EECS 275 Matrix Computation

Ming-Hsuan Yang

Electrical Engineering and Computer Science
University of California at Merced
Merced, CA 95344
http://faculty.ucmerced.edu/mhyang

Lecture 23

Overview

- Multidimensional scaling
- Spectral methods for dimensionality reduction
- Spectral graph theory
- Spectral clustering
- Random walk

Principal component analysis (PCA)

- Compute the low dimensional representation of high dimensional data
- The input $\mathbf{x}_i \in \mathbb{R}^m$ are projected into the d-dimensional subspace that minimizes the reconstruction error

$$\mathcal{E}_{PCA} = \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{\alpha=1}^{m} (\mathbf{x}_i \cdot \mathbf{y}_\alpha)\mathbf{y}_\alpha \right\|^2$$

- The basis vectors $\mathbf{y}$ of the subspace are given by the top q eigenvectors of the $q \times q$ covariance matrix (assume $\mathbf{x}_i$ is centered)

$$C = \frac{1}{n}\sum_{i}^{n} \mathbf{x}_i \mathbf{x}_i^\top = \frac{1}{n}\sum_{i}^{n} \mathbf{x}_i \otimes \mathbf{x}_i = \frac{1}{n} X X^\top$$

  where $\otimes$ is the outer product operator, and X is a matrix of all data points, $X = [\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_n]$
- Based on the second order statistics

Multidimensional scaling (MDS)

- Compute the low dimensional representation $\phi \in \mathbb{R}^q$ of a high dimensional data $\mathbf{x} \in \mathbb{R}^m$ that most faithfully preserves pairwise distances (or similarities which are inversely proportional to distances)
- Euclidean distance between two points

$$d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = (\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j)$$

- The solution is obtained by minimizing

$$\mathcal{E}_{MDS} = \sum_i \sum_j (\mathbf{x}_i \cdot \mathbf{x}_j - \phi_i \cdot \phi_j)^2$$

and the minimum error is obtained from the spectral decomposition of the $n \times n$ Gram matrix of inner products

$$G = X^\top X, \qquad G_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j$$

Denoting the top q eigenvectors of the Gram matrix by $\{\mathbf{u}_\alpha\}_{\alpha=1}^m$ and their respective eigenvalues by $\{\lambda_\alpha\}_{\alpha=1}^m$, the outputs of MDS are given by $\phi_{i\alpha} = \sqrt{\lambda_\alpha}\mathbf{u}_{\alpha i}$

MDS: derivation

- Assume the centroid of the configuration of n points is at the origin

$$\sum_{i=1}^{n} \mathbf{x}_{ij} = 0,\ j = 1, \ldots, m$$

- To find the Gram matrix G, from

$$d_{ij} = (\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_i + \mathbf{x}_j^\top \mathbf{x}_j - 2\mathbf{x}_i^\top \mathbf{x}_j$$

, and hence

$$\begin{aligned}
\tfrac{1}{n}\textstyle\sum_{i=1}^{n} d_{ij}^2 &= \tfrac{1}{n}\textstyle\sum_{i=1}^{n} \mathbf{x}_i^\top \mathbf{x}_i + \mathbf{x}_j^\top \mathbf{x}_j \\
\tfrac{1}{n}\textstyle\sum_{j=1}^{n} d_{ij}^2 &= \mathbf{x}_i^\top \mathbf{x}_i + \tfrac{1}{n}\textstyle\sum_{j=1}^{n} \mathbf{x}_j^\top \mathbf{x}_j \\
\tfrac{1}{n^2}\textstyle\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^2 &= \tfrac{2}{n}\textstyle\sum_{i=1}^{n} \mathbf{x}_i^\top \mathbf{x}_i
\end{aligned}$$

$$\begin{aligned}
G_{ij} &= \mathbf{x}_i^\top \mathbf{x}_j \\
&= -\tfrac{1}{2}(d_{ij}^2 - \tfrac{1}{n}\textstyle\sum_{i=1}^{n} d_{ij}^2 - \tfrac{1}{n}\textstyle\sum_{j=1}^{n} d_{ij}^2 + \tfrac{1}{n^2}\textstyle\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^2) \\
&= a_{ij} - a_{i.} - a_{.j} + a_{..}
\end{aligned}$$

where

$$a_{i.} = \frac{1}{n}\sum_{j=1} a_{ij},\ a_{.j} = \frac{1}{n}\sum_{i=1} a_{ij},\ a_{..} = \frac{1}{n^2}\sum_i \sum_j a_{ij}$$

