Lecture 5-6: MDS, LLE, Intrinsic dimensions

Multi-dimensional scaling

MDS is a technique motivated by 2-problems in understanding data in high dimensional spaces. Its objective is to project an ensemble of data points into 1, 2, or 3-dimensional spaces so that the spatial distance of these data points are preserved.

Thus, MDS is used for two purposes:

1). Visualize the structures and properties of data, so that we may select proper models for them.

2). Verify some distance (metric) measure on an unknown dataset.

With a good distance measure, the data of a class should correspond to a "meaningful" cluster, e.g., in image database retrieval, or art authentication.

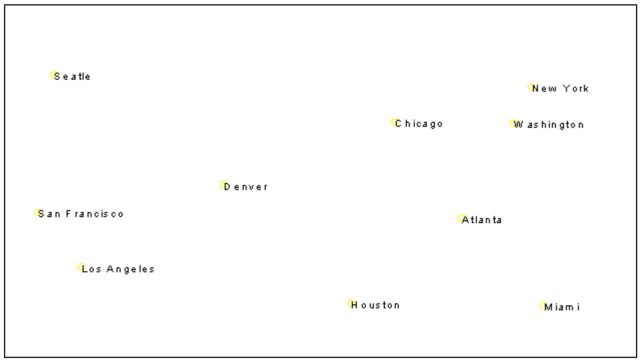
Example I: distance visualization

	1	2	3	4	5	6	7	8	9	10
1. Atlanta	0	587	1212	701	1936	604	748	2139	2182	543
2. Chicago	587	0	920	940	1745	1188	713	1858	1737	597
3. Denver	1212	920	0	879	831	1726	1631	949	1021	1494
4. Houston	701	940	879	0	1374	968	1420	1645	1891	1220
5. Los Angeles	1936	1745	831	1374	0	2339	2451	347	959	2300
6. Miami	604	1188	1726	968	2339	0	1092	2594	2734	923
7. New York	748	713	1631	1420	2451	1092	0	2571	2408	205
8. San Francisco	2139	1858	949	1645	347	2594	2571	0	678	2448
9. Seatle	2182	1737	1021	1891	959	2734	2408	678	0	2329
10. Washington DC	543	597	1494	1220	2300	923	205	2442	2329	0

Airline distances between ten U.S. cities.

Reconstructed 2D Map

One computes the (x,y) coordinates for the 10 cities that best preserve the distance matrix.



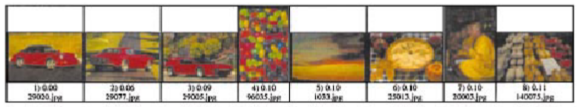
Example II: mapping mental representations

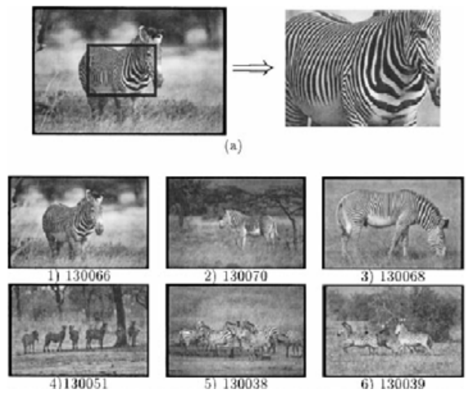
E.g.1 Color map,

Another example is to map various colors in a 2D matrix so that some perceptual distances are preserved. I am sorry that we cannot print out color, but the pdf file will be in color. One can calculate a perceptual color distance by psychology experiments, then obtains a distance matrix, like the city matrix, then we can map colors in 2D

Example III: image query

E.g.2 Image search
(face search in
a dating website),





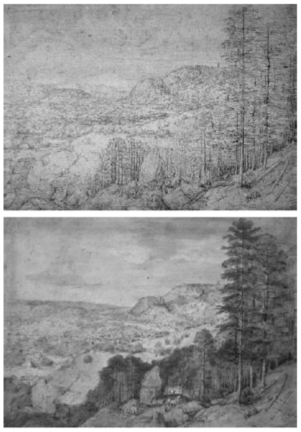
From Rubner, Tomasi, and Guibus, 00

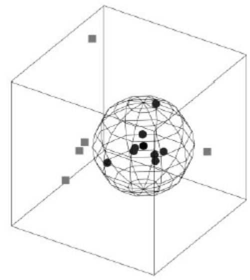
Example III: 2D mapping of images

Each image is treated as a vector point and some distance is defined between any two images. Then one project them into a 2D map to visualize the structures

