

Subspace Clustering of High Dimensional Data

.....
Dimitris Papadopoulos Dimitrios Gunopoulos

University of California, Riverside

Carlotta Domeniconi
George Mason University

Sheng Ma
IBM T J Watson Research Center

Introduction

Clustering suffers from the curse of dimensionality, and similarity functions that use all input features with equal relevance may not be effective.

We introduce an algorithm that

- **discovers clusters** in subspaces spanned by different combinations of dimensions via local weightings of features.
- **associates to each cluster a weight vector**, whose values capture the relevance of features within the corresponding cluster.

This approach avoids the risk of loss of information encountered in global dimensionality reduction techniques, and does not assume any data distribution model.

Clustering

- Fundamental to all clustering techniques is the choice of distance measure between data points;

$$D(x_i, x_j) = \sum_{k=1}^q (x_{ik} - x_{jk})^2 \quad \text{Squared Euclidean distance}$$

- Assumption: All features are **equally important**;
- Such approaches fail in high dimensional spaces

Example

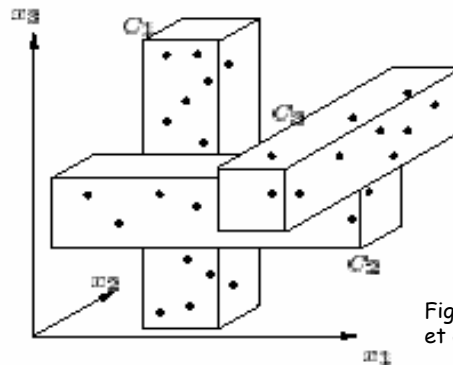


Fig. from C. Procopiuc et al., 02

Each dimension is relevant to at least one cluster

In General: Clusters may exist in different subspaces, comprised of different combinations of features