MDS: derivation (cont'd)

- Define matrix A as $A_{ij} = a_{ij}$ and hence the Gram matrix G is

$$G = HAH$$

  where H is the centering matrix,

$$H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$$

  where $\mathbf{1} = (1, 1, \ldots, 1)^\top$, a vector of n ones
- G can be written in terms of spectral decomposition (e.g., SVD)

$$G = U\Sigma U^\top$$

- Since G has a most q non-zero singular values

$$G \approx U_q \Sigma_q U_q^\top = (U_q \Sigma_q^{1/2})(U_q \Sigma_q^{1/2})^\top$$

- $G \approx \Phi\Phi^\top$, the coordinate in lower dimensional space is $U_q \Sigma_q^{1/2}$
- MDS on Euclidean distance is equivalent to PCA

Isometric mapping (Isomap)

- Compute the low dimensional representation of a high dimensional data set that most faithfully preserves the pairwise *geodesic* distance [Tenenbaum et al. Science 00]
- Geodesic distances are approximated as measured along the submanifold from which the data points are sampled
- Can be understood as a variant of MDS in which estimates of geodesic distances along the submanifold are substituted (instead of Euclidean distance)
- Main steps:
  1. Construct adjacency graph: Find neighbors using K nearest neighbor or ϵ distance
  2. Estimate geodesic distance: Compute pairwise shortest distance using dynamic programming
  3. Metric MDS: Uncover the embedding from the top d eigenvectors of Gram matrix

Spectral graph theory

- Analyze graph structure and properties using linear algebra, i.e., the study of eigenvalues and eigenvectors of matrices associated graphs
- Related to random walk
- Applications: spectral clustering, shape matching, mesh compression, PageRank, etc.
- Given a graph $G = (V, E)$ and its weighted adjacency matrix W, we compute a diagonal matrix D

$$D_{ii} = \sum_j W_{ij}$$

the graph Laplacian is

$$D - W$$

and normalized graph Laplacian is

$$\mathcal{L} = D^{-1/2}(D - W)D^{-1/2} = I - D^{-1/2}WD^{-1/2}$$

Spectral graph theory (cont'd)

- Graph Laplacian
  - a symmetric, positive semidefinite matrix which can be thought as an operator on function defined on vertices of G
  - the eigenvalues of $\mathcal{L}$ are called *spectrum* of L (or the spectrum of the associated graph G)
  - used to find the properties of diameter of a graph, graph cut, etc.
  - used to determine the spectral embedding of the graph
- Recall Laplacian operator

$$\Delta = \nabla^2 = \nabla \cdot \nabla$$

  In Euclidean space

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

- See Fan R. K. Chung's book "Spectral Graph Theory" on the relationship of graph Laplacian and Laplace-Beltrami operator for Riemannian manifold
- See Daniel Spielman's lecture notes (`www.cs.yale.edu/homes/spielman/`)

Laplacian eigenmap

- Algorithm: Given n points in $\mathbb{R}^m$ [Belkin and Niyogi NIPS 02]
  1. Constructing the graph: nodes i and j are connected by an edge if $||\mathbf{x}_i - \mathbf{x}_j||^2 < \epsilon$ or based on K nearest neighbors
  2. Choosing the weights: compute the weighted graph

$$W_{ij} = e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{t}}$$

     where t is the kernel width (i.e., heat kernel)
  3. Compute Laplacian eigenmap: Assume G is connected, otherwise apply this step to each component
     Compute eigenvalues and eigenvectors for the generalized eigenvalue problem:

$$L\mathbf{y} = \lambda D\mathbf{y}$$

     where D is the diagonal matrix and $L = D - W$ is the graph Laplacian matrix
     Let $\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_{k-1}$ be the eigenvectors, ordered ascendingly to their eigenvalues. The image of $\mathbf{x}_i$ under the embedding into the lower dimensional space R^m is given by $(\mathbf{y}_1(i), \ldots, \mathbf{y}_m(i))$.