Lecture note for Stat 231-CS276A

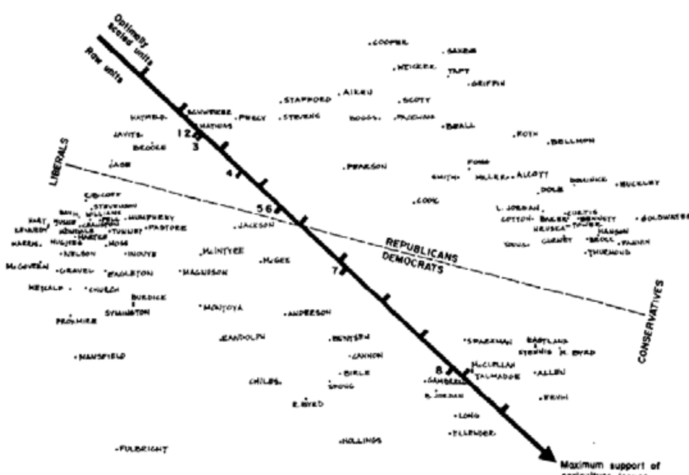
Example V: Art Authentication





S. Lyu, D. Rockmore, and H. Farid, PNAS, 2004

Example VI: Senator map by MDS



Formulation of MDS

Given: a set of data points in d-space $\{x_1, x_2, \ldots, x_n\}$
a dissimilarity / distance measure/metric between two points x_i, x_j: δ_{ij}

Objective: find points in 1,2, or 3-space $\{y_1, y_2, \ldots, y_n\}$ with usually Eclidean distances d_{ij} for two points y_i and y_j.

A criterion (Kruskal 1964) is to minimize

$$\text{Stress} = \frac{\sum_{i,j}(d_{ij} - \delta_{ij})^2}{\sum_{i,j}\delta_{ij}^2}$$

MDS for non-metric data

In some applications, the quantitative distance or dissimilarity is less important than the rank order. Thus an MDS mapping criterion will be a *monotonic constraint* that the project points preserve the rank order of the original data points.

Suppose we re-order the m=n(n-1)/2 distance in the original data

$$\delta_{i_1,j_1} \leq \cdots \leq \delta_{i_m,j_m}$$

For any m numbers that preserve the monotonic constraints,

$$\hat{d}_{i_1,j_1} \leq \cdots \leq \hat{d}_{i_m,j_m}$$

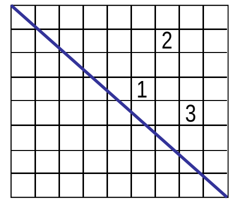
We define a criterion for the projected points as,

$$(y_1, \cdots y_n) = \arg\min J_{\text{mon}}$$

$$J_{\text{mon}} = \frac{\min_{\hat{d}} \sum_{i<j}(d_{ij} - \hat{d}_{ij})^2}{\sum_{i<j} d_{ij}^2}$$

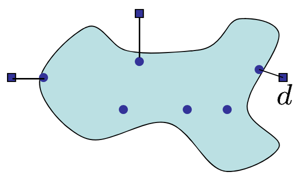
Non-metric MDS

Let δ be the original dissimilarity matrix on x, and d the matrix for y.
$\hat{d}$ the matrix that have the same rank order as δ



The set of matrices $\hat{d}$ that satisfy the rank order criterion is illustrated By the shadowed area. Each point is a matrix.

Our objective is to find y and therefore d so that it d has the shortest distance to this set.
In comparison to the previous MDS, this gives more flexibility in computing y.



$$d = d(y_1, y_2, ..., y_n)$$

Lecture note for Stat 231-CS276A © S.C. Zhu

LLE: local linear embedding

Dimension reduction techniques can be classified in three axes:

1. Generative (e.g. PCA) vs. discriminative (e.g. Fisher's linear discriminant)

2. Linear (e.g. PCA, Fisher) vs. Non-linear (e.g. MDS)

3. Global (projection e.g. PCA) vs. Local (nearest neighbor e.g. LLE below).

In this lecture, we introduce a local linear embedding (LLE) method by Roweis and Saul 2000 which is a generative, non-linear, and local technique for dimension reduction.

