient no. 43
Patient no. 31
Patient no. 7
Patient no. 1
Patient no. 4

Different Embedding Methods





- ISOMAP seems to provide a greater separation than the classical MDS.
- Using the geodesics on the manifold (instead of direct distances) improved performance.

Maximum Entropy Manifolds [Behmardi et al'12]

- Parametric approach: use maximum entropy to describe each bag-of-instances as a PDF:

$$f(x|\theta) = \exp(\phi(x)^T \theta - Z(\theta))$$

- ML estimation

$$\widehat{\theta}_{ML} = \arg\min_\theta Z(\theta) - \overline{\phi(x)}^T \theta$$

  – Convex
  – Simple sufficient statistics for each bag: $\overline{\phi(x)}$

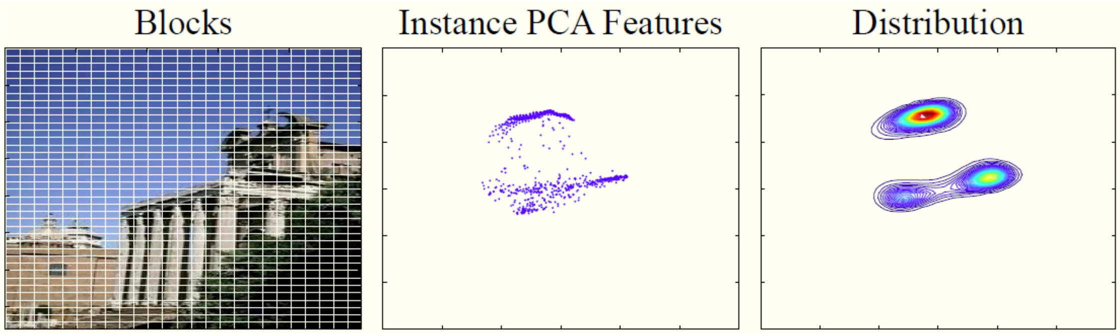
- KL-divergence:

$$D_{KL}(\theta_1 \| \theta_2) = \dot{Z}(\theta_1)(\theta_1 - \theta_2) - (Z(\theta_1) - Z(\theta_2))$$

$$D_{KL}^S(\theta_1, \theta_2) = (\dot{Z}(\theta_1) - \dot{Z}(\theta_2))(\theta_1 - \theta_2)$$

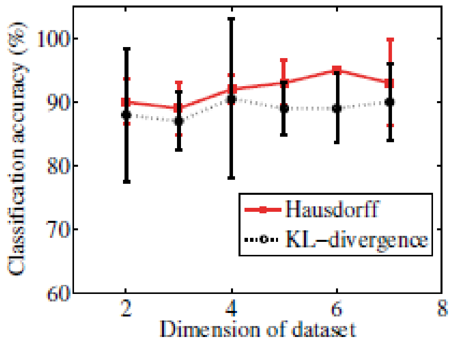
Maximum Entropy Manifolds [Behmardi et al'12]

- Experiment:
- Corel 1000 data set
- Each image is divided in to blocks
- Use PCA to represent each block using a low-dimensional vector.

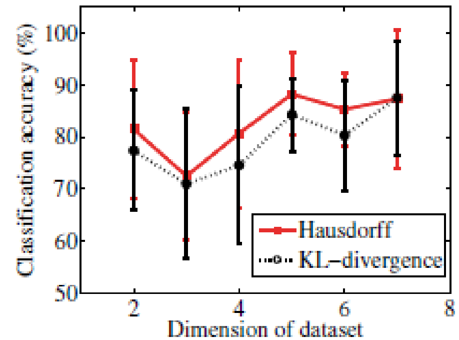


Maximum Entropy Manifolds [Behmardi et al'12]

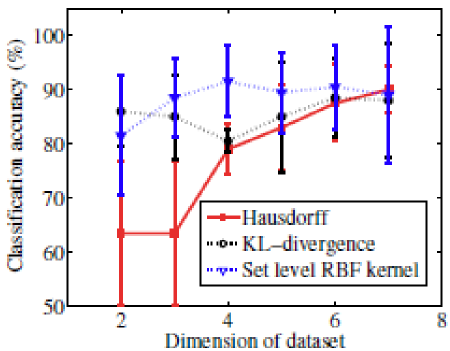
- Accuracy:



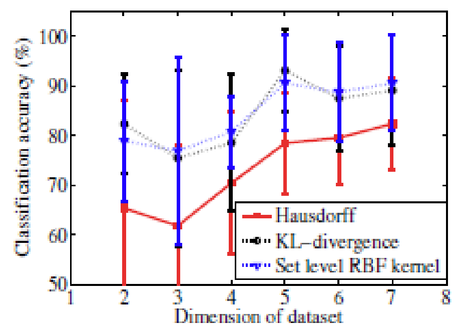
(a) Citation-kNN, Corel1000



(b) Citation-kNN, Musk2



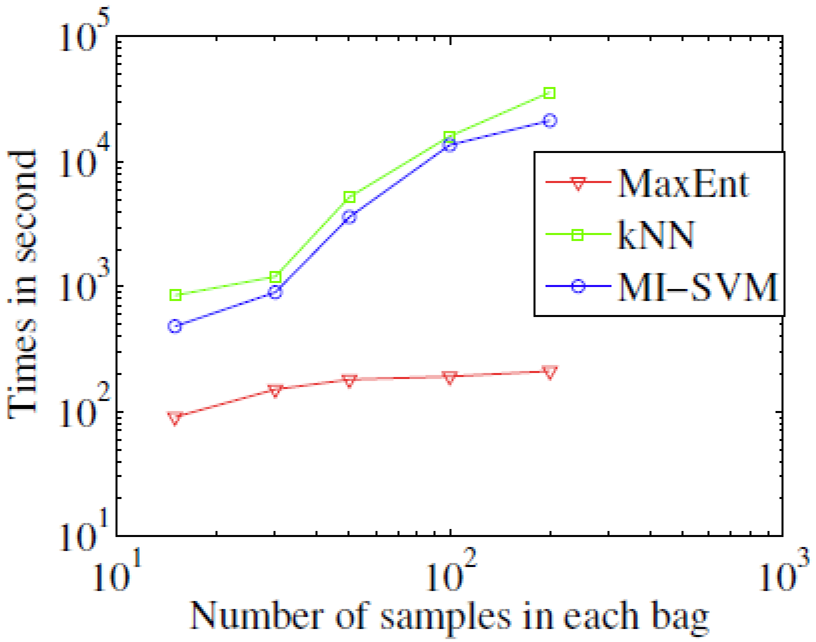
(c) SVM, Corel1000



(d) SVM, Musk2

Maximum Entropy Manifolds [Behmardi et al'12]

- Runtime:



Conclusion

- Introduced linear and nonlinear dimension reduction
- Presented manifold and manifold learning techniques
- Common tools
- Geometric vs. probabilistic
- Generalization to probability spaces

List of references for manifold learning

1 Algorithms

1.1 Graph-based approach

1.1.1 Globally Embedding

1. A global geometric framework for nonlinear dimensionality reduction (ISOMAP) [1]
2. Maximum variance unfolding (MVU) [2, 3]
3. Diffusion maps [4]
4. Graph approximations to geodesics on embedded manifolds [5]
5. Unsupervised learning of curved manifolds[6]

1.1.2 Locally Embedding

1. Locally Linear Embedding (LLE) [7, 8]
2. Laplacian eigenmaps [9]
3. Hessian eigenmpas [10]
4. Local tangent space alignment (LTSA) [11]
5. Manifold charting [12]
6. Two-Manifold Problems with Applications to Nonlinear System Identification [13]
7. Robust Multiple Manifolds Structure Learning [14]

1.1.3 Variations of global and local embedding

1. Conformal Isomap Embedding [15]
2. Graph laplacian regularization for large-scale semidefinite programming [16]
3. Modified locally linear embedding [17]
4. Colored maximum variance unfolding [18]
5. Grouping and dimensionality reduction by locally linear embedding [19]
6. Sparse multidimensional scaling using landmark points [20]
7. Improved local coordinate coding using local tangents [21]

1.2 Probabilistic approach

1. Mixture of factor analysis (MFA) [22]
2. Stochastic neighborhood embedding (SNE) [23]
3. The generative topographic mapping (GTM) [24]
4. Probabilistic principal component analysis [25]
5. Global coordinate of local linear models [26]
6. Automatic alignment of local representations [27]
7. Coordinating principal component analysis [28]

1.3 Non-probabilistic approach

1. Multilayer autoencoders [29]
2. The self-organizing map [30]
3. Sammon mapping [31]
4. Kernel PCA [32]
5. Principal curves [33]
6. A variational approach to recovering a manifold from sample points [34]

7. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering [35]
8. Continuous nonlinear dimensionality reduction by kernel eigenmaps [36]
9. Learning eigenfunctions links spectral embedding and kernel PCA [37]
10. Sparse manifold clustering and embedding [38]

1.4 Supervised and semisupervised manifold learning

1. Vector-valued manifold regularization [39]
2. Multiple instance learning with manifold bags [40]
3. The manifold tangent classifier [41]

2 Applications

1. Maximum covariance unfolding: Manifold learning for bimodal data [42]
2. Humans learn using manifolds, reluctantly [43]
3. Learning multiple tasks using manifold regularization [44]
4. Online learning in the manifold of low-rank matrices [45]
5. Manifold Precis: An Annealing Technique for Diverse Sampling of Manifolds [46]
6. Nonlinear dimensionality reduction as information retrieval [47]
7. Information retrieval perspective to nonlinear dimensionality reduction for data visualization [48]
8. Unified Locally Linear Embedding and Linear Discriminant Analysis Algorithm (ULLELDA) for Face Recognition [49]
9. Generative modeling for continuous non-linearly embedded visual inference [50]
10. Manifold learning and applications in recognition [51]
11. Graph-driven features extraction from microarray data using diffusion kernels and kernel CCA [52]
12. Manifold based analysis of facial expression [53]
13. A dimensionality reduction approach to modeling protein flexibility [54]

14. Face recognition from face motion manifolds using robust kernel resistor-average distance [55]
15. Coloring of DT-MRI fiber traces using Laplacian eigenmaps [56]
16. Freeway traffic stream modeling based on principal curves and its analysis [57]
17. Super-resolution through neighbor embedding [58]

3 Dimension Estimation

1. Manifold-adaptive dimension estimation [59]
2. Towards manifold-adaptive learning [60]
3. Maximum likelihood estimation of intrinsic dimension [61]
4. Manifold learning using Euclidean k-nearest neighbor graphs [62]
5. An intrinsic dimensionality estimator from near-neighbor information [63]
6. Intrinsic dimension estimation of manifolds by incising balls [64]
7. Intrinsic dimension estimation by maximum likelihood in probabilistic PCA [65]

References

[1] J.B. Tenenbaum, V. De Silva, and J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[2] K.Q. Weinberger and L.K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. IEEE, 2004, vol. 2, pp. II–988.

[3] K.Q. Weinberger, F. Sha, and L.K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 106.

[4] R.R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[5] M. Bernstein, V. De Silva, J.C. Langford, and J.B. Tenenbaum, "Graph approximations to geodesics on embedded manifolds," Tech. Rep., Technical report, Department of Psychology, Stanford University, 2000.

[6] V. de Silva and J. Tenenbaum, "Unsupervised learning of curved manifolds," in *Proceedings of the MSRI workshop on nonlinear estimation and classification*, 2002.

[7] S.T. Roweis and L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[8] L.K. Saul and S.T. Roweis, "An introduction to locally linear embedding," *unpublished. Available at: http://www. cs. toronto. edu/˜ roweis/lle/publications. html*, 2000.

[9] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *Advances in neural information processing systems*, vol. 14, pp. 585–591, 2001.

[10] D.L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 10, pp. 5591, 2003.

[11] T. Zhang, J. Yang, D. Zhao, and X. Ge, "Linear local tangent space alignment and application to face recognition," *Neurocomputing*, vol. 70, no. 7, pp. 1547–1553, 2007.

[12] M. Brand, "Charting a manifold," *Advances in neural information processing systems*, pp. 985–992, 2003.

[13] B. Boots and G. Gordon, "Two-manifold problems with applications to nonlinear system identification," *ICML2012*, 2012.

[14] D. Gong, X. Zhao, and G. Medioni, "Robust multiple manifolds structure learning," *ICML2012*, 2012.

[15] V. Silva and J.B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," *Advances in neural information processing systems*, vol. 15, pp. 705–712, 2003.

[16] K.Q. Weinberger, F. Sha, Q. Zhu, and L.K. Saul, "Graph laplacian regularization for large-scale semidefinite programming," *Advances in neural information processing systems*, vol. 19, pp. 1489, 2007.

[17] Z. Zhang and J. Wang, "Mlle: Modified locally linear embedding using multiple weights," *Advances in Neural Information Processing Systems*, vol. 19, pp. 1593, 2007.

[18] L. Song, A. Smola, K. Borgwardt, and A. Gretton, "Colored maximum variance unfolding," *Advances in neural information processing systems*, vol. 20, pp. 1385–1392, 2008.

[19] M. Polito and P. Perona, "Grouping and dimensionality reduction by locally linear embedding," *Advances in Neural Information Processing Systems*, vol. 14, pp. 1255–1262, 2001.

[20] V. De Silva and J.B. Tenenbaum, "Sparse multidimensional scaling using landmark points," *Technology*, pp. 1–41, 2004.

[21] K. Yu and T. Zhang, "Improved local coordinate coding using local tangents," in *Proc. of the Intl Conf. on Machine Learning (ICML)*, 2010.

[22] Z. Ghahramani and G.E. Hinton, "The em algorithm for mixtures of factor analyzers," Tech. Rep., Technical Report CRG-TR-96-1, University of Toronto, 1996.

[23] G. Hinton and S. Roweis, "Stochastic neighbor embedding," *Advances in neural information processing systems*, vol. 15, pp. 833–840, 2002.

[24] C.M. Bishop, M. Svensén, and C.K.I. Williams, "Gtm: The generative topographic mapping," *Neural computation*, vol. 10, no. 1, pp. 215–234, 1998.

[25] M.E. Tipping and C.M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[26] ST Roweis, L.K. Saul, and G.E. Hinton, "Global coordination of local linear models," *Advances in neural information processing systems*, vol. 2, pp. 889–896, 2002.

[27] Y.W. Teh and S. Roweis, "Automatic alignment of local representations," *Advances in neural information processing systems*, vol. 15, pp. 841–848, 2002.

[28] J. Verbeek, N. Vlassis, and B. Kröse, "Coordinating principal component analyzers," *Artificial Neural NetworksICANN 2002*, pp. 140–140, 2002.

[29] D. DeMers and G. Cottrell, "Non-linear dimensionality reduction," *Advances in neural information processing systems*, pp. 580–580, 1993.

[30] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[31] J.W. Sammon Jr, "A nonlinear mapping for data structure analysis," *Computers, IEEE Transactions on*, vol. 100, no. 5, pp. 401–409, 1969.

[32] B. Schölkopf, A. Smola, and K.R. Müller, "Kernel principal component analysis," *Artificial Neural NetworksICANN'97*, pp. 583–588, 1997.

[33] T. Hastie and W. Stuetzle, "Principal curves," *Journal of the American Statistical Association*, pp. 502–516, 1989.

[34] J. Gomes and A. Mojsilovic, "A variational approach to recovering a manifold from sample points," *Computer VisionECCV 2002*, pp. 3–17, 2002.

[35] Y. Bengio, J.F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet, "Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering," *Advances in neural information processing systems*, vol. 16, pp. 177–184, 2004.

[36] M. Brand, "Continuous nonlinear dimensionality reduction by kernel eigenmaps," in *International Joint Conference on Artificial Intelligence*. LAWRENCE ERLBAUM ASSOCIATES LTD, 2003, vol. 18, pp. 547–554.

[37] Y. Bengio, O. Delalleau, N.L. Roux, J.F. Paiement, P. Vincent, and M. Ouimet, "Learning eigenfunctions links spectral embedding and kernel pca," *Neural Computation*, vol. 16, no. 10, pp. 2197–2219, 2004.

[38] E. Elhamifar and R. Vidal, "Sparse manifold clustering and embedding," *Advances in Neural Information Processing Systems*, vol. 24, pp. 55–63, 2011.

[39] H.Q. Minh and V. Sindhwani, "Vector-valued manifold regularization," *ICML2011*, 2011.

[40] N. Dollar P. Babenko, B. Verma and S. Belongie, "Multiple instance learning with manifold bags," *ICML2011*, 2011.

[41] S. Rifai, Y. Dauphin, P. Vincent, Y. Bengio, and X. Muller, "The manifold tangent classifier," *Advances in Neural Information Processing Systems*, 2011.

[42] V. Mahadevan, C.W. Wong, J.C. Pereira, T.T. Liu, N. Vasconcelos, and L.K. Saul, "Maximum covariance unfolding: Manifold learning for bimodal data," *Advances in Neural Information Processing Systems*, vol. 24, 2011.

[43] B. Gibson, X. Zhu, T. Rogers, C. Kalish, and J. Harrison, "Humans learn using manifolds, reluctantly," *Advances in neural information processing systems*, vol. 24, 2010.

[44] A. Agarwal, H. Daumé III, and S. Gerber, "Learning multiple tasks using manifold regularization," *Advances in neural information processing systems*, vol. 23, pp. 46–54, 2010.

[45] U. Shalit, D. Weinshall, and G. Chechik, "Online learning in the manifold of low-rank matrices," *Advances in Neural Information Processing Systems*, vol. 23, pp. 2128–2136, 2010.

[46] N. Shroff, P. Turaga, and R. Chellappa, "Manifold précis: An annealing technique for diverse sampling of manifolds," *Advances in Neural Information Processing Systems*, 2011.

[47] J. Venna and S. Kaski, "Nonlinear dimensionality reduction as information retrieval," *AISTAT*, 2007.

[48] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski, "Information retrieval perspective to nonlinear dimensionality reduction for data visualization," *The Journal of Machine Learning Research*, vol. 11, pp. 451–490, 2010.

[49] J. Zhang, H. Shen, and Z.H. Zhou, "Unified locally linear embedding and linear discriminant analysis algorithm (ullelda) for face recognition," *Advances in Biometric Person Authentication*, pp. 1–16, 2005.

[50] C. Sminchisescu and A. Jepson, "Generative modeling for continuous non-linearly embedded visual inference," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 96.

[51] J. Zhang, S. Li, and J. Wang, "Manifold learning and applications in recognition," *Intelligent Multimedia Processing with Soft Computing*, pp. 281–300, 2005.

[52] J.P. Vert and M. Kanehisa, "Graph-driven features extraction from microarray data using diffusion kernels and kernel cca," *Advances in Neural Information Processing Systems*, vol. 15, pp. 1425–1432, 2002.

[53] Y. Chang, C. Hu, R. Feris, and M. Turk, "Manifold based analysis of facial expression," *Image and Vision Computing*, vol. 24, no. 6, pp. 605–614, 2006.

[54] M.L. Teodoro, G.N. Phillips Jr, and L.E. Kavraki, "A dimensionality reduction approach to modeling protein flexibility," in *Proceedings of the sixth annual international conference on Computational biology*. ACM, 2002, pp. 299–308.

[55] O. Arandjelovic and R. Cipolla, "Face recognition from face motion manifolds using robust kernel resistor-average distance," in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*. IEEE, 2004, pp. 88–88.

[56] A. Brun, H.J. Park, H. Knutsson, and C.F. Westin, "Coloring of dt-mri fiber traces using laplacian eigenmaps," *Computer Aided Systems Theory-EUROCAST 2003*, pp. 518–529, 2003.

[57] D. Chen, J. Zhang, S. Tang, and J. Wang, "Freeway traffic stream modeling based on principal curves and its analysis," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 5, no. 4, pp. 246–258, 2004.

[58] H. Chang, D.Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. IEEE, 2004, vol. 1, pp. I–275.

[59] A. massoud Farahmand, C. Szepesvári, and J.Y. Audibert, "Manifold-adaptive dimension estimation," in *Proceedings of the 24th international conference on Machine learning*. Citeseer, 2007, pp. 265–272.

[60] A. Farahmand, C. Szepesvári, and J. Audibert, "Towards manifold-adaptive learning," 2007.

[61] E. Levina and P.J. Bickel, "Maximum likelihood estimation of intrinsic dimension," *Ann Arbor MI*, vol. 48109, pp. 1092, 2004.

[62] J.A. Costa and A.O. Hero III, "Manifold learning using euclidean k-nearest neighbor graphs [image processing examples]," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*. IEEE, 2004, vol. 3, pp. iii–988.

[63] K.W. Pettis, T.A. Bailey, A.K. Jain, and R.C. Dubes, "An intrinsic dimensionality estimator from near-neighbor information," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, , no. 1, pp. 25–37, 1979.

[64] M. Fan, H. Qiao, and B. Zhang, "Intrinsic dimension estimation of manifolds by incising balls," *Pattern Recognition*, vol. 42, no. 5, pp. 780–787, 2009.

[65] C. Bouveyron, G. Celeux, S. Girard, et al., "Intrinsic dimension estimation by maximum likelihood in probabilistic pca," in *73rd Annual Meeting of the Institute of Mathematical Statistics, Gothenburg, Sweden*, 2010.