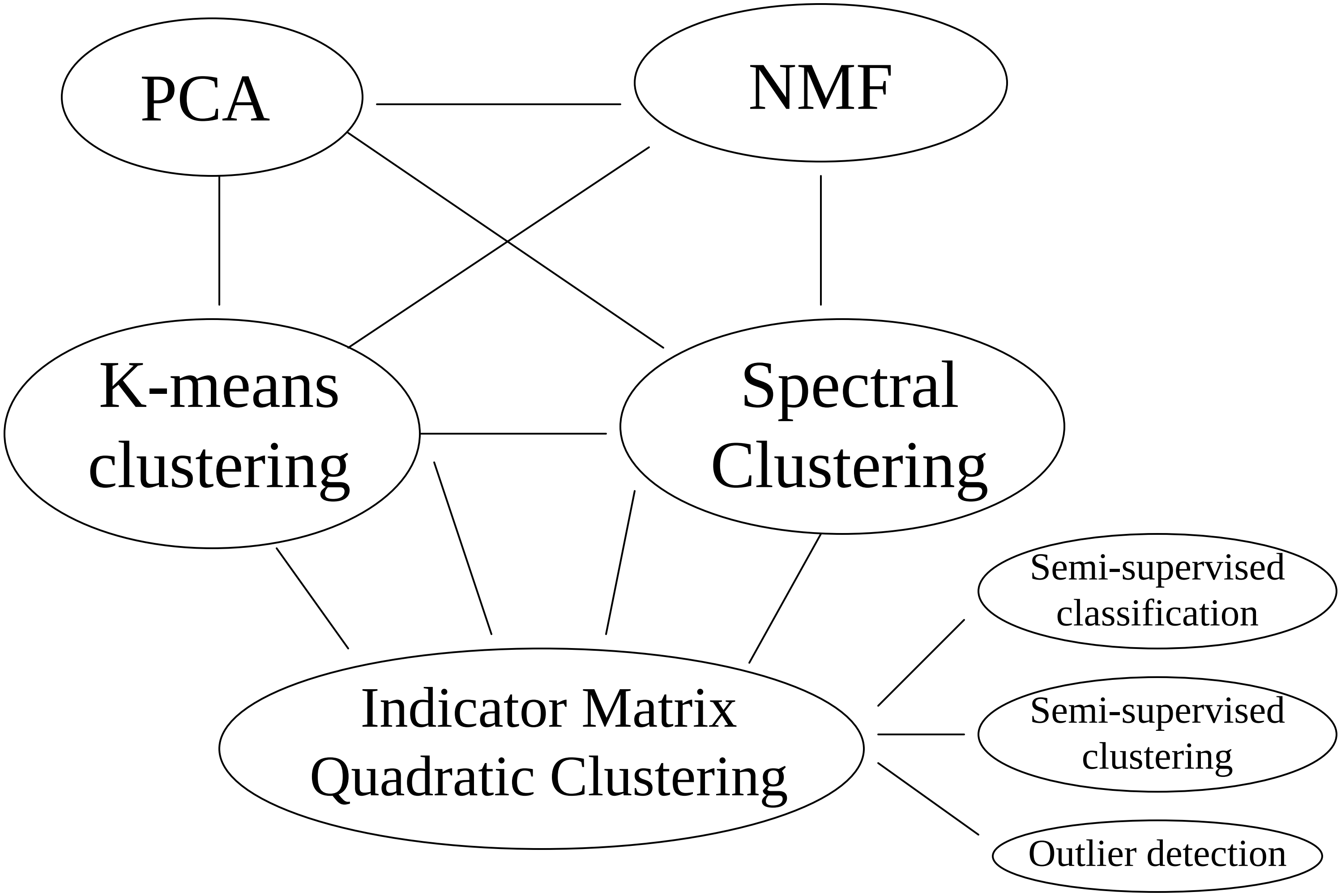
Principal Component Analysis and Matrix Factorizations for Learning

Chris Ding

Lawrence Berkeley National Laboratory

Supported by Office of Science, U.S. Dept. of Energy

Many unsupervised learning methods are closely related in a simple way


PCA
NMF
K-means clustering
Spectral Clustering
Indicator Matrix Quadratic Clustering
Semi-supervised classification
Semi-supervised clustering
Outlier detection

Part 1.A.
Principal Component Analysis (PCA) and Singular Value Decomposition (SVD)

- Widely used in large number of different fields
- Most widely known as PCA (multivariate statistics)
- SVD is the theoretical basis for PCA

Brief history

- PCA
  - Draw a plane closest to data points (Pearson, 1901)
  - Retain most variance (Hotelling, 1933)
- SVD
  - Low-rank approximation (Eckart-Young, 1936)
  - Practical application/Efficient Computation (Golub-Kahan, 1965)
- Many generalizations

PCA and SVD

Data: *n* points in *p*-dim: $X = (x_1, x_2, \cdots, x_n)$

Covariance $C = XX^T = \sum_{k=1}^{p} \lambda_k u_k u_k^T$

Gram (kernel) matrix $X^T X = \sum_{k=1}^{r} \lambda_k v_k v_k^T$

Principal directions: u_k

(Principal axis,subspace)

Principal components: v_k

(projection on the subspace)

Underlying basis: SVD $X = \sum_{k=1}^{p} \sigma_k u_k v_k^T = U \Sigma V^T$

Further Developments

SVD/PCA

- Principal Curves
- Independent Component Analysis
- Sparse SVD/PCA (many approaches)
- Mixture of Probabilistic PCA
- Generalization to exponential familty, max-margin
- Connection to K-means clustering

Kernel (inner-product)

- Kernel PCA

Methods of PCA Utilization

Principal components (uncorrelated random variables):

$$X = (x_1, x_2, \cdots, x_n)$$

$$u_k = u_k(1) \cdot X_1 + \cdots + u_k(d) \cdot X_d$$

Dimension reduction: $X = \sum_{k=1}^{p} \sigma_k u_k v_k^T = U \Sigma V^T$

Projection to low-dim subspace $\tilde{X} = U^T X \qquad U = (u_1, \cdots, u_k)$

Sphereing the data $\tilde{X} = C^{-1/2} X = U \Sigma^{-1} U^T X$

Transform data to N(0,1)

Applications of PCA/SVD

- Most popular in multivariate statistics
- Image processing, signal processing
- Physics: principal axis, diagonalization of 2nd tensor (mass)
- Climate: Empirical Orthogonal Functions (EOF)
- Kalman filter. $s^{(t+1)} = As^{(t)} + E,\ P^{(t+1)} = AP^{(t)}A^{T}$
- Reduced order analysis

Applications of PCA/SVD

- PCA/SVD is as widely as Fast Fourier Transforms
  - Both are spectral expansions
  - FFT is more on Partial Differential Equations
  - PCA/SVD is more on discrete (data) analysis
  - PCA/SVD surpass FFT as computational sciences further advance
- PCA/SVD
  - Select combination of variables
  - Dimension reduction
    - An image has 10^4 pixels. True dimension is 20 !