Corresponding continuum model

- Let $\mathcal{M}$ be a Riemannian manifold (isometrically) embedded in $\mathbb{R}^m$
- For a differential map $f : \mathcal{M} \to \mathbb{R}$

$$|f(\mathbf{x}') - f(\mathbf{x})| \sim \|\nabla f(\mathbf{x})\| \cdot d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}') + O(d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}'))$$

- The geodesic distance on $\mathcal{M}$ and the ambient Euclidean distance are locally similar

$$d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\| + O(\|\mathbf{x} - \mathbf{x}'\|)$$

- Choose f to preserve distance by minimizing

$$\int_{\mathcal{M}} \|\nabla f(\mathbf{x})\|^2 d\mathbf{x} \quad \text{subject to } \|f\|_{L^2(\mathcal{M})} = 1, \langle f, 1 \rangle_{L^2(\mathcal{M})} = 0$$

  where $d\mathbf{x}$ is the uniform measure on $\mathcal{M}$
- Minimizing $\int_{\mathcal{M}} \|\nabla f(\mathbf{x})\|^2$ corresponds to minimizing $Lf = \frac{1}{2} \sum_{ij} (f_i - f_j)^2 W_{ij}$ on a graph, i.e., finding eigenfunctions of the Laplace-Beltrami operator $\mathcal{L}$

Laplace Beltrami operator

- Graph Laplacian is analogous to the Laplace-Beltrami operator on manifolds
- Define $\mathcal{L}f = -\text{div}\ \nabla f$ where div is the divergence
- From Stokes' theorem

$$\int_{\mathcal{M}} \|\nabla f(\mathbf{x})\|^2 d\mathbf{x} = \int_{\mathcal{M}} f(\mathbf{x})(\mathcal{L}f)(\mathbf{x}) d\mathbf{x}$$

- That is $\mathcal{L}$ is positive semi-definite, with spectrum $0 = \lambda_0 \leq \lambda_1 \leq \cdots$ and corresponding eigenfunctions f_0, f_1, $\cdots$
- The embedding is given by

$$\mathbf{x} \to \mathbf{y} = (f_1(\mathbf{x}), \ldots, f_q(\mathbf{x}))$$

Spectral clustering

- See Tommi Jaakkola's lecture notes on spectral clustering
- Unified view of existing algorithms: [Weiss ICCV 99]
  - Feature grouping [Scott and Longuet-Higgins BMVC 90]
  - Multibody factorization [Costeria and Kanade ICCV 95]
  - Image segmentation [Shi and Malik CVPR 97]
  - Grouping [Perona and Freeman ECCV 98]
- Analysis of spectral clustering: [Ng et al. NIPS 01] [Kannan et al. JACM 04]
- Image segmentation: [Shi and Malik CVPR 97] [Meila and Shi NIPS 01]
- See also semi-supervised learning with spectral graph

Normalized graph Laplacian and random walk

- Given an undirected weighted graph $G = (V, E, W)$, the random walk on the graph is given by the transition matrix

$$P = D^{-1}W \tag{1}$$

where D is a diagonal matrix

$$D_{ii} = \sum_j W_{ij}$$

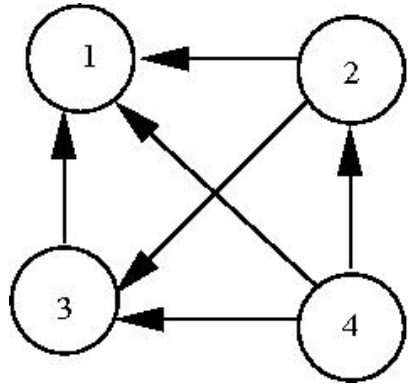
- Normalized graph Laplacian

$$\mathcal{L} = D^{-1/2}(D - W)D^{-1/2} = I - D^{-1/2}WD^{-1/2} \tag{2}$$

- The random walk matrix has the same eigenvalues as $I - \mathcal{L}$

$$D^{-1}W = D^{-1/2}(D^{-1/2}WD^{-1/2})D^{1/2} = D^{-1/2}(I - \mathcal{L})D^{1/2} \tag{3}$$