Some figures in this lecture are extracted from the Roweis and Saul 2000 papers.

Lecture note for Stat 231-CS276A © S.C. Zhu

Low-dimensional manifold in high dimensional space

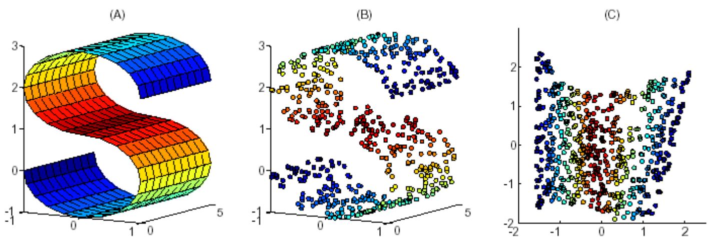


Figure 1: The problem of nonlinear dimensionality reduction, as illustrated for three dimensional data (B) sampled from a two dimensional manifold (A). An unsupervised learning algorithm must discover the global internal coordinates of the manifold without signals that explicitly indicate how the data should be embedded in two dimensions. The shading in (C) illustrates the neighborhood-preserving mapping discovered by LLE.

Example

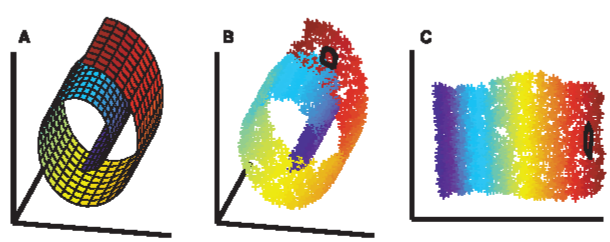
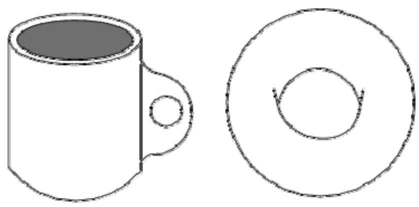


Fig. 1. The problem of nonlinear dimensionality reduction, as illustrated (*10*) for three-dimensional data (B) sampled from a two-dimensional manifold (**A**). An unsupervised learning algorithm must discover the global internal coordinates of the manifold without signals that explicitly indicate how the data should be embedded in two dimensions. The color coding illustrates the neighborhood-preserving mapping discovered by LLE; black outlines in (**B**) and (**C**) show the neighborhood of a single point. Unlike LLE, projections of the data by principal component analysis (PCA) (*28*) or classical MDS (*2*) map faraway data points to nearby points in the plane, failing to identify the underlying structure of the manifold. Note that mixture models for local dimensionality reduction (*29*), which cluster the data and perform PCA within each cluster, do not address the problem considered here: namely, how to map high-dimensional data into a single global coordinate system of lower dimensionality.

What is a manifold

From mathworld.wolfram.com

A manifold is a topological space that is locally Euclidean (i.e., around every point, there is a neighborhood that is topologically the same as the open unit ball). To illustrate this idea, consider the ancient belief that the Earth was flat as contrasted with the modern evidence that it is round. The discrepancy arises essentially from the fact that on the small scales that we see, the Earth does indeed look flat. In general, any object that is nearly "flat" on small scales is a manifold, and so manifolds constitute a generalization of objects we could live on in which we would encounter the round/flat Earth problem, as first codified by Poincaré. More formally, any object that can be "charted" is a manifold.



a torus   another torus

Algorithm

LLE ALGORITHM

1. Compute the neighbors of each data point, $\vec{X}_i$.
2. Compute the weights W_{ij} that best reconstruct each data point $\vec{X}_i$ from its neighbors, minimizing the cost in eq. (1) by constrained linear fits.
3. Compute the vectors $\vec{Y}_i$ best reconstructed by the weights W_{ij}, minimizing the quadratic form in eq. (2) by its bottom nonzero eigenvectors.