PCA is a Matrix Factorization (spectral/eigen decomposition)

Principal directions: $U = (u_1, u_2, \cdots, u_k)$

Principal components: $V = (v_1, v_2, \cdots, v_k)$

Covariance $C = XX^T = \sum_{k=1}^{p} \lambda_k u_k u_k^T = U \Lambda U^T$

Kernel matrix $X^T X = \sum_{k=1}^{r} \lambda_k v_k v_k^T = V \Lambda V^T$

Underlying basis: SVD $X = \sum_{k=1}^{p} \sigma_k u_k v_k^T = U \Sigma V^T$

From PCA to spectral clustering using generalized eigenvectors

Consider the kernel matrix: $W_{ij} = \left\langle \phi(x_i), \phi(x_j) \right\rangle$

In Kernel PCA we compute eigenvector: $Wv = \lambda v$

Generalized Eigenvector: $Wq = \lambda Dq$

$D = diag(d_1, \cdots, d_n) \quad d_i = \sum_j w_{ij}$

This leads to Spectral Clustering !

Scale PCA ⇒ Spectral Clustering

PCA: $$W = \sum_k v_k \lambda_k v_k^T$$

Scaled PCA: $$W = D^{\frac{1}{2}} \tilde{W} D^{\frac{1}{2}} = D \sum_{k=1} q_k \lambda_k q_k^T D$$

$$\tilde{W} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}, \quad \tilde{w}_{ij} = w_{ij} / (d_i d_j)^{1/2}$$

$q_k = D^{-\frac{1}{2}} v_k$ scaled principal component

Scaled PCA on a Rectangle Matrix
$\Rightarrow$ Correspondence Analysis

Re-scaling: $\tilde{P} = D_r^{-\frac{1}{2}} P D_c^{-\frac{1}{2}}, \; \tilde{p}_{ij} = p_{ij} / (p_{i.} p_{j.})^{1/2}$

Apply SVD on $\tilde{P}$ Subtract trivial component

$$P - rc^T / p_{..} = D_r \sum_{k=1} f_k \lambda_k g_k^T D_c \qquad r = (p_{1.}, \cdots, p_{n.})^T$$

$$f_k = D_r^{-\frac{1}{2}} u_k, \; g_k = D_c^{-\frac{1}{2}} v_k \qquad c = (p_{.1}, \cdots, p_{.n})^T$$

are scaled row and column principal component (standard coordinates in CA)

(Zha, et al, CIKM 2001, Ding et al, PKDD2002)

Nonnegative Matrix Factorization

Data Matrix: *n* points in *p*-dim:

$$X = (x_1, x_2, \cdots, x_n)$$

x_i is an image, document, webpage, etc

Decomposition
(low-rank approximation)

$$X \approx FG^T$$

Nonnegative Matrices

$$X_{ij} \ge 0, \;\; F_{ij} \ge 0, \;\; G_{ij} \ge 0$$

$$F = (f_1, f_2, \cdots, f_k) \qquad G = (g_1, g_2, \cdots, g_k)$$

Solving NMF with multiplicative updating

$$J = \| X - FG^T \|^2, F \geq 0, G \geq 0$$

Fix *F*, solve for *G*; Fix *G*, solve for *F*

Lee & Seung (2000) propose

$$F_{ik} \leftarrow F_{ik} \frac{(XG)_{ik}}{(FG^T G)_{ik}} \qquad G_{jk} \leftarrow G_{jk} \frac{(X^T F)_{jk}}{(GF^T F)_{jk}}$$

Matrix Factorization Summary

	Symmetric (kernel matrix, graph)	Rectangle Matrix (contigency table, bipartite graph)
PCA:	$W = V\Lambda V^T$	$X = U\Sigma V^T$
Scaled PCA:	$W = D^{\frac{1}{2}}\tilde{W}D^{\frac{1}{2}} = D\,Q\Lambda Q^T D$	$X = D_r^{\frac{1}{2}}\tilde{X}\,D_c^{\frac{1}{2}} = D_r F\Lambda G^T D_c$
NMF:	$W \approx QQ^T$	$X \approx FG^T$

Indicator Matrix Quadratic Clustering

Unsigned Cluster indicator Matrix $H=(h_1, \cdots, h_K)$

Kernel K-means clustering:

$$\max_H \mathrm{Tr}(H^T W H), \quad s.t.\ H^T H = I, H \geq 0$$

K-means: $W = X^T X;$ Kernel K-means $W = (<\phi(x_i), \phi(x_j)>)$

Spectral clustering (normalized cut)

$$\max_H \mathrm{Tr}(H^T W H), \quad s.t.\ H^T D H = I, H \geq 0$$

Difference between the two is the orthogonality of *H*

Indicator Matrix Quadratic Clustering

Additional features:

Semi-suerpvised classification: $\max_H \mathrm{Tr}(H^T WH + C^T H)$

Semi-supervised clustering: (A) must-link and (B) cannot-link constraints

$$\max_H \mathrm{Tr}(H^T WH + \alpha H^T AH - \beta H^T BH)$$

Outlier Detection: $\max_H \mathrm{Tr}(H^T WH)$ allowing zero rows in *H*

Nonnegative Lagrangian Relaxation:

$$H_{ik} \leftarrow H_{ik} \sqrt{\frac{(WH)_{ik} + C_{ik}/2}{(H\alpha)_{ik}}}, \quad \alpha = H^T WH + H^T C.$$

Tutorial Outline

- PCA
  - Recent developments on PCA/SVD
  - Equivalence to K-means clustering
- Scaled PCA
  - Laplacian matrix
  - Spectral clustering
  - Spectral ordering
- Nonnegative Matrix Factorization
  - Equivalence to K-means clustering
  - Holistic vs. Parts-based
- Indicator Matrix Quadratic Clustering
  - Use Nonnegative Lagrangian Relaxtion
  - Includes
    - K-means and Spectral Clustering
    - semi-supervised classification
    - Semi-supervised clustering
    - Outlier detection

