Visualization
(Nonlinear dimensionality reduction)

Lecture 23: April 16

Based on Fei Sha's presentation at
Radlab Machine learning short course (8/24/2007)

Administrative

- Quiz 3 is April 27
- This one will have a Scantron portion
- Know your USC ID #
- Bring a #2 Pencil

Dimensionality reduction

- **Question:**

How can we detect low dimensional structure in high dimensional data?

- **Motivations:**

Exploratory data analysis & visualization

Compact representation

Robust statistical modeling

Linear dimensionality reductions

- **Many examples**

Principal component analysis (PCA)

Linear discriminant analysis (LDA)

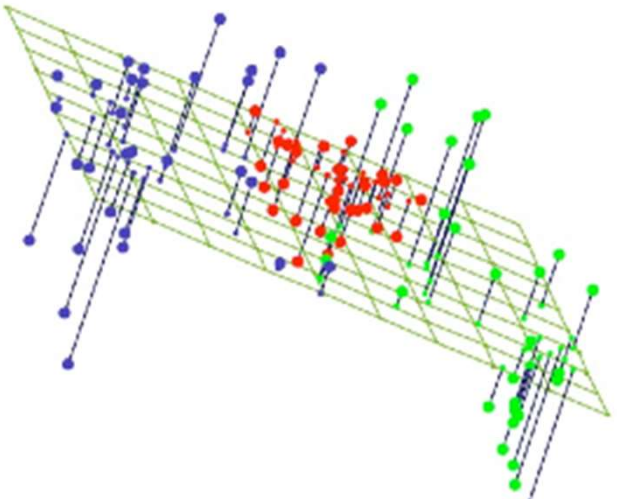
Non-negative matrix factorization (NMF)

- **Framework**

$$\boldsymbol{x} \in \Re^D \rightarrow \boldsymbol{y} \in \Re^d$$

$$D \gg d$$

$$\boldsymbol{y} = \boldsymbol{U}\boldsymbol{x}$$



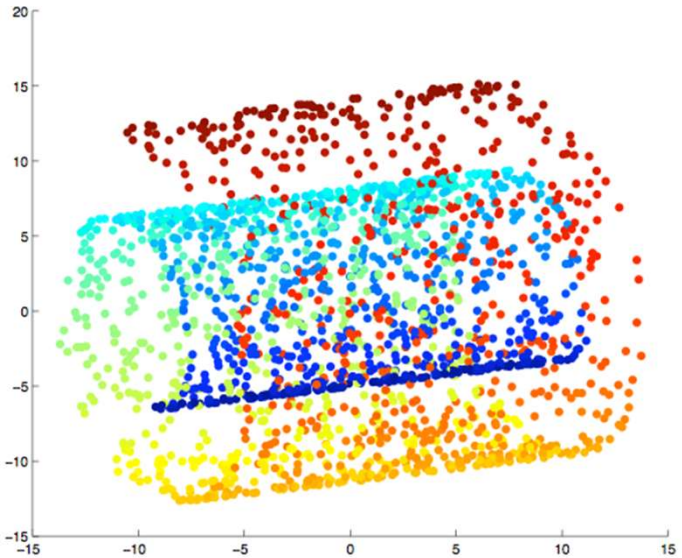
linear transformation of original space

Linear methods are not sufficient

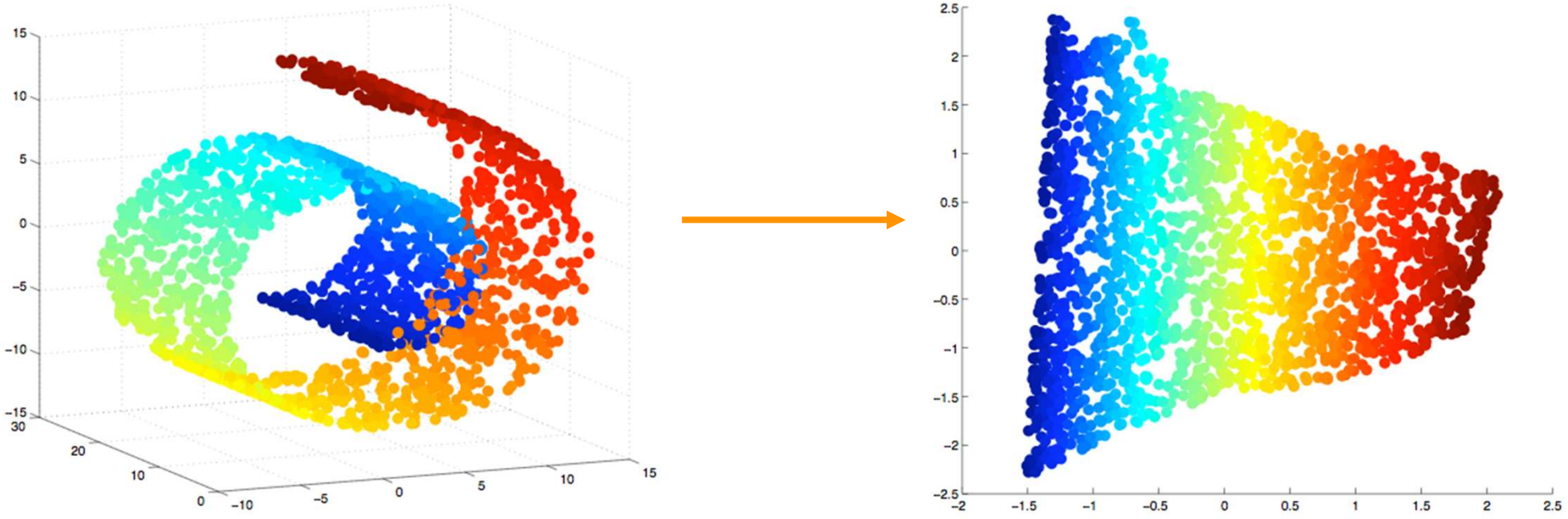
- **What if data is "nonlinear"?**

  classic toy example of Swiss roll

- **PCA results**



What we really want is “unrolling”



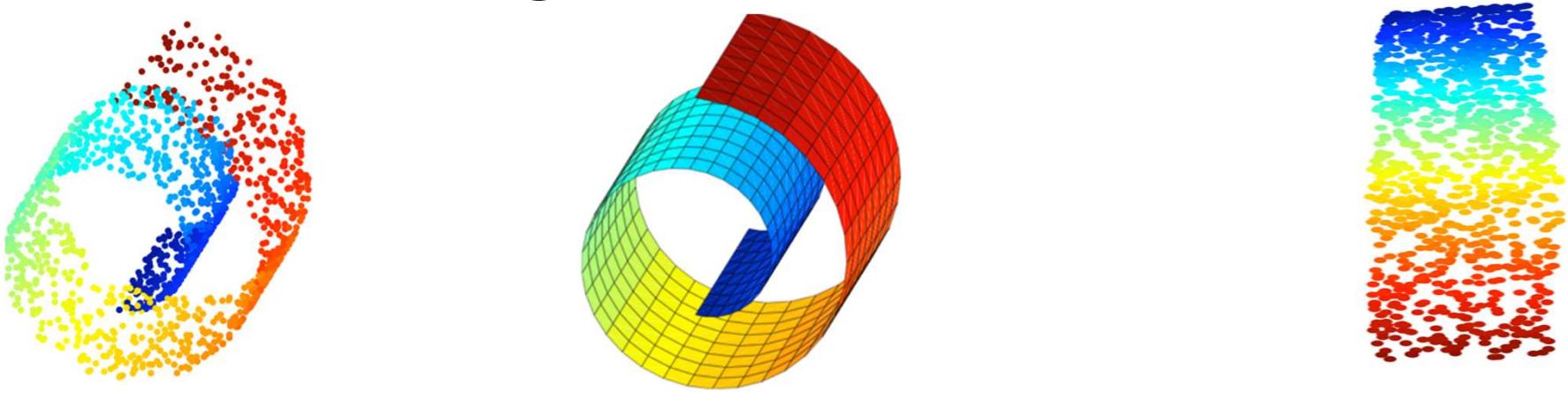
Simple geometric intuition:

distortion in local areas

faithful in global structure

Manifold learning: focus of this lecture

Given high dimensional data sampled from a low dimensional nonlinear submanifold, how to compute a faithful embedding?