Find W to minimize

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

$$\sum_j W_{ij} = 1.$$

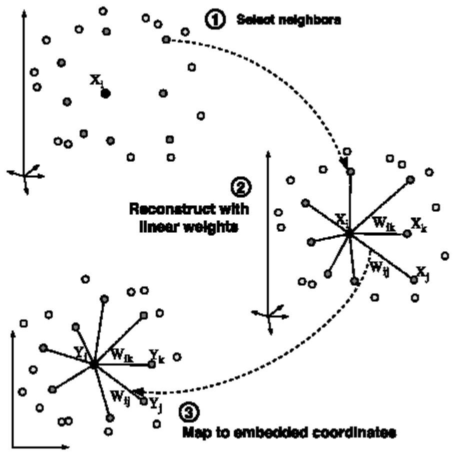
Find Y to minimize

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

W is supposed to preserve the local structures.

Illustration of Algorithm

Fig. 2. Steps of locally linear embedding: (1) Assign neighbors to each data point $\vec{X}_i$ (for example by using the K nearest neighbors). (2) Compute the weights W_{ij} that best linearly reconstruct $\vec{X}_i$ from its neighbors, solving the constrained least-squares problem in Eq. 1. (3) Compute the low-dimensional embedding vectors $\vec{Y}_i$ best reconstructed by W_{ij}, minimizing Eq. 2 by finding the smallest eigenmodes of the sparse symmetric matrix in Eq. 3. Although the weights W_{ij} and vectors Y_i are computed by methods in linear algebra, the constraint that points are only reconstructed from neighbors can result in highly nonlinear embeddings.



Example on faces

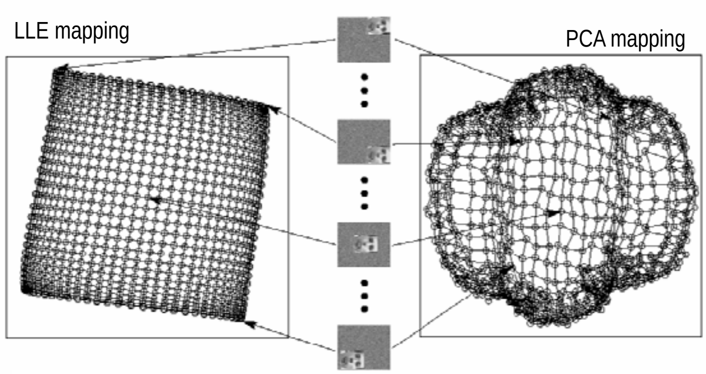


Figure 3: The results of PCA (top) and LLE (bottom), applied to images of a single face translated across a two-dimensional background of noise. Note how LLE maps the images with corner faces to the corners of its two dimensional embedding, while PCA fails to preserve the neighborhood structure of nearby images.

Example on images

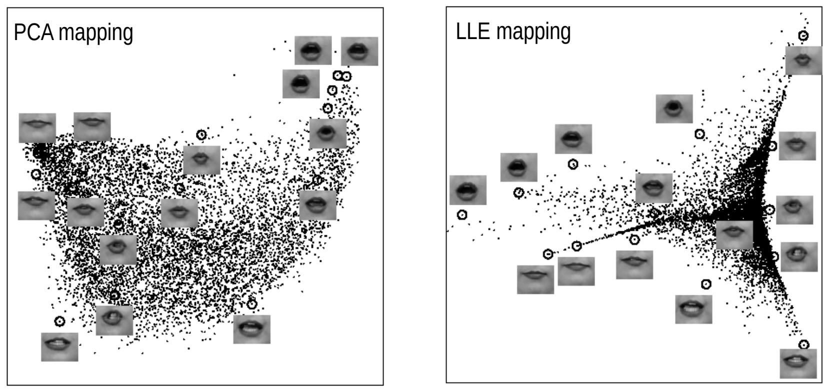


Figure 4: Images of lips mapped into the embedding space described by the first two coordinates of PCA (top) and LLE (bottom). Representative lips are shown next to circled points in different parts of each space. The differences between the two embeddings indicate the presence of nonlinear structure in the data.