Part 1.B.
Recent Developments on PCA and SVD

Principal Curves

Independent Component Analysis

Kernel PCA

Mixture of PCA (probabilistic PCA)

Sparse PCA/SVD

- Semi-discrete, truncation, L1 constraint, Direct sparsification

Column Partitioned Matrix Factorizations

2D-PCA/SVD

Equivalence to K-means clustering

PCA and SVD

Data Matrix: $X = (x_1, x_2, \cdots, x_n)$

Covariance $C = XX^T = \sum_{k=1}^{p} \lambda_k u_k u_k^T$

Gram (kernel) matrix $X^T X = \sum_{k=1}^{r} \lambda_k v_k v_k^T$

Principal directions: u_k
(Principal axis,subspace)

Principal components: v_k
(projection on the subspace)

Underlying basis: SVD $X = \sum_{k=1}^{p} \sigma_k u_k v_k^T$

Kernel PCA

$$x_i \rightarrow \phi(x_i)$$

Kernel $$K_{ij} = \left\langle \phi(x_i), \phi(x_j) \right\rangle$$

PCA Component v

Feature extraction $$\left\langle v, \phi(x) \right\rangle = \sum_i v_i \left\langle \phi(x_i), \phi(x) \right\rangle$$

Indefinite Kernels

Generalization to graphs with nonnegative weights

(Scholkopf, Smola, Muller, 1996)

Mixture of PCA

- Data has local structures.
  - Global PCA on all data is not useful
- Clustering PCA (Hinton et al):
  - Using clustering to cluster data into clusters
  - Perform PCA in each cluster
  - No explicit generative model
- Probabilistic PCA (Tipping & Bishop)
  - Latent variables
  - Generative model (Gaussian)
  - Mixture of Gaussians ⇒ mixture of PCA
  - Adding Markov dynamics for latent variables (Linear Gaussian Models)

Probabilistic PCA
Linear Gaussian Model

Latent variables $S = (s_1, \cdots, s_n)$

$$x_i = Ws_i + \mu + \varepsilon, \quad \varepsilon \sim N(0, \sigma_\varepsilon^2 I)$$

Gaussian prior $P(s) \sim N(s_0, \sigma_s^2 I)$

$$x \sim N(Ws_0, \sigma_\varepsilon^2 I + \sigma_s WW^T)$$

Linear Gaussian Model

$$s_{i+1} = As_i + \eta, \qquad x_i = Ws_i + \varepsilon,$$

(Tipping & Bishop, 1995; Roweis & Ghahramani, 1999)

Sparse PCA

- Compute a factorization $X \approx UV^T$
  - *U* or *V* is sparse or both are sparse
- Why sparse?
  - Variable selection (sparse *U*)
  - When $n >> d$
  - Storage saving
  - Other new reasons?
- L_1 and L_2 constraints

Sparse PCA: Truncation and Discretization

$$X \approx U \Sigma V^T$$

$$U = (u_1 \cdots u_k) \quad V = (v_1 \cdots v_k)$$

- Sparsified SVD
  - Compute $\{u_k, v_k\}$ one at a time, truncate those entries below a threshold.
  - Recursively compute all pairs using deflation.
  - (Zhang, Zha, Simon, 2002)

$$X \leftarrow X - \sigma u v^T$$

- Semi-discrete decomposition
  - *U*, *V* only contains {-1, 0, 1}
  - Iterative algorithm to compute U,V using deflation
  - (Kolda & O'leary, 1999)

Sparse PCA: L_1 constraint

- LASSO (Tibshirani, 1996)

$$\min \| y - X^T \beta \|^2, \quad \| \beta \|_1 \le t$$

- SCoTLASS (Joliffe & Uddin, 2003)

$$\max u^T (XX^T) u^T, \quad \| u \|_1 \le t, \quad u^T u_h = 0$$

- Least Angle Regression (Efron, et al 2004)
- Sparse PCA (Zou, Hastie, Tibshirani,2004)

$$\min_{\alpha,\beta} \sum_{i=1}^{n} \| x_i - \alpha\beta^T x_i \|^2 + \lambda \sum_{j=1}^{k} \| \beta_j \|^2 + \sum_{j=1}^{k} \lambda_{1,j} \| \beta_j \|_1, \alpha^T \alpha = I$$

$$v_j = \beta_j / \| \beta_j \|$$

Sparse PCA: Direct Sparsification

- Sparse SVD with explicit sparsification

$$\min_{u,v} \| X - udv^T \|_F + \mathrm{nnz}(u) + \mathrm{nnz}(v)$$

  - rank-one approximation (Zhang, Zha, Simon 2003)
  - Minimize a bound
  - deflation

- Direct sparse PCA, on covariance matrix S

$$u = \max u^T S u = \max \mathrm{Tr}(Suu^T) = \max \mathrm{Tr}(SU)$$

$$s.t.\ \ \mathrm{Tr}(U) = 1,\ \mathrm{nnz}(U) \le k^2,\ U \succeq 0,\ \mathrm{rank}(U) = 1$$

(D'Aspremont, Gharoui, Jordan,Lancriet, 2004)

Sparse PCA Summary

- Many different approaches
  - Truncation, discretization
  - L1 Constraint
  - Direct sparsification
  - Other approaches
- Sparse Matrix factorization in general
  - L_1 constraint
- Many questions
  - Orthogonality
  - Unique solution, global solution

PCA: Further Generalizations

- Generalization to Exponential Family
  - (Collins, Dasgupta, Schapire, 2001)
- Maximum Margin Factorization (Srebro, Rennie, Jaakkola, 2004)
  - Collaborative filtering
  - Input Y is binary
  - Hard margin $Y_{ia}X_{ia} \geq 1, \forall ia \in S$
  - Soft margin

$$\min \| X \|_{\Sigma} + c \sum_{ia \in S} \max(0, 1 - Y_{ia} X_{ia})$$

$$X = UV^T, \quad \| X \| = \tfrac{1}{2}(\| U \|_{Fro}^2 + \| V \|_{Fro}^2)$$

Column Partitioned Matrix Factorizations

$$X = (x_1, \cdots x_n) = (\overbrace{x_1 \cdots x_{n_1}}^{n_1}, \overbrace{x_{n_1+1} \cdots x_{n_2}}^{n_2}, \cdots, \overbrace{x_{n_{k-1}+1} \cdots x_n}^{n_k}) \qquad n_1 + \cdots + n_k = n$$

- Column Partitioned Data Matrix

(Zhang & Zha, 2001)