PageRank algorithm



- Interpret the weighted graph as transition matrix, P, where P_{ij} is the probability of jumping from j to i
- Suppose we have a set of four web pages with distribution $\boldsymbol{\pi} = [\pi_1\ \pi_2 \ldots \pi_n]^\top$ at the outset, The probability of reaching page 1 is

$$\begin{aligned} \pi_1 &= \tfrac{1}{2}\pi_2 + 1\pi_3 + \tfrac{1}{3}\pi_4 \\ \pi_3 &= 0 + \tfrac{1}{2}\pi_2 + \tfrac{1}{3}\pi_4 \\ \pi_1 &= \textstyle\sum_j P_{1j}\pi_j \\ \pi_i &= \textstyle\sum_j P_{ij}\pi_j \end{aligned}$$

Note that $\sum_i p_{ij} = 1$ and $\sum_i \pi_i = 1$

PageRank algorithm (cont'd)

- In matrix form

$$\boldsymbol{\pi} = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_n \end{bmatrix}$$

$$\boldsymbol{\pi} = P\boldsymbol{\pi}, \quad \boldsymbol{\pi}^\top \mathbf{e} = 1$$

  where $\mathbf{e} = [1\ 1 \ldots 1]^\top$
- Can be viewed as random walk or Markov chain

$$\begin{aligned} \boldsymbol{\pi}^{(1)} &= P\boldsymbol{\pi}^{(0)} \\ \boldsymbol{\pi}^{(2)} &= P\boldsymbol{\pi}^{(1)} = P^2\boldsymbol{\pi}^{(0)} \\ &\ldots \end{aligned}$$

- The transition matrix after t-step converges

$$P^{(t)} = P^{(t-1)}P = P^{(t-2)}P^2 = \ldots$$

- Find the stationary distribution of P as $t \to \infty$ by solving the homogeneous linear system $\boldsymbol{\pi}(I - P) = 0$

PageRank algorithm (cont'd)

- The dominant eigenvector is the PageRank vector
- Random surfer:

$$M = \frac{1-c}{N}\mathbf{e}\mathbf{e}^\top + cP$$

  where c is a damping factor to account for whether a surfer follows a link or not (empirically set to 0.85 by Page and Brim)
- The PR values are the entries of the dominant (i.e., first) eigenvector of the modified transition matrix M

$$\boldsymbol{\pi} = M\boldsymbol{\pi} = \frac{1-c}{N}\mathbf{e}\mathbf{e}^\top\boldsymbol{\pi} + cP\boldsymbol{\pi} = \frac{1-c}{N}\mathbf{e} + cP\boldsymbol{\pi}$$

- The world's largest matrix computation!
- Solved by power iteration
- See "An eigenvector based ranking approach for hypertext" [Page and Brim SIGIR 98]

Locally linear embedding (LLE)

- Compute the low dimensional representation that most faithfully preserves the local liner structure of nearby data points [Roweis and Saul Science 00]
  1. Find the neighbors of each data point
  2. Find the best local linear reconstruction

$$\mathcal{E}_W = \sum_i \left\| \mathbf{x}_i - \sum_j W_{ij} \mathbf{x}_j \right\|^2$$

subject to $\sum_j W_{ij} = 1$

  3. Preserving the structure by minimizing

$$\mathcal{E}_\phi = \sum_i \left\| \phi_i - \sum_j W_{ij} \phi_j \right\|^2$$

subject to two constraints: (1) $\sum_i \phi_i = 0$, and (2) the outputs have unit covariance matrix, $\frac{1}{n} \sum_i \phi_i \phi_i^\top = I$