Example

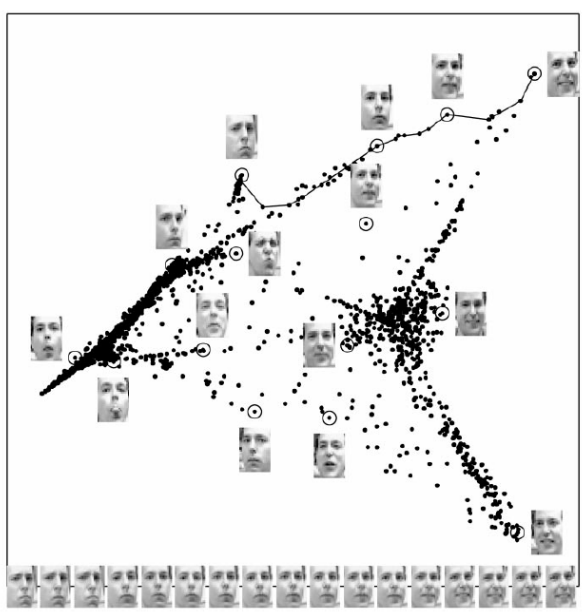
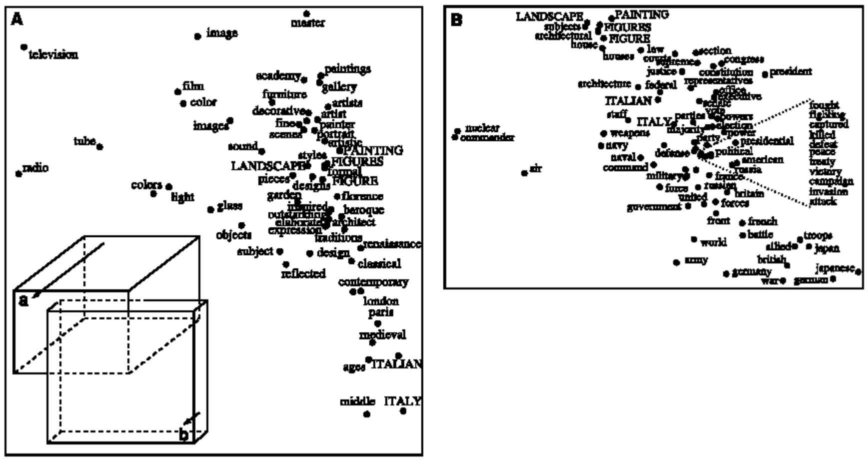


Fig. 3. Images of faces (*11*) mapped into the embedding space described by the first two coordinates of LLE. Representative faces are shown next to circled points in different parts of the space. The bottom images correspond to points along the top-right path (linked by solid line), illustrating one particular mode of variability in pose and expression.

Example on word semantics

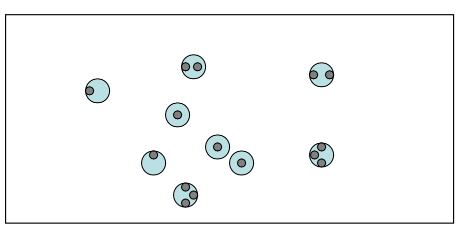


What is the intrinsic dimension of a data cloud?

Data in very high dimensional feature spaces often reside in much lower dimensional manifolds. To measure the intrinsic dimension of a data set, one starts with a measure of *volume* (or *massiveness*) of the set. This is often done by the ε-cover.

Let $D=\{x\}$ be the dataset, and ρ a metric in the feature space, $S=\{y\}$ be a cover so that

$$\forall x \in D, \exists y \in S, \text{ and } \rho(x,y) \leq \epsilon.$$



Pre-condition: the data are from a space of fixed dimensions, not a mixture of many subspaces.

Kolmogorov Capacity Dimension

Let N(e) be the minimum e-cover of the dataset D, we define a Kolmogorov capacity dimension (or Box counting dimension) by

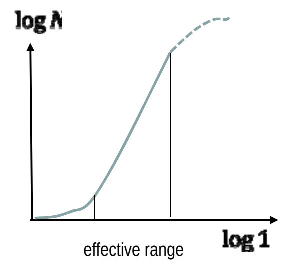
$$D_{\text{cap}} = \lim_{\epsilon \to 0} \frac{\log N(\epsilon)}{\log 1/\epsilon}$$

In other word, the number (volume) has an exponential rate

$$N(\epsilon) \sim (1/\epsilon)^{D_{\text{cap}}}$$

Or we have a linear relation in a log-log plot

$$\log N(\epsilon) = D_{\text{cap}} \log 1/\epsilon$$



Fractal diemensions: Cantor set (d=$\log_3^{2}$) and Koch curve (d=$\log_3^{4}$) Discussed on board.