- Partitions are generate by clustering

(Dhillon & Modha, 2001)

(Park, Jeon & Rosen, 2003)

- Centroid matrix $U = (u_1 \cdots u_k)$
  - u_k is centroid
  - Fix U, compute V $\min \| X - UV^T \|_F^2$ $V = X^T U (U^T U)^{-1}$
- Represent each partition by a SVD.
  - Pick leading Us to form U
  - Fix U, compute V

$$U = (U_1, \cdots U_\ell) = (\overbrace{u_1^{(1)} \cdots u_{k_1}^{(1)}}^{k_1}, \ \cdots, \overbrace{u_1^{(\ell)} \cdots u_{k_\ell}^{(\ell)}}^{k_\ell})$$

- Several other variations

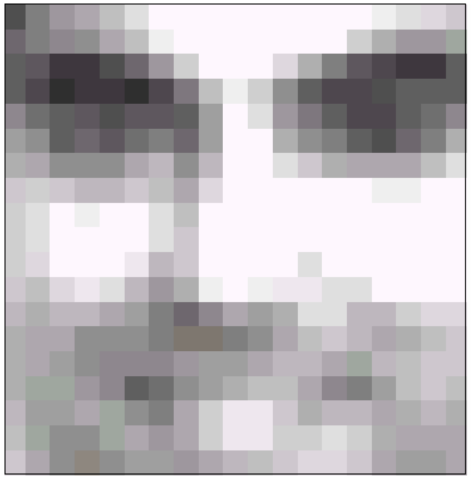
(Castelli, Thomasian & Li 2003)

(Zeimpekis & Gallopoulos, 2004)

Two-dimensional SVD

- Large number of data objects are 2-D: images, maps
- Standard method:
  - convert (re-order) each image as a 1D vector
  - collect all 1D vectors into a single (big) matrix
  - apply SVD on the big matrix
- 2D-SVD is developed for 2D objects
  - Extension of standard SVD
  - Keeping the 2D characteristics
  - Improves quality of low-dimensional approximation
  - Reduces computation, storage

Linearize a 2D object into 1D object



$$\longrightarrow \begin{bmatrix} 0.0 \\ 0.5 \\ 0.7 \\ 1.0 \\ \vdots \\ 0.8 \\ 0.2 \\ 0.0 \end{bmatrix}$$

Pixel vector

SVD and 2D-SVD

SVD $X = (x_1, x_2, \cdots, x_n)$

Eigenvectors of XX^T and $X^T X$

$$X = U \Sigma V^T \qquad \Sigma = U^T X V$$

2D-SVD $\{A\} = \{A_1, A_2, \cdots, A_n\}$

Eigenvectors of

$$F = \sum_i (A_i - \bar{A})(A_i - \bar{A})^T$$ row-row covariance

$$G = \sum_i (A_i - \bar{A})^T (A_i - \bar{A})$$ column-column cov

$$A_i = U M_i V^T \qquad M_i = U^T A_i V$$

2D-SVD

$$\{A\} = \{A_1, A_2, \cdots, A_n\}$$ assume $\overline{A} = 0$

row-row cov: $F = \sum_i A_i A_i^T = \sum \lambda_k u_k u_k^T$

col-col cov: $G = \sum_i A_i^T A_i = \sum_{k=1} \zeta_k u_k u_k^T$

Bilinear $U = (u_1, u_2, \cdots, u_k)$

subspace $V = (v_1, v_2, \cdots, v_k)$ $M_i = U^T A_i V$

$$A_i = U M_i V^T, i = 1, \cdots, n$$

$$A_i \in \Re^{r \times c}, U \in \Re^{r \times k}, V \in \Re^{c \times k}, M_i \in \Re^{k \times k}$$

2D-SVD Error Analysis

$$\text{SVD:} \quad \min \| X - U\Sigma V^T \|^2 = \sum_{i=k+1}^{p} \sigma_i^2$$

$$A_i \approx L M_i R^T, A_i \in R^{r\times c}, L \in R^{r\times k}, R \in R^{c\times k}, M_i \in R^{k\times k}$$

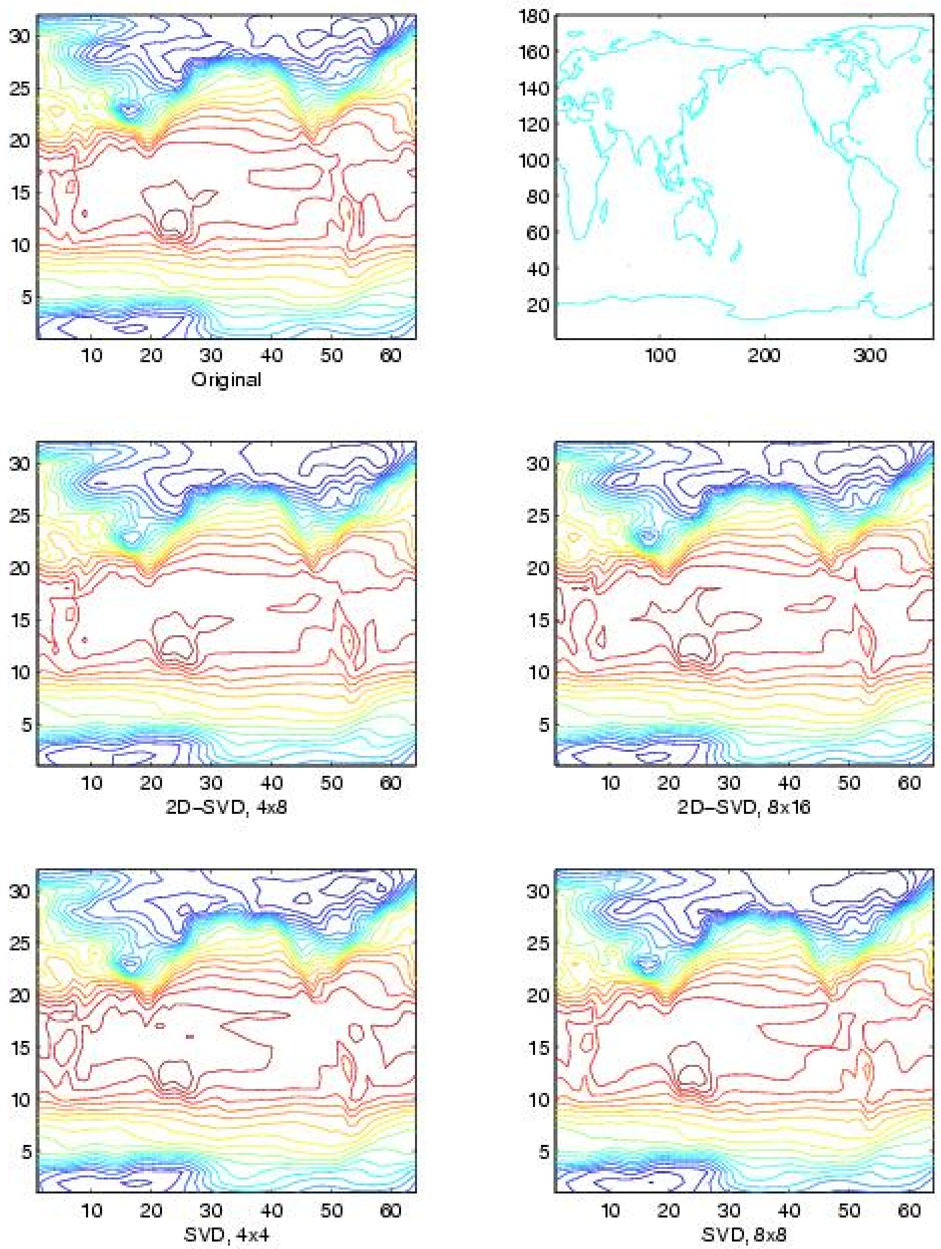
$$\min J_1 = \sum_{i=1}^{n} \| A_i - L M_i \|^2 = \sum_{j=k+1}^{c} \zeta_j$$