Maximum variance unfolding

- Find a low dimensional representation that most faithfully preserves the *distance* and *angles* between *nearby* input data points [Weinberger and Saul CVPR 04]
  1. First find k-nearest neighbors of each input data point. Denote $\eta_{ij} = 1$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are neighbors
  2. The constraints to preserve distances and angles between k nearest neighbors are
     $$\|\phi_i - \phi_j\|^2 = \|\mathbf{x}_i - \mathbf{x}_j\|^2$$
     for all $\eta_{ij} = 1$, $\mathbf{x} \in \mathbb{R}^m$ and $\phi \in \mathbb{R}^q$
     To eliminate a translational degree of freedom
     $$\sum_i \phi_i = 0, \ \phi_i \in R^q$$
  3. Unfold the input data points by maximizing the variance of the outputs
     $$\text{var}(\phi) = \sum_i \|\phi_i\|^2$$
- The optimization problem is formulated as a semi-definite programming problem

Maximum variance unfolding (cont'd)

- Solving

$$\begin{aligned} &\max \textstyle\sum_{ij} \|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\|^2 \\ &\text{subject to } \textstyle\sum_i \boldsymbol{\phi}_i = 0 \\ &\|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\|^2 = D_{ij} \quad \text{for all} \quad (i,j) \quad \text{whose} \quad \boldsymbol{\eta}_{ij} = 1 \end{aligned}$$

- The above optimization problem is not convex as it involves maximizing a quadratic function with quadratic equality constraints
- Reformulate the problem to a convex one
- Let $K_{ij} = \boldsymbol{\phi}_i \cdot \boldsymbol{\phi}_j$ denote the Gram matrix of the outputs, the semidefinite program is

$$\begin{aligned} &\max \ \operatorname{tr}(K) \\ &\text{subject to } K \succeq 0 \\ &\textstyle\sum_i \sum_j K_{ij} = 0 \\ &K_{ii} - 2K_{ij} + K_{jj} = \|\mathbf{x}_i - \mathbf{x}_j\|^2 \quad \text{for all} \quad (i,j) \quad \text{whose} \quad \boldsymbol{\eta}_{ij} = 1 \end{aligned}$$

Convex optimization

- Second Order Cone Program (SOCP): a linear program that is minimized over the intersection of an affine set and the product of second order (quadratic) cones

$$\begin{aligned} &\min \ f^\top \mathbf{x} \\ &\text{subject to } \|A_i\mathbf{x} + \mathbf{b}_i\| \leq \mathbf{c}_i^\top \mathbf{x} + \mathbf{d}_i, \ i = 1, \ldots, n \end{aligned}$$

- Semidefinite program (SDP): a linear program that is minimized over the intersection of positive semidefinite matrices
- Linear and positive semidefinite constraints are convex
- Efficient (polynomial time) algorithms exist to compute *global minimum*
- See "Convex optimization" by Boyd and Vandeberghe
- See "Convex optimization of graph Laplacian eigenvalues" [Boyd ICM 06]
- See "A duality view of spectral methods for dimensionality reduction" [Xiao et al. ICML 06]

Kernel PCA

- Conventional techniques capture only the second-order statistics of an image ensemble (e.g., PCA based on covariance matrix)
- A large part of the interesting image structure, however, is contained in the higher-order statistics
- Unfortunately, the estimation of these statistics involves a huge number of terms which makes their explicit computation for images infeasible in practice
- Kernel methods such as KPCA provide a computationally efficient way to compute *higher order statistics*
- Based on Gram matrix and exploits the duality of PCA and MDS

Various perspectives

- Isomap, graph Laplacian, LLE all use local neighborhood structure to construct a global embedding of manifold
- Can be interpreted using diffusion kernels
  - "Diffusion kernels on graphs and other discrete input spaces" [Kondor and Lafferty ICML 02]
  - "Diffusion kernels" [Kondor and Vert 04]
- Can be viewed as kernel PCA with different Gram matrices
  - "A kernel view of the dimensionality reduction of manifolds" [Ham et al ICML 04]
- From diffusion map and geometry
  - "Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators" [Nadler et al. NIPS 05]
  - "Diffusion maps and coarse-graining: A unifying framework for dimensionality reduction, graph partitioning, and data set parameterization" [Lafon and Lee PAMI 06]
- Constrained vs. unconstrained optimization