Information Dimension

The capacity dimension assumes a uniform probability for each ball. If this is not uniform, we have a modified version called the information dimension,

$$D_{\text{inf}} = \lim_{\epsilon \to 0} \frac{\sum_{y \in S} p(y;\epsilon) \log 1/p(y;\epsilon)}{\log 1/\epsilon}$$

Where It is easy to check that

$$D_{\text{cap}} \geq D_{\text{inf}}$$

Theorem:

$$D_{\text{corr}} \leq D_{\text{inf}} \leq D_{\text{cap}}$$

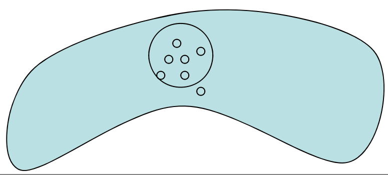
Correlation dimension

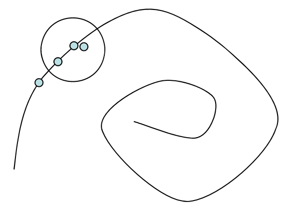
Given N data points, $\{x_1, x_2, ..., x_N\}$

$$C(\epsilon) = \lim_{N\to\infty} \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \mathbf{1}(|x_i - x_j| < \epsilon)$$

The correlation dimension is

$$D_{\text{corr}} = \lim_{\epsilon\to 0} \frac{\log C(\epsilon)}{\log \epsilon}$$



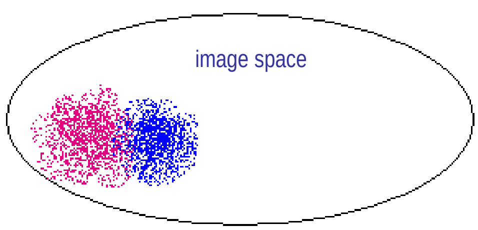


Intuitively, the higher dimension the manifold is, the more neighbors a point will have.

Stochastic sets in the image space

How do we define concepts as sets of image/video:

e.g. noun concepts: human face, vehicle, chair?
verbal concept: opening door, making coffee?



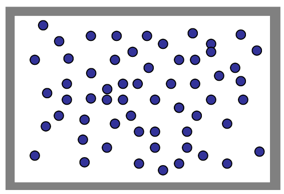
A point is an image or a video clip

Method 1, Stochastic set in statistical physics

Statistical physics studies macroscopic properties of systems that consist of massive elements with microscopic interactions.

e.g.: a tank of insulated gas or ferro-magnetic material

$N = 10^{23}$



Micro-canonical Ensemble

A state of the system is specified by the position of the N elements X^N and their momenta p^N

$$S = (x^N,\ p^N)$$

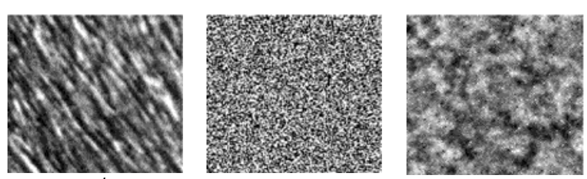
But we only care about some global properties
Energy E, Volume V, Pressure, ….

$$\text{Micro-canonical Ensemble} = \Omega(N, E, V) = \{\, s : \ h(S) = (N, E, V) \,\}$$

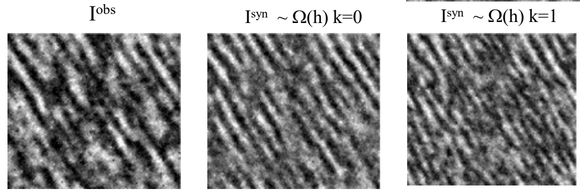
Example in texture modeling and definition

$$\text{a texture} = \Omega(h_c) = \{\, I : \ h_i(I) = h_{c,i} \ , i = 1,2,\ldots,K \,\}$$

h_c are histograms of Gabor filters



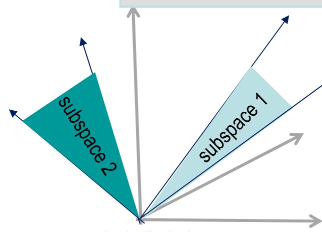
I^{obs} $I^{syn} \sim \Omega(h)$ k=0 $I^{syn} \sim \Omega(h)$ k=1



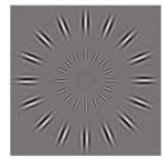
$I^{syn} \sim \Omega(h)$ k=3 $I^{syn} \sim \Omega(h)$ k=4 $I^{syn} \sim \Omega(h)$ k=7

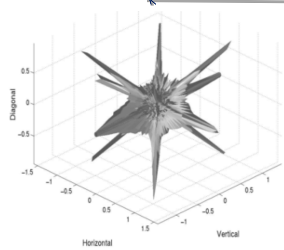
Method 2, Lower dimensional sets or subspaces

$$\text{a texton} = \Omega(h_c) = \{ I: \; I = \sum_i \alpha_i \psi_i, \;\; \|\alpha\|_0 < k \}$$



K is far smaller than the dimension of the image space.
φ is a basis function from a dictionary.





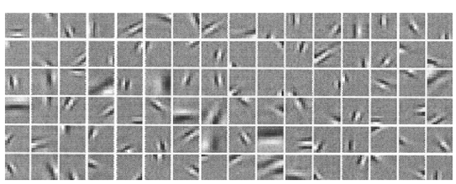
Here is an example of how real world data can be truly complex – non-Gaussian and highly kurtotic. This is an iso-density contour for a 3D histogram of log(range) images (2x2 patches minus their means) (Brown range image database, thesis of James Huang)

Sparsity and harmonic analysis

Stochastic set from sparse coding

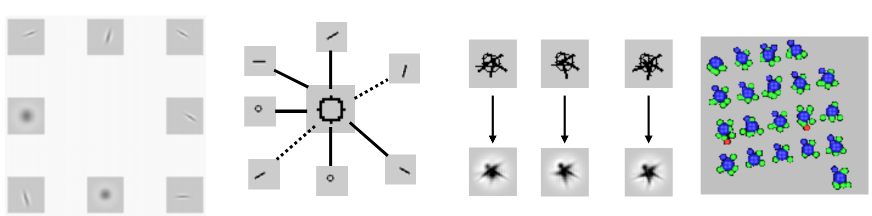
Learning an over-complete image basis from natural images

$$I = \Sigma_i \alpha_i \psi_i + n$$



(Olshausen and Fields, 1995-97)

Textons



B. Olshausen and D. Fields, "Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1?" *Vision Research, 37*: 3311-25, 1997.
S.C. Zhu, C. E. Guo, Y.Z. Wang, and Z.J. Xu, "What are Textons?" *Int'l J. of Computer Vision,* vol.62(1/2), 121-143, 2005.

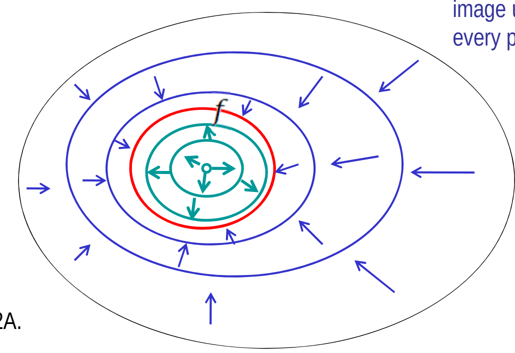
Advanced Topics: learning by Manifold pursuit

f : target distribution; *p*: our model; *q*: initial model

$$q = p_0 \to p_1 \to \cdots \to p_k \quad to \quad f$$

1, $q = unif()$

2, $q = \delta()$



To be taught in Stat232A.

Intuitive ideas: a professor grading an exam

The full score (like dimension in our case) is 100. You have two ways:

For top students (high dimensional manifolds), you start from 100 and deduct points :

100 – 2 – 0 – 0 – 3 – 0– 2 – 0 – 0 – 0 – 0 – 0 – 1 = 92

For bottom students (low dimensional manifolds), you start from 0 and add points

0 + 8 + 0 + 0 + 3 + 0 + 2 + 0 + 0 + 5 + 0 + 0 + 1 = 19

In reality, suppose the exam is very long (just like the large image has >1M pixels), a student may have mixed performance, e.g. doing excellent in the 1st half and doing poorly in the 2nd half. Thus a most effective way is to use the two methods for different sections of the exam.

(50 – 2 – 0 – 0 – 3 – 0) + (0 + 5 + 3 + 0 + 0 + 2) = 45 + 10 = 55

In fact, most of the object categories are middle entropy manifolds and have mixed structures.

Manifold pursuit

In a simple case: *f* is a Gaussian distribution