$$\min J_2 = \sum_{i=1}^{n} \| A_i - M_i R^T \|^2 = \sum_{j=k+1}^{r} \lambda_j$$

$$\min J_3 = \sum_{i=1}^{n} \| A_i - L M_i R^T \|^2 \cong \sum_{j=k+1}^{r} \lambda_j + \sum_{j=k+1}^{c} \zeta_j$$

$$\min J_4 = \sum_{i=1}^{n} \| A_i - L M_i L^T \|^2 \cong 2 \sum_{j=k+1}^{r} \lambda_j$$

Temperature maps (January over 100 years)



Original

2D–SVD, 4x8

2D–SVD, 8x16

SVD, 4x4

SVD, 8x8

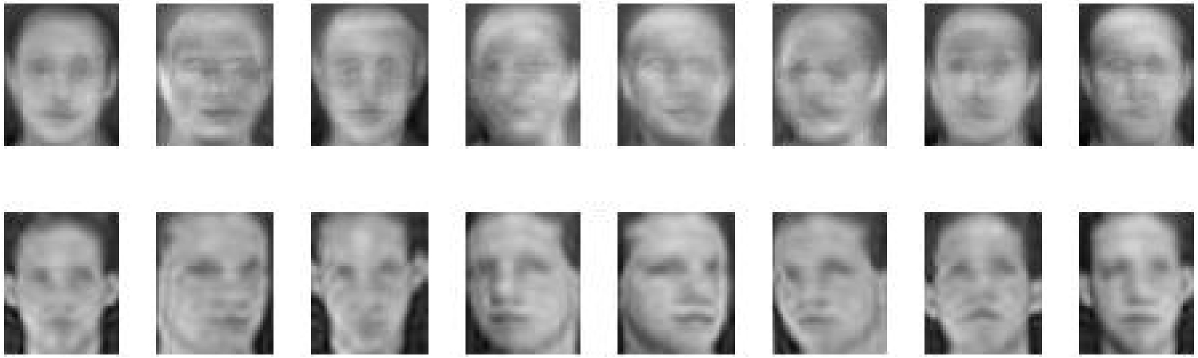
Reconstruction Errors

SVD/2DSVD=1.1

Storages

SVD/2DSVD=8

Reconstructed image



SVD

2dSVD

SVD (K=15), storage 160560

2DSVD (K=15), storage 93060

2D-SVD Summary

- 2DSVD is extension of standard SVD
- Provides optimal solution for 4 representations for 2D images/maps
- Substantial improvements in storage, computation, quality of reconstruction
- Capture 2D characteristics

Part 1.C.
K-means Clustering ⇔ Principal Component Analysis

(Equivalence between PCA and K-means)

K-means clustering

- Also called “isodata”, “vector quantization”
- Developed in 1960’s (Lloyd, MacQueen, Hatigan, etc)
- Computationally Efficient (order-mN)
- Widely used in practice
  - Benchmark to evaluate other algorithms

Given n points in m-dim: $X = (x_1, x_2, \cdots, x_n)^T$

K-means objective $\min J_K = \sum_{k=1}^{K} \sum_{i \in C_k} \| x_i - c_k \|^2$

PCA is equivalent to *K*-means

Continuous optimal solution for cluster indicators in ***K***-means clustering are given by principal components.

Subspace spanned by *K* cluster centroids is given by PCA subspace.

2-way *K* -means Clustering

Cluster membership indicator:

$$q(i) = \begin{cases} +\sqrt{n_2 / n_1 n} & \text{if } \; i \in C_1 \\ -\sqrt{n_1 / n_2 n} & \text{if } \; i \in C_2 \end{cases}$$

$$J_K = n\langle x^2 \rangle - J_D, \quad J_D = \frac{n_1 n_2}{n}\left[2\frac{d(C_1,C_2)}{n_1 n_2} - \frac{d(C_1,C_1)}{n_1^2} - \frac{d(C_2,C_2)}{n_2^2} \right]$$

Define distance matrix: $D = (d_{ij}), \; d_{ij} = |x_i - x_j|^2$

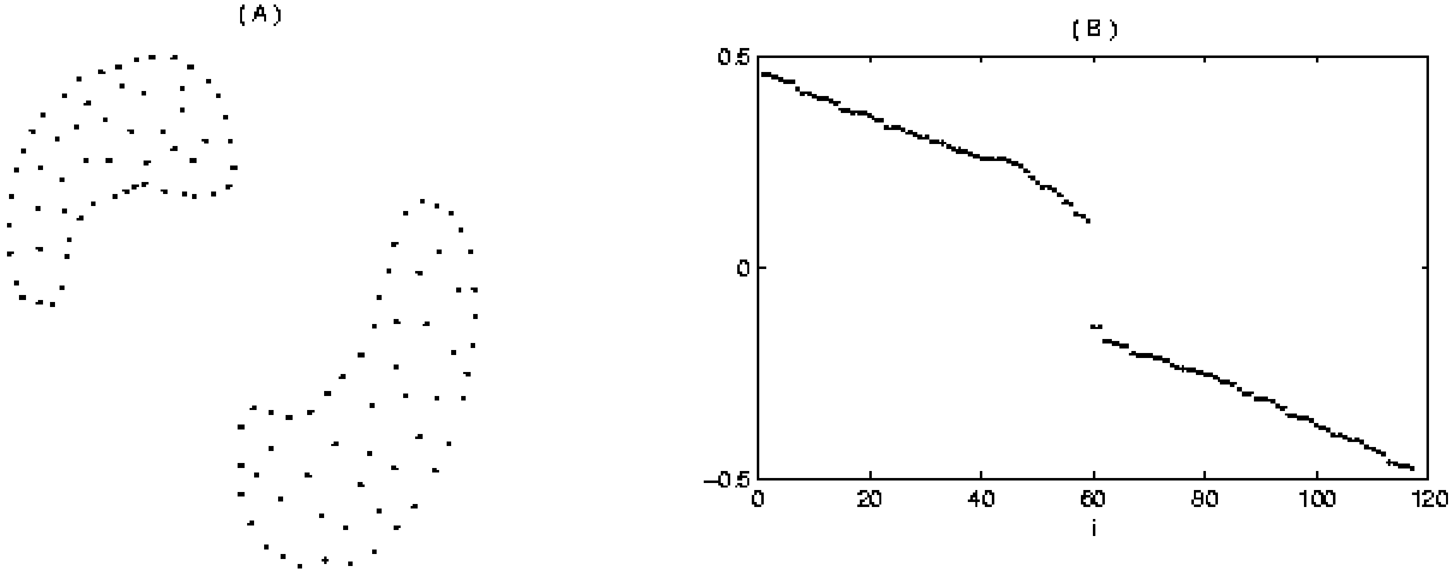
$$J_D = -q^T D q = -q^T \widetilde{D} q = 2q^T (X^T X) q = 2q^T K q \qquad \widetilde{D} = K$$

$\min J_K \Rightarrow \max J_D$

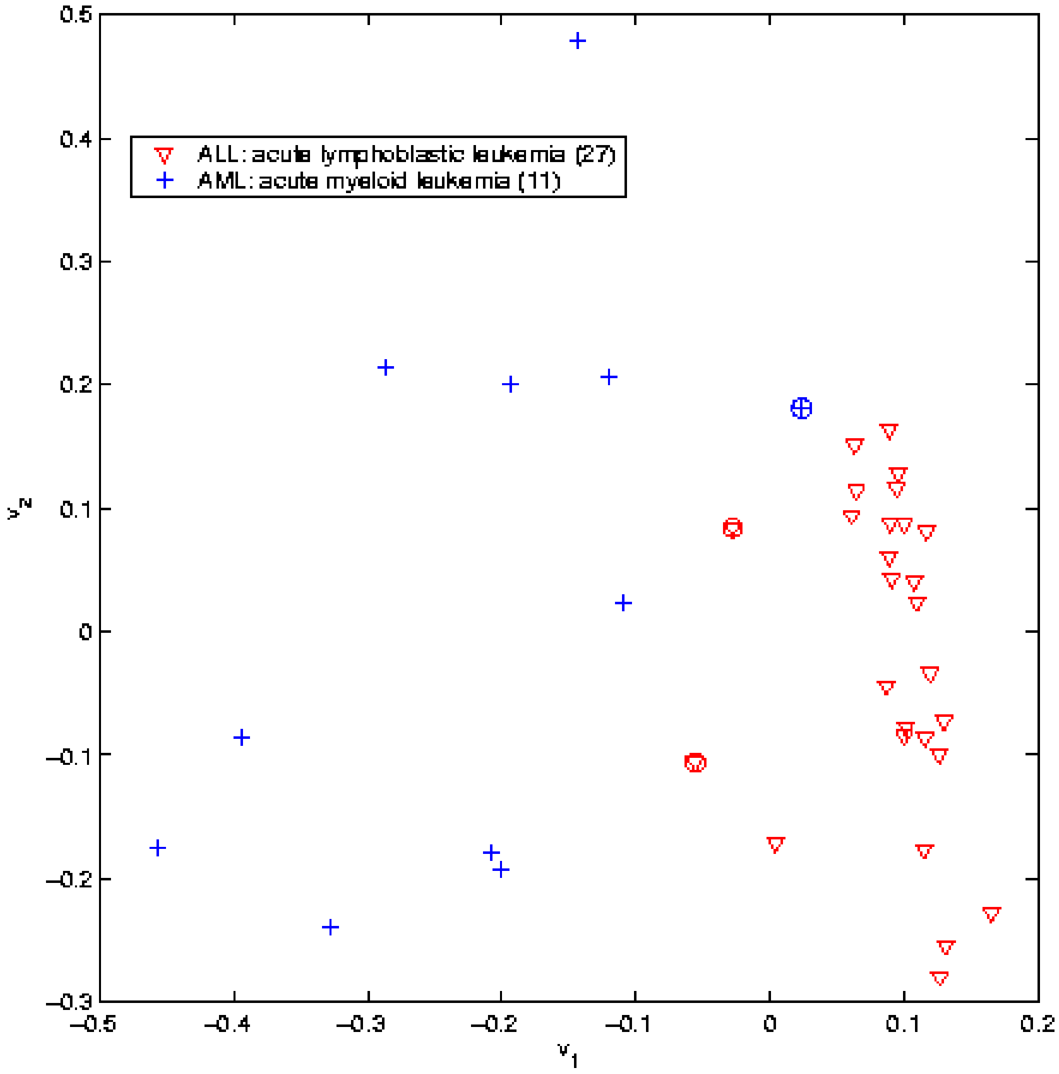
Solution is principal eigenvector v_1 of *K*

Clusters C_1, C_2 are determined by: $C_1 = \{i \,|\, v_1(i) < 0\}, C_2 = \{i \,|\, v_1(i) \geq 0\}$

A simple illustration



DNA Gene Expression File for Leukemia



Using v_1 , tissue samples separated into 2 clusters, 3 errors

Do one more K-means, reduce to 1 error

Multi-way K-means Clustering

Unsigned Cluster membership indicators $h_1, \cdots, h_K$:

$$\begin{array}{c} \begin{array}{ccc} C_1 & C_2 & C_3 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} = (h_1, h_2, h_3)$$

Multi-way K-means Clustering

$$J_K = \sum_i x_i^2 - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,j\in C_k} x_i^T x_j = \sum_i x_i^2 - \sum_{k=1}^{K} h_k^T X^T X h_k$$

(Unsigned) Cluster indicators $H=(h_1, \cdots, h_K)$

$$J_K = \sum_i x_i^2 - \mathrm{Tr}(H_k^T X^T X H_k)$$

Regularized Relaxation   Redundancy: $\sum_{k=1}^{K} n_k^{1/2} h_k = e$

Transform $h_1, \cdots, h_K$ to q_1 - q_k via orthogonal matrix T

$$(q_1,\dots,q_k) = (h_1,\cdots,h_k)T \qquad Q_k = H_k T \qquad q_1 = e/n^{1/2}$$

Multi-way K-means Clustering

$$\max \mathrm{Tr}[Q_{k-1}^T (X^T X) Q_{k-1}] \qquad Q_{k-1} = (q_2, \ldots, q_k)$$

Optimal solutions of $q_2 \cdots q_\mathrm{k}$ are given by principal components $v_2 \cdots v_\mathrm{k}$.

J_K is bounded below by total variance minus sum of K eigenvalues of covariance:

$$n\overline{x^2} - \sum_{k=1}^{K-1} \lambda_k < \min J_K < n\overline{x^2}$$

Consistency: 2-way and K-way approaches

Orthogonal Transform: $T = \begin{pmatrix} \sqrt{n_2/n} & -\sqrt{n_1/n} \\ \sqrt{n_1/n} & \sqrt{n_2/n} \end{pmatrix}$

T transforms (h_1, h_2) to (q_1,q_2):

$$h_1 = (1\cdots1,0\cdots0)^T, \quad h_2 = (0\cdots0,1\cdots1)^T \qquad a = \sqrt{\frac{n_2}{n_1 n}}$$

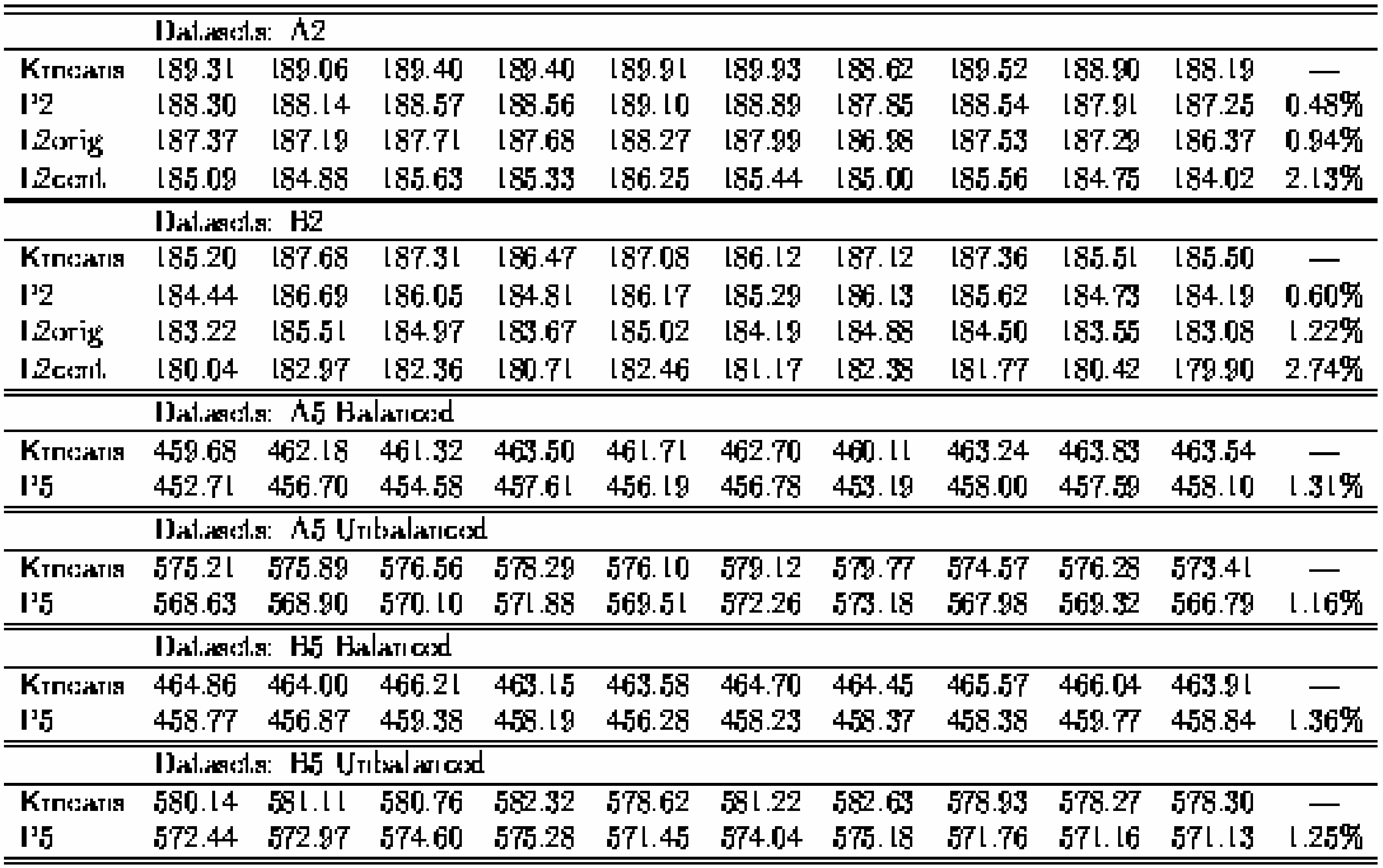
$$q_1 = (1\cdots1)^T, \quad q_2 = (a,\cdots,a,-b,\cdots,-b)^T \qquad b = \sqrt{\frac{n_1}{n_2 n}}$$

Recover the original 2-way cluster indicator

Test of Lower bounds of K-means clustering

$$\frac{|J_{opt} - J_{LB}|}{J_{opt}}$$

Kmeans objective function values and theoretical bounds for 6 datasets.