Approximate nearest neighbor problem

- Given a set P of points in $\mathbb{R}^m$, for any query $\mathbf{q}$, returns a point $\mathbf{p} \in P$ minimizing $\|\mathbf{p} - \mathbf{q}\|$
- Want to find an approximate algorithm to save space and/or time
- Locality sensitive hashing (LSH): Construct hash functions $g : \mathbb{R}^m \to U$ (U is a unit ball) such that for any points $\mathbf{p}$ and $\mathbf{q}$
  - If $\|\mathbf{p} - \mathbf{q}\| \leq r$, then $Pr[g(\mathbf{p}) = g(\mathbf{q})] > P_1$
  - If $\|\mathbf{p} - \mathbf{q}\| > cr$, then $Pr[g(\mathbf{p}) = g(\mathbf{q})] < P_2$
- Used in various applications (e.g., vision, multimedia/database retrieval)
- See "Approximate nearest neighbors" [Indyk and Motwani STOC 98], and "Two algorithms for nearest neighbor search in high dimensions" [Kleinberg STOC 97]

Multilinear algebra in data analysis

- If $\mathbf{u} \in R^l$, $\mathbf{v} \in R^n$, $\mathbf{w} \in R^n$, the tensor product

$$\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w} = \left| u_i v_j w_k \right|_{i,j,k=1}^{l,m,n}$$

- For example

$$\mathbf{b} \otimes \mathbf{a} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \otimes \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} = \begin{bmatrix} a_1 b_1 & a_2 b_1 & a_3 b_1 \\ a_1 b_2 & a_2 b_2 & a_3 b_2 \\ a_1 b_3 & a_2 b_3 & a_3 b_3 \\ a_1 b_4 & a_2 b_4 & a_3 b_4 \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \otimes \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} & a_{12} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \\ a_{21} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} & a_{22} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} b_{11} & a_{11} b_{12} & a_{12} b_{11} & a_{12} b_{12} \\ a_{11} b_{21} & a_{11} b_{22} & a_{12} b_{21} & a_{12} b_{22} \\ a_{21} b_{11} & a_{21} b_{12} & a_{22} b_{11} & a_{22} b_{12} \\ a_{21} b_{21} & a_{21} b_{22} & a_{22} b_{21} & a_{22} b_{22} \end{bmatrix}$$

Multilinear algebra in data analysis (cont'd)

- Matrix is rank 2 tensor
- Related to moments, cumulants, and are of particular importance in independence component analysis. Let $\mathbf{x}^{(n)} = \mathbf{x} \otimes \mathbf{x} \ldots \otimes \mathbf{x}$

$$\begin{aligned} M_2 &= E[\mathbf{x} \otimes \mathbf{x}^\top] = Cov(\mathbf{x}) \\ M_n &= E[\mathbf{x}^{(n-1)} \otimes \mathbf{x}^\top] \end{aligned}$$

- Computationally expensive even for 3-tensor space
- Needs low rank tensor approximation
- Applications: signal processing, information retrieval, computer vision, etc.

Summary

- Different perspectives to solve the same problems
- Manifold, geometry, graph, dynamics, random walk, low rank approximation, etc.
- Useful materials:
  - "Geometric methods for feature extraction and dimensional reduction" [Burges 05]
  - "Spectral methods for dimensionality reduction" [Saul NIPS 05 tutorial]
  - "Learning representation and behavior: Manifold and spectral methods for Markov decision processes and reinforcement learning" [Mahadevan and Maggioni ICML 06 tutorial]
  - "Spectral methods for dimensionality reduction" by Saul et al.
  - Hessian eigenmap by Donoho and Grimes
  - Diffusion maps by Lafon et al.