Input

$$\{\boldsymbol{x}_i \in \Re^D, i = 1, 2, \ldots, n\}$$

Output

$$\{\boldsymbol{y}_i \in \Re^d, i = 1, 2, \ldots, n\}$$

Flash card for some concepts

- **Manifold: locally looks like a Euclidean space**

Trivial example: Euclidean space

Slightly more interesting: sphere, torus, etc

- **Submanifold: an object that it is self a manifold**

Example: equator of a sphere

- **Riemannian manifold: a manifold that is a metric space or has inner product defined in its tangent bundle**

Outline

- Linear method: redux and new intuition
- Graph based spectral methods

  Isomap

  Locally linear embedding
- Kernel methods

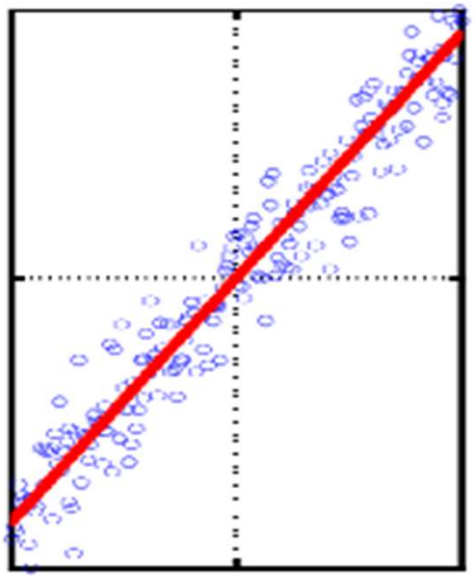
  Kernel PCA

  Kernel CCA
- Case studies

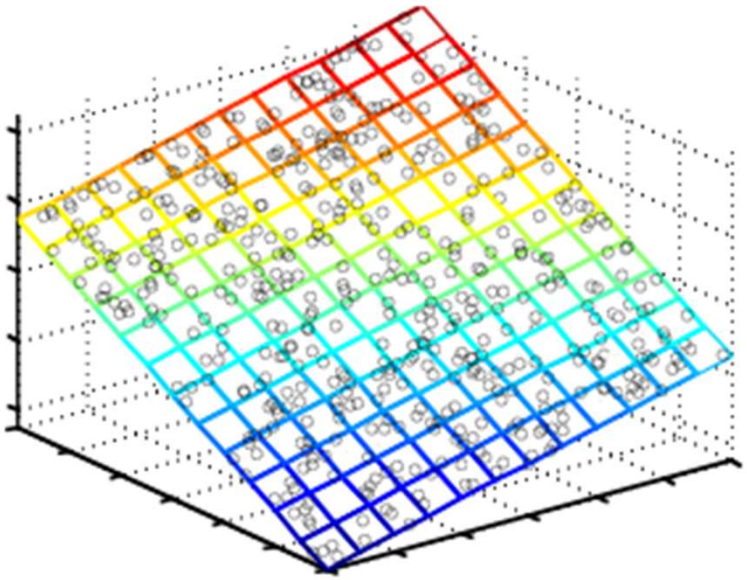
Linear methods: redux

PCA: does the data mostly lie in a subspace? If so, what is its dimensionality?

D = 2
d = 1



D = 3
d = 2



The framework of PCA

- **Assumption:**

Centered inputs

$$\sum_i \boldsymbol{x}_i = \mathbf{0}$$

Projection into subspace

$$\boldsymbol{y}_i = \boldsymbol{U}\boldsymbol{x}_i$$

$$\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{I}$$

- **Interpretation**

maximum variance preservation

$$\arg\max \sum_i \|\boldsymbol{y}_i\|^2$$

minimum reconstruction errors

$$\arg\min \sum_i \|\boldsymbol{x}_i - \boldsymbol{U}^{\mathrm{T}}\boldsymbol{y}_i\|^2$$

Other criteria we can think of...

How about preserve pairwise distances?

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\| = \|\boldsymbol{y}_i - \boldsymbol{y}_j\|$$

equivalently, preserve inner product

$$\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j = \boldsymbol{y}_i^{\mathrm{T}} \boldsymbol{y}_j$$

This leads to a new linear method ....

Multidimensional scaling (MDS)

- Compute Gram matrix

$$\boldsymbol{G} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$$

$$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$$

- Diagonalize

$$\boldsymbol{G} = \sum_i \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^{\mathrm{T}} \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$$

- Derive outputs and estimate dimensionality

$$d = \min \arg\max \mathbb{1}\left(\sum_{i=1}^{d} \lambda_i \geq \mathrm{THRESHOLD}\right)$$

$$y_{id} = \sqrt{\lambda_i} v_{id}$$

PCA vs MDS: is MDS really that new?

- Same set of eigenvalues

$$\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{v} = \lambda\boldsymbol{v} \rightarrow \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\frac{1}{N}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{v} = N\lambda\frac{1}{N}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{v}$$

PCA diagonalization MDS diagonalization

- Same low dimensional representation
- Different computational cost

PCA scales quadratically in D

MDS scales quadratically in N

Big win for MDS is D is much greater than N !

How to generalize to nonlinear manifolds?

All we need is a simple twist on MDS

Outline

- Linear method: redux and new intuition
- **Graph based spectral methods**

Isomap

Locally linear embedding

- Kernel methods

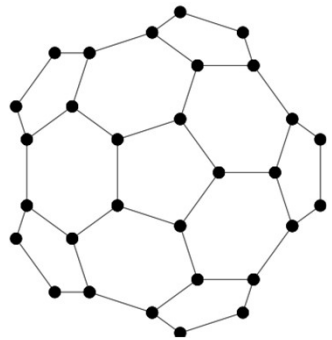
Kernel PCA

Kernel CCA

- Case studies

Graph based spectral methods: a receipe

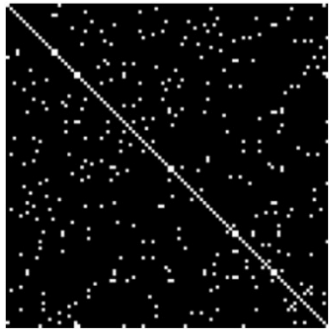
- **Construct nearest neighbor graph**



Vertices are data points

Edges indicate nearest neighbors

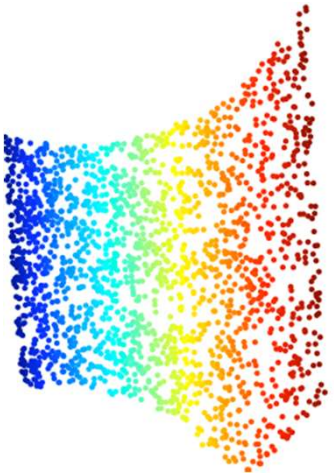
- **Spectral decomposition**



Formulate matrix from the graph

Diagonalize the matrix

- **Derive embedding**



Eigenvector as embedding

Estimate dimensionality

A small jump from MDS to Isomap

- **Key idea**

Preserve pairwise distances

- **Algorithm in a nutshell**

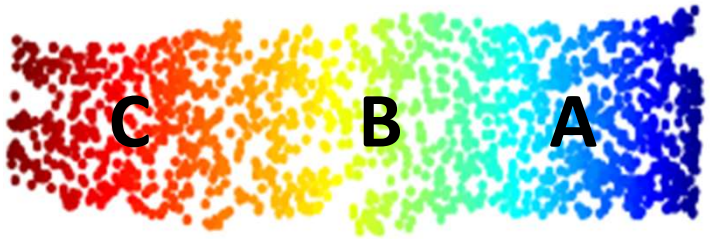
Estimate geodesic distance along submanifold

Perform MDS as if the distances are Euclidean

Why geodesic distances?

Euclidean distance is not appropriate measure of proximity between points on nonlinear manifold.

A closer to C in Euclidean distance



A closer to B in geodesic distance

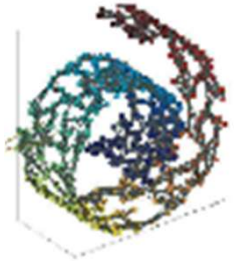
Step 1. Build adjacency graph

- **Graph from nearest neighbor**

Vertices represent inputs

Edges connect nearest neighbors

- **How to choose nearest neighbor**



k-nearest neighbors

Epsilon-radius ball

Q: Why nearest neighbors?

A1: local information more reliable than global information

A2: geodesic distance ≈ Euclidean distance

Building the graph

- **Computation cost**

kNN scales naively as $O(N^2D)$

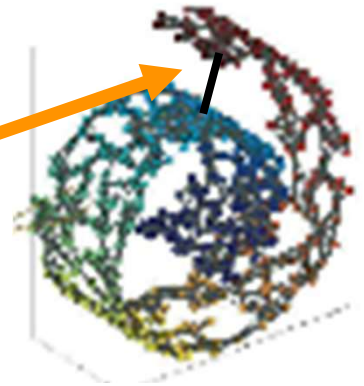
Faster methods exploit data structure (eg, KD-tree)

- **Assumptions**

Graph is connected (if not, run algorithms on each connected component)

No short-circuit

Large k would cause this problem



Step 2. Construct geodesic distance matrix

- **Dynamic programming**

Weight edges by local Euclidean distance

Compute all-pair shortest paths on the graph

- **Geodesic distances**

Approximate geodesic by shortest paths

Require dense sampling

- **Computational cost**

Djikstra's algorithm: $O(N^2 \log N + N^2 k)$

Very intensive for large graph

Step 3. Metric MDS

- **Convert geodesic matrix to Gram matrix**

Pretend the geodesic matrix is from Euclidean distance matrix

- **Diagonalize the Gram matrix**

Gram matrix is a dense matrix, ie, no sparsity

Can be intensive if the graph is big.

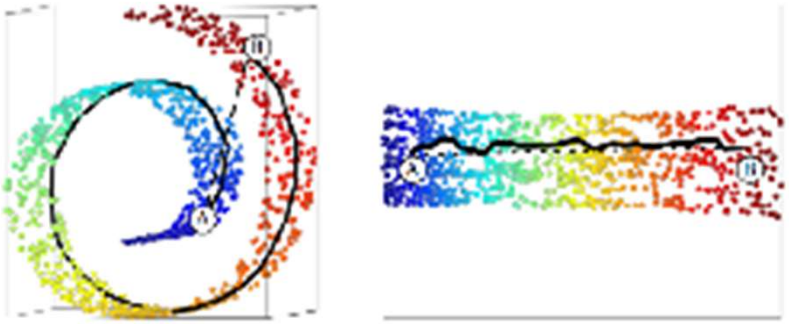
- **Embedding**

 # significant eigenvalues :estimate of dimensionality

Top eigenvectors yield embedding.

Examples

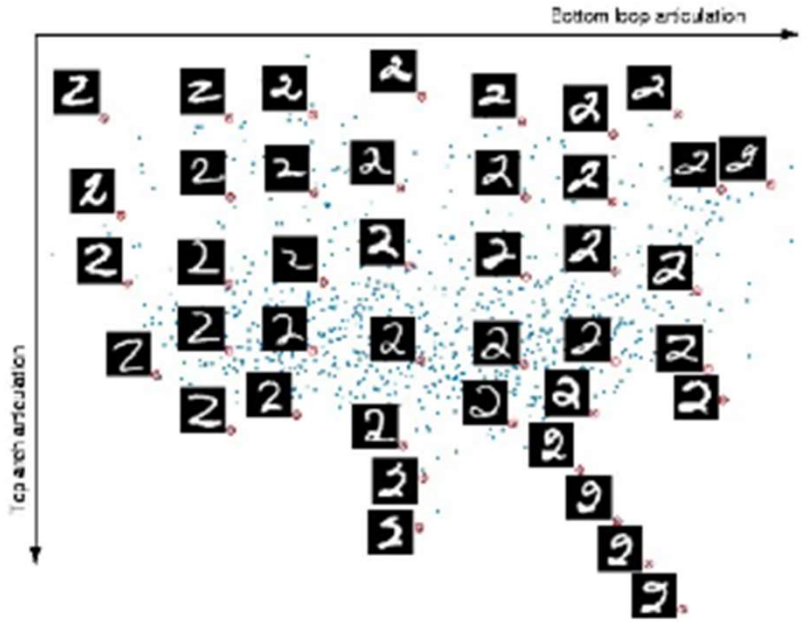
- **Swiss roll**



N = 1024
k = 12

- **Digit images**

N = 1000
r = 4.2
D = 400



Properties of Isomap

- **Strengths**

Simple: kNN, shortest path, diagonalization

Polynomial time complexity

No local optimum, no iterative procedure

One free parameter: neighborhood size

- **Weakness**

Sensitive to short-circuits

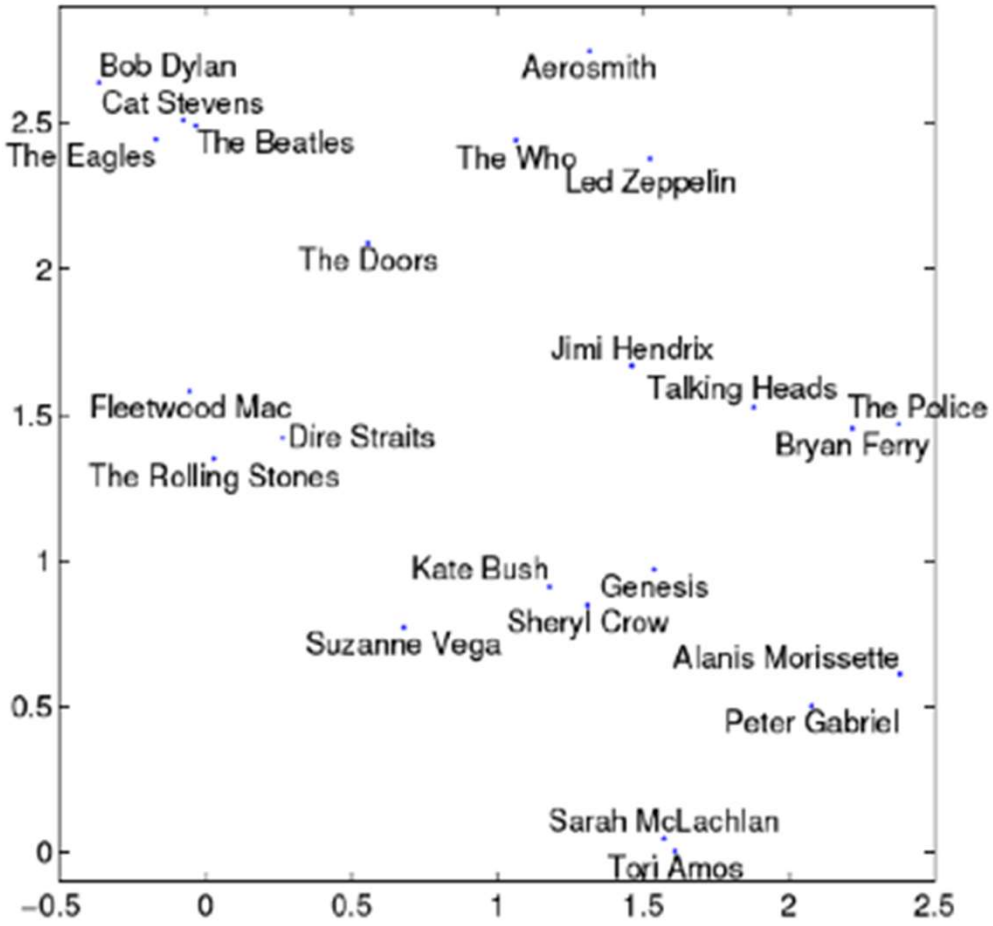
Computation can be intensive

No out-of-sample extension: what if we need to incorporate a new data point?

Applications: Isomap for music

Embedding of sparse music similarity graph (Platt, NIPS 2004)

N = 267,000
E = 3.22 million



Algorithm #2: Maximum Variance Unfolding

- Goal: faithfully preserve distances & angles on nearby input patterns.

- Algorithm:
  1. Compute *k*-nearest neighbors.
  2. “Unfold” by maximizing variance on outputs.

- This can be done via semi-definite programming.

As a semi-definite program

- Maximize trace(K) subject to:
  - $K \geq 0$
  - $\Sigma_{ij} K_{ij} = 0$
  - $K_{ii} - 2K_{ij} + K_{jj} = ||x_i - x_j||^2$ all neighbors

Comparison to Isomap

- Similarities:
  - Both use isometry
  - Both form a Gram Matrix, use its eigenvalues
- Differenes:
  - MVU attempts to “pull apart” (subject to distance constraints)

Algorithm #3: locally linear embedding (LLE)

- **Intuition**

Better off being myopic and trusting only local information

- **Steps**

Define locality by nearest neighbors

Encode local information

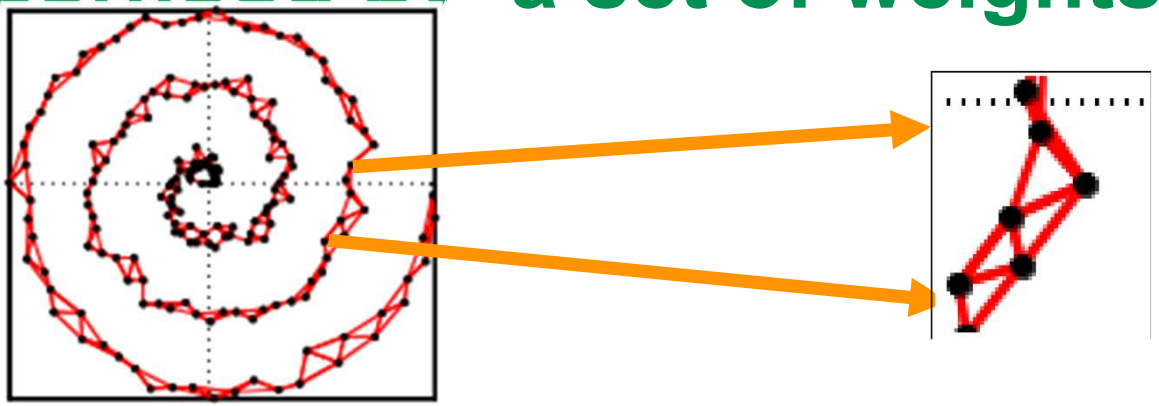
Least square fit locally

Minimize global objective to preserve local information

Think globally

Step 2. Least square fits

- Characterize local geometry of each neighborhood by a set of weights

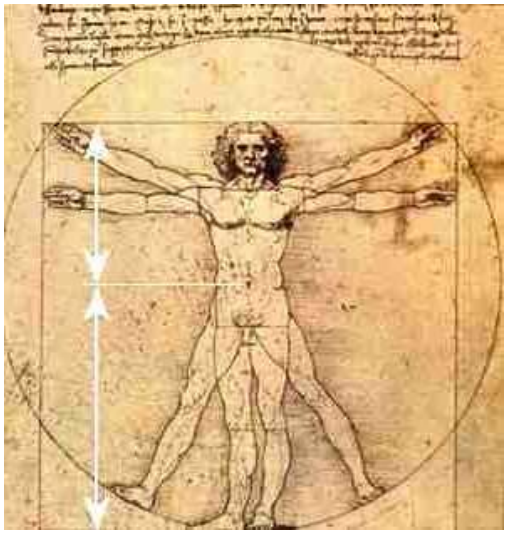


- Compute weights by reconstructing each input linearly from its neighbors

$$\Phi(\boldsymbol{W}) = \sum_i \|\boldsymbol{x}_i - \sum_k \boldsymbol{W}_{ik}\boldsymbol{x}_k\|^2$$

Symmetries encoded by weights

The head should sit in the middle of left and right finger tips.



Step 3. Preserve local information

- The embedding should follow same local encoding

$$\boldsymbol{y}_i \approx \sum_k \boldsymbol{W}_{ik} \boldsymbol{y}_k$$

- Minimize a global reconstruction error

$$\Psi(\boldsymbol{Y}) = \sum_i \| \boldsymbol{y}_i - \sum_k \boldsymbol{W}_{ik} \boldsymbol{y}_k \|^2$$

Sparse eigenvalue problem

- **Quadratic form**

$$\arg\min \Psi(\boldsymbol{Y}) = \sum_{ij} \Psi_{ij} \boldsymbol{y}_i^{\mathrm{T}} \boldsymbol{y}_j$$

$$\boldsymbol{\Psi} = (\boldsymbol{I} - \boldsymbol{W})^{\mathrm{T}} (\boldsymbol{I} - \boldsymbol{W})$$

- **Rayleigh-Ritz quotient**

Embedding given by bottom eigenvectors

Discard bottom eigenvector [1 1 ... 1]

Other d eigenvectors yield embedding

Summary of LLE

- **Three steps**

Compute nearest neighbors

Compute local weights

Compute embedding

- **Optimizations**

Least square fits

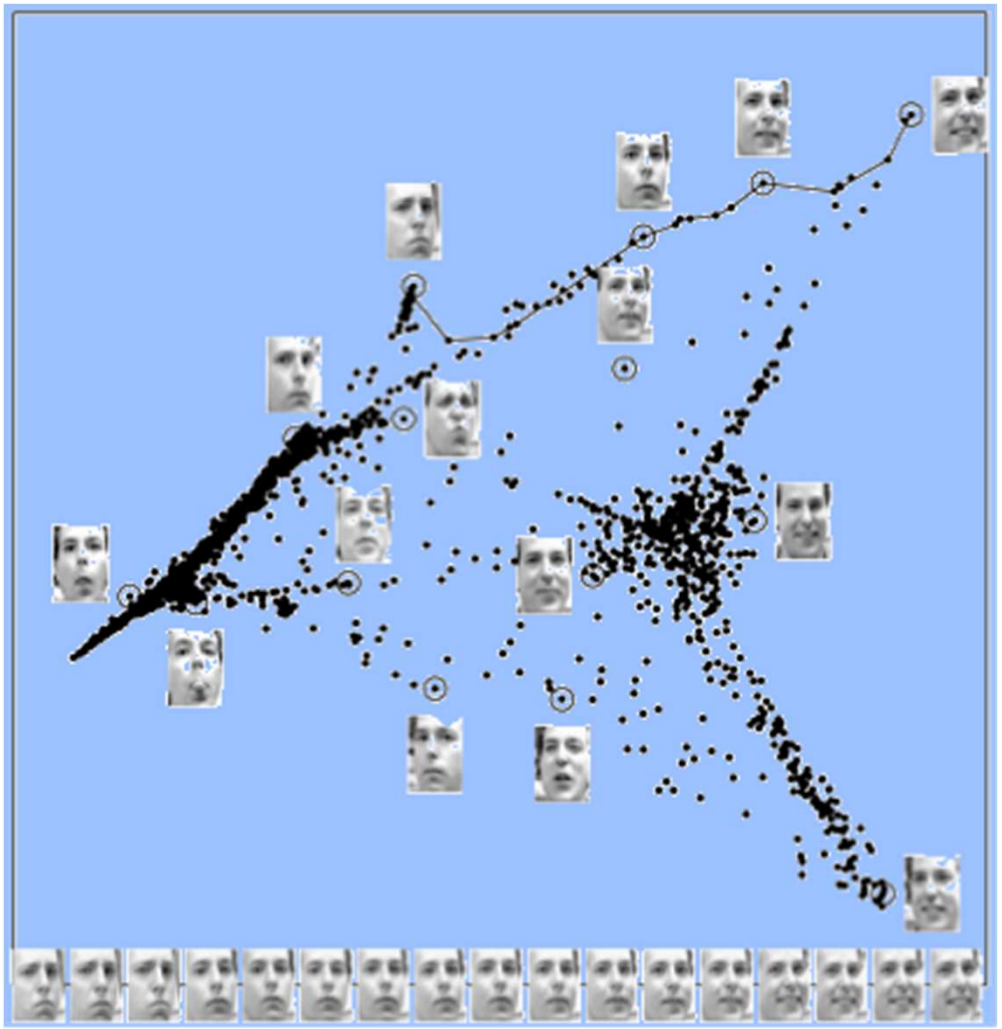
Sparse matrix diagonalization

Every step is relatively trivial, however the combined effect is quite complicated.

Examples of LLE

- Pose and expression

N = 1965
k = 12
D = 560
d = 2



Properties of LLE

- **Strength**

Polynomial-time complexity (even cheaper than Isomap)

No local minima, no iterative procedure

Only free parameter is neighborhood size

- **Weakness**

Sensitive to short circuits

Distortion can be serious

No estimate of dimensionality

Problems of LLE

distortion in along boundaries

bottom eigenvalues, no telltale cutoff point

Recap: Isomap vs. LLE

Isomap	LLE
Preserve geodesic distance	Preserve local symmetry
construct nearest neighbor graph; formulate quadratic form; diagonalize	construct nearest neighbor graph; formulate quadratic form; diagonalize
pick top eigenvector; estimate dimensionality	pick bottom eigenvector; does not estimate dimensionality

There are still many

- Laplacian eigenmaps
- Hessian LLE
- Local Tangent Space Analysis
- ...

Outline

- Linear method: redux and new intuition
- Graph based spectral methods

Isomap

Locally linear embedding

- **Kernel methods**

Kernel PCA

Kernel CCA

- **Case studies**

Another twist on MDS to get nonlinearity

- **Key idea**

Map data points with nonlinear functions

$$\phi : \boldsymbol{x} \rightarrow \phi(\boldsymbol{x})$$

Perform PCA/MDS in the new space

$$\phi(\boldsymbol{X})^{\mathrm{T}} \phi(\boldsymbol{X}) \boldsymbol{v} = \lambda \boldsymbol{v}$$

(MDS: diagonlizing Gram matrix)

The kernel trick

The inner product

$$\phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi(\boldsymbol{x}_j)$$

is more relevant than the exact form of the mapping function.

For certain mapping function, we can find a kernel function

$$\boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi(\boldsymbol{x}_j)$$

Therefore, all we need to do is to specify a kernel function to find the projections!

Kernel PCA

- **Algorithm**

Select a kernel: Gaussian kernel, string kernel

Construct kernel matrix $\boldsymbol{K} = [K_{ij}] = [K(\boldsymbol{x}_i, \boldsymbol{x}_j)]$

Diagonalize the kernel matrix

- **Caveat**

Kernel PCA does not always reduce dimensions.

Very important in choosing appropriate kernel

Pre-image problem

Why would you care about kernels?

- **Handle complex data types.**

Kernels for numerican data (eg., CPU load)

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/\sigma\right)$$

"String" kernels for text data (eg. URL/http request)

$$K(s_i, s_j) = \#\ of\ common\ substrings$$

- **Building blocks**

Multiple kernels can be combined into a single kernel.

Visualize related information

- **Canonical correlation analysis (CCA)**

  Works on two or multiple data sets

  Find projections that exhibit strongest correlation across data sets

  Solve generalized eigenvalue problems

- **Kernel CCA**

  Kernelized CCA

Outline

- Linear method: redux and new intuition
- Graph based spectral methods

Isomap

Locally linear embedding

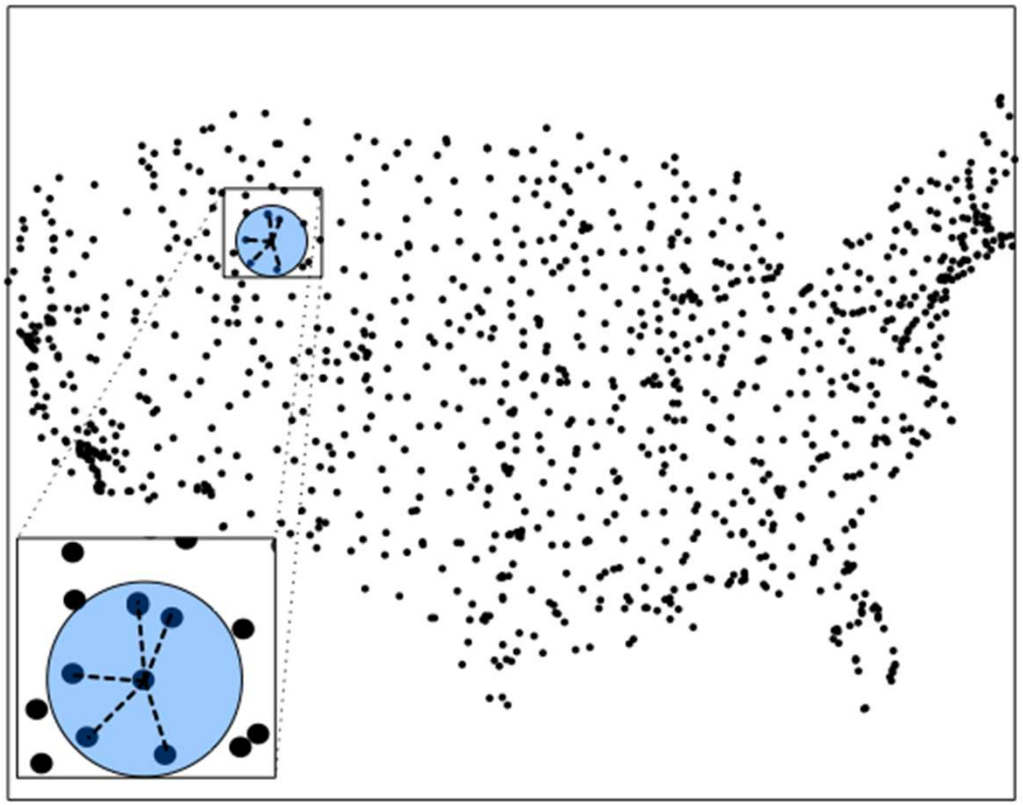
- Kernel methods

Kernel PCA

Kernel CCA

- **Case studies**

Case study (I): sensor network localization



cities

cities $\begin{bmatrix} 0 & d_{12} & ? & d_{14} \\ d_{21} & 0 & d_{23} & ? \\ ? & d_{32} & 0 & d_{34} \\ d_{41} & ? & d_{43} & 0 \end{bmatrix}$

sensors distributed in US cities.
Infer coordinates from limited measurement of distances

(Weinberger, Sha & Saul, NIPS 2006)

Embedding in 2D while ignoring distances

Turn distance matrix into adjacency matrix
Compute 2D embedding with Laplacian eigenmaps

Assumption: measurements exist only if sensors are close to each other

Adding distance constraints

Start from Lapalcian eigenmap results

Enforce known distances constraints

Find embedding using maximum variance unfolding

Recover almost perfectly!

Case study (II): workload & behavior

(Work in progress by Ganapathi et al @ Radlab)

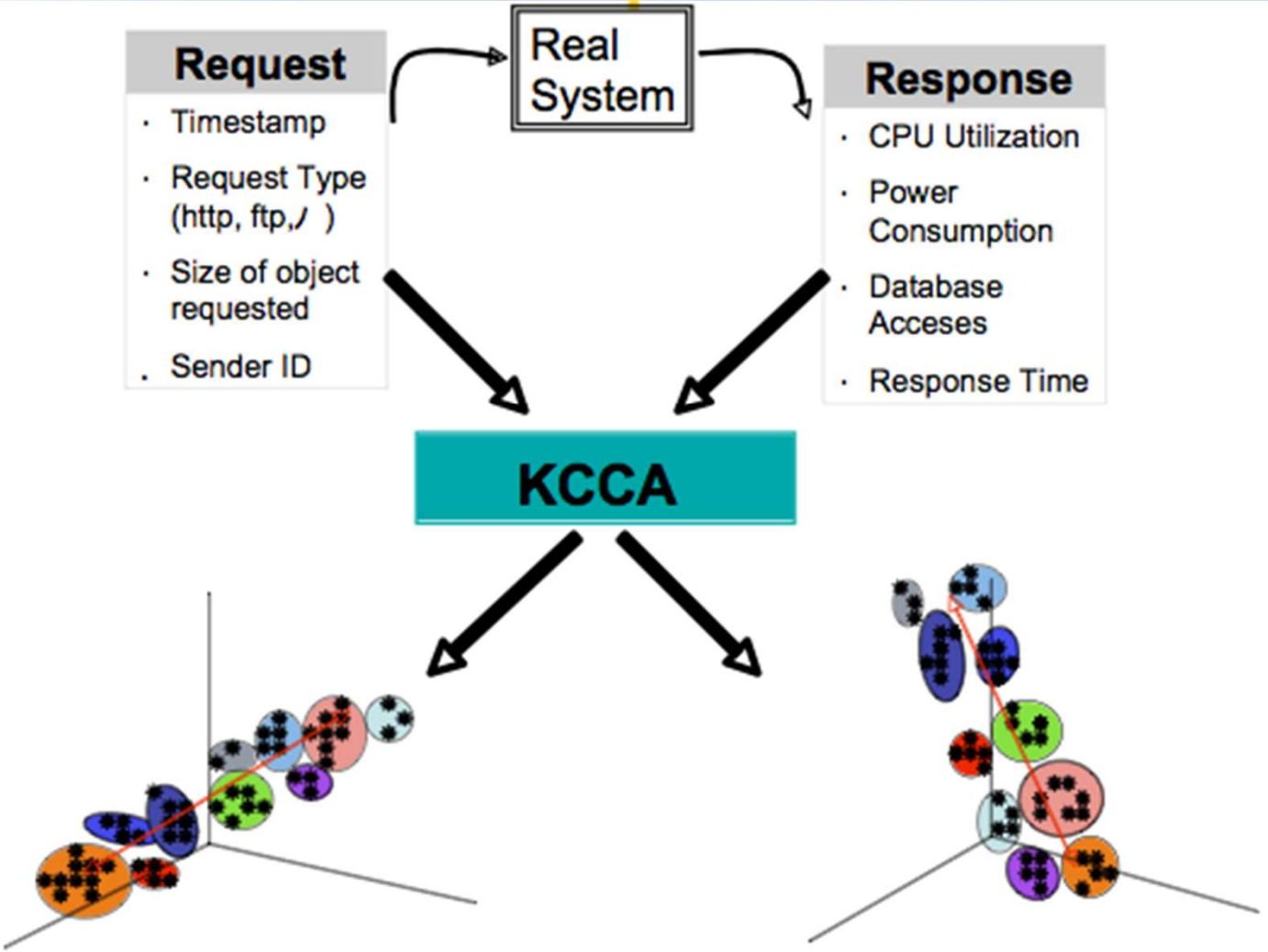
- Web Services are composed of a variety of components (web servers, DBs, routers,…)
  - How can we describe their interactions?
- What we observe
  - User requests coming into the system
  - Resource utilization and Performance metrics gathered from various components
  - Faults/Failures/everything that can go wrong

Goal

Automatically extract realistic model of system workload and how it affects system behavior

- Def'n: Workload = user request traces
- Def'n: Behavior = time series of fine grained system metrics

Using KCCA to tackle our problem



Sample Trace - Requests

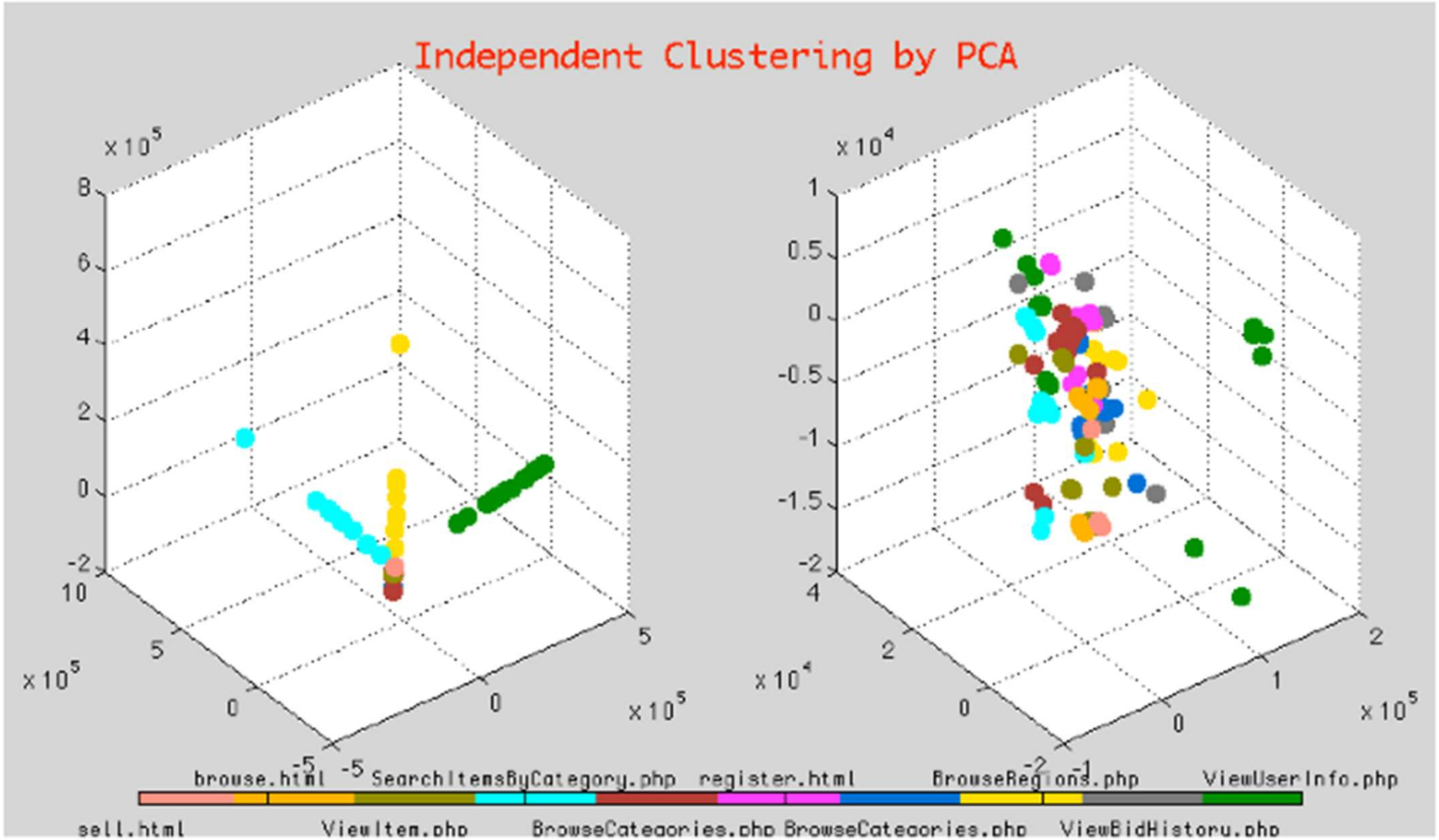
[03/May/2007:01:00:46] Rubis vm105.vm-rubis-web2 6/0/2/23/31 200 1905 - - ---- 36/37/37 0/0 "GET //PHP/sell.html HTTP/1.0"
[03/May/2007:01:00:46] Rubis vm105.vm-rubis-web2 0/0/3/26/29 200 1905 - - ---- 35/36/36 0/0 "GET //PHP/sell.html HTTP/1.0"
[03/May/2007:01:00:46] Rubis vm105.vm-rubis-web2 0/0/5/26/31 200 1905 - - ---- 34/35/35 0/0 "GET //PHP/sell.html HTTP/1.0"
[03/May/2007:01:00:46] Rubis vm105.vm-rubis-web2 0/0/3/27/32 200 1905 - - ---- 33/34/34 0/0 "GET //PHP/sell.html HTTP/1.0"
[03/May/2007:01:00:46] Rubis vm105.vm-rubis-web2 0/0/12/20/32 200 1905 - - ---- 32/33/33 0/0 "GET //PHP/sell.html HTTP/1.0"
[03/May/2007:01:05:51] Rubis vm105.vm-rubis-web2 0/0/0/18/18 200 1623 - - ---- 8/9/9 0/0 "GET //PHP/browse.html HTTP/1.0"
[03/May/2007:01:05:51] Rubis vm105.vm-rubis-web2 0/0/0/22/22 200 1623 - - ---- 6/7/7 0/0 "GET //PHP/browse.html HTTP/1.0"
[03/May/2007:01:05:51] Rubis vm105.vm-rubis-web2 0/0/0/20/20 200 1623 - - ---- 6/7/7 0/0 "GET //PHP/browse.html HTTP/1.0"
[03/May/2007:01:05:51] Rubis vm105.vm-rubis-web2 0/0/0/22/23 200 1623 - - ---- 5/6/6 0/0 "GET //PHP/browse.html HTTP/1.0"
[03/May/2007:01:05:51] Rubis vm105.vm-rubis-web2 0/0/1/22/23 200 1623 - - ---- 4/5/5 0/0 "GET //PHP/browse.html HTTP/1.0"
[03/May/2007:01:10:59] Rubis vm105.vm-rubis-web2 0/0/0/193/194 200 2494 - - ---- 29/30/30 0/0 "GET //PHP/ViewItem.php?itemId=89 HTTP/1.0"
[03/May/2007:01:10:59] Rubis vm105.vm-rubis-web2 0/0/0/228/228 200 2546 - - ---- 24/25/25 0/0 "GET //PHP/ViewItem.php?itemId=5670 HTTP/1.0"
[03/May/2007:01:10:59] Rubis vm105.vm-rubis-web2 0/0/0/232/233 200 2500 - - ---- 23/24/24 0/0 "GET //PHP/ViewItem.php?itemId=7718 HTTP/1.0"
[03/May/2007:01:10:59] Rubis vm105.vm-rubis-web2 0/0/0/237/237 200 2527 - - ---- 22/23/23 0/0 "GET //PHP/ViewItem.php?itemId=9230 HTTP/1.0"
[03/May/2007:01:10:59] Rubis vm105.vm-rubis-web2 0/0/0/239/239 200 2548 - - ---- 21/22/22 0/0 "GET //PHP/ViewItem.php?itemId=12910 HTTP/1.0"
[03/May/2007:01:16:15] Rubis vm105.vm-rubis-web2 0/0/0/73/74 200 7318 - - ---- 40/41/41 0/0 "GET //PHP/SearchItemsByCategory.php?category=9&categoryName=Business%2C
[03/May/2007:01:16:15] Rubis vm105.vm-rubis-web2 0/0/0/79/80 200 7315 - - ---- 39/40/40 0/0 "GET //PHP/SearchItemsByCategory.php?category=18&categoryName=Business%2
[03/May/2007:01:16:15] Rubis vm105.vm-rubis-web2 0/0/0/92/92 200 7313 - - ---- 42/43/43 0/0 "GET //PHP/SearchItemsByCategory.php?category=8&categoryName=Business%2C
[03/May/2007:01:16:15] Rubis vm105.vm-rubis-web2 0/0/0/93/93 200 7316 - - ---- 41/42/42 0/0 "GET //PHP/SearchItemsByCategory.php?category=12&categoryName=Business%2
[03/May/2007:01:22:31] Rubis vm105.vm-rubis-web2 0/0/0/190/192 200 3770 - - ---- 3/3/3 0/0 "GET //PHP/BrowseCategories.php?region=48 HTTP/1.0"
[03/May/2007:01:22:31] Rubis vm105.vm-rubis-web2 0/0/0/197/198 200 3770 - - ---- 2/2/2 0/0 "GET //PHP/BrowseCategories.php?region=40 HTTP/1.0"
[03/May/2007:01:22:31] Rubis vm105.vm-rubis-web2 0/0/0/267/268 200 3770 - - ---- 1/1/1 0/0 "GET //PHP/BrowseCategories.php?region=11 HTTP/1.0"
[03/May/2007:01:22:31] Rubis vm105.vm-rubis-web2 0/0/0/222/223 200 3750 - - ---- 0/0/0 0/0 "GET //PHP/BrowseCategories.php?region=7 HTTP/1.0"
[03/May/2007:01:26:50] Rubis vm105.vm-rubis-web2 0/0/0/4/6 200 2364 - - ---- 0/0/0 0/0 "GET //PHP/register.html HTTP/1.0"
[03/May/2007:01:26:50] Rubis vm105.vm-rubis-web2 0/0/0/15/15 200 2364 - - ---- 2/2/2 0/0 "GET //PHP/register.html HTTP/1.0"
[03/May/2007:01:26:50] Rubis vm105.vm-rubis-web2 0/0/0/15/15 200 2364 - - ---- 1/1/1 0/0 "GET //PHP/register.html HTTP/1.0"
[03/May/2007:01:26:50] Rubis vm105.vm-rubis-web2 0/0/0/2/17 200 2364 - - ---- 0/0/0 0/0 "GET //PHP/register.html HTTP/1.0"
[03/May/2007:01:26:52] Rubis vm105.vm-rubis-web2 0/0/0/5/6 200 2364 - - ---- 10/10/10 0/0 "GET //PHP/register.html HTTP/1.0"

Sample Trace - Metrics

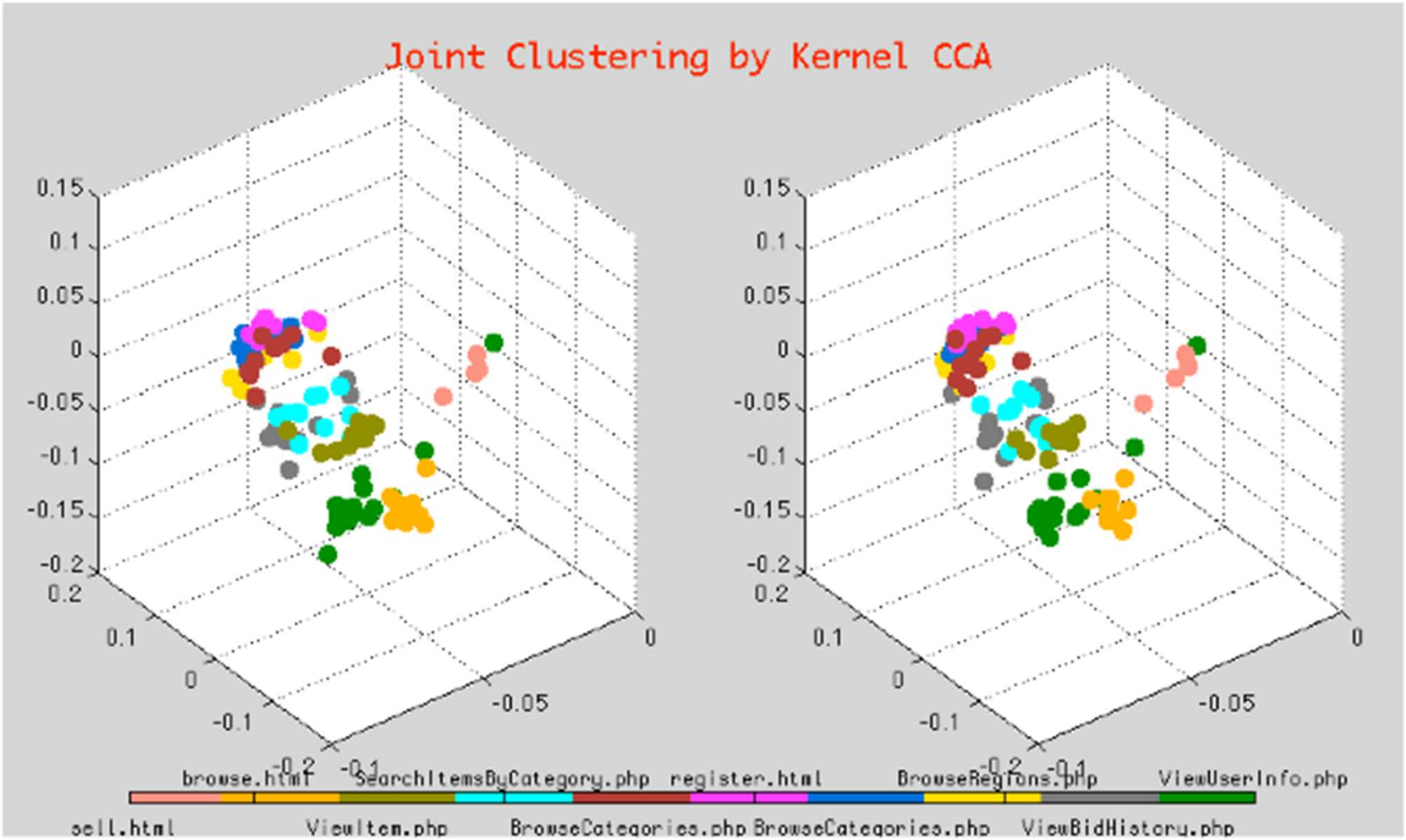
Web server metrics

DB metrics

PCA on the data



KCCA on the data (Demo)



Conclusion

- **Big picture**

Large-scale high dimensional data everywhere.

Many of them have intrinsic low dimension representation.

Nonlinear techniques can be very helpful for exploratory data analysis and visualization.

- **Techniques we sampled today**

Manifold learning techniques.

Kernel methods.

Credits

- Lecture based on Fei Sha's presentation at Yahoo! Labs Radlab summer

- Part of this lecture is based on manifold learning tutorials by Lawrence K. Saul (UCSD)

  http://www.cs.ucsd.edu/~saul/tutorials.html

- Pictures borrowed from various sources

- Slides by Archana Ganapathi @ Radlab