	Datasets: A2										
Kmeans	189.31	189.06	189.40	189.40	189.91	189.93	188.62	189.52	188.90	188.19	—
P2	188.30	188.14	188.57	188.56	189.10	188.89	187.85	188.54	187.91	187.25	0.48%
L2orig	187.37	187.19	187.71	187.68	188.27	187.99	186.98	187.53	187.29	186.37	0.94%
L2cent	185.09	184.88	185.63	185.33	186.25	185.44	185.00	185.56	184.75	184.02	2.13%
	Datasets: B2										
Kmeans	185.20	187.68	187.31	186.47	187.08	186.12	187.12	187.36	185.51	185.50	—
P2	184.44	186.69	186.05	184.81	186.17	185.29	186.13	185.62	184.73	184.19	0.60%
L2orig	183.22	185.51	184.97	183.67	185.02	184.19	184.88	184.50	183.55	183.08	1.22%
L2cent	180.04	182.97	182.36	180.71	182.46	181.17	182.38	181.77	180.42	179.90	2.74%
	Datasets: A5 Balanced										
Kmeans	459.68	462.18	461.32	463.50	461.71	462.70	460.11	463.24	463.83	463.54	—
P5	452.71	456.70	454.58	457.61	456.19	456.78	453.19	458.00	457.59	458.10	1.31%
	Datasets: A5 Unbalanced										
Kmeans	575.21	575.89	576.56	578.29	576.10	579.12	579.77	574.57	576.28	573.41	—
P5	568.63	568.90	570.10	571.88	569.51	572.26	573.18	567.98	569.32	566.79	1.16%
	Datasets: B5 Balanced										
Kmeans	464.86	464.00	466.21	463.15	463.58	464.70	464.45	465.57	466.04	463.91	—
P5	458.77	456.87	459.38	458.19	456.28	458.23	458.37	458.38	459.77	458.84	1.36%
	Datasets: B5 Unbalanced										
Kmeans	580.14	581.11	580.76	582.32	578.62	581.22	582.63	578.93	578.27	578.30	—
P5	572.44	572.97	574.60	575.28	571.45	574.04	575.18	571.76	571.16	571.13	1.25%

Lower bound is within 0.6-1.5% of the optimal value

Cluster Subspace (spanned by K centroids) = PCA Subspace

Given a data point x,

$$P = \sum_k c_k c_k^T$$ project x into the cluster subspace

Centroid is given by $c_k = \sum_k h_k(i) x_i = X h_k$

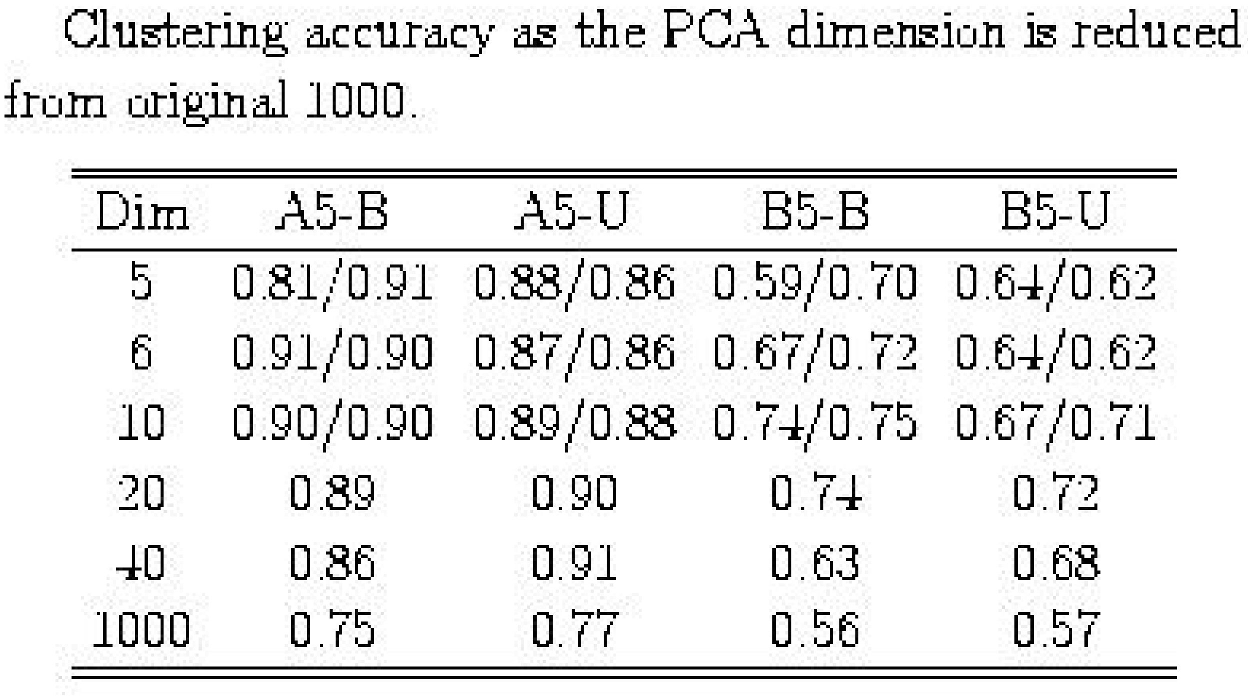
$$P = \sum_k c_k c_k^T = X \sum_k h_k h_k^T X^T = X \sum_k v_k v_k^T X^T = \sum_k \lambda_k u_k u_k^T$$

$$P_{K-means} = \sum_k \lambda_k u_k u_k^T \quad \Leftrightarrow \quad \sum_k u_k u_k^T \equiv P_{PCA}$$

PCA automatically project into cluster subspace

PCA is unsupervised version of LDA

Effectiveness of PCA Dimension Reduction



Clustering accuracy as the PCA dimension is reduced from original 1000.

Dim	A5-B	A5-U	B5-B	B5-U
5	0.81/0.91	0.88/0.86	0.59/0.70	0.64/0.62
6	0.91/0.90	0.87/0.86	0.67/0.72	0.64/0.62
10	0.90/0.90	0.89/0.88	0.74/0.75	0.67/0.71
20	0.89	0.90	0.74	0.72
40	0.86	0.91	0.63	0.68
1000	0.75	0.77	0.56	0.57

Kernel ***K***-means Clustering

Kernal ***K***-means objective: $x_i \to \phi(x_i)$

$$\min J_K^\phi = \sum_{k=1}^{K} \sum_{i\in C_k} \| \phi(x_i) - \bar{\phi}(c_k) \|^2$$

$$= \sum_i |\phi(x_i)|^2 - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,j\in C_k} \phi(x_i)^T \phi(x_j)$$

Kernal ***K***-means $\max J_K^\phi = \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,j\in C_k} \left\langle \phi(x_i), \phi(x_j) \right\rangle$

Kernel *K*-means clustering is equivalent to Kernal PCA

Continuous optimal solution for cluster indicators are given by Kernal PCA components

Subspace spanned by *K* cluster centroids are given by Kernal PCA principal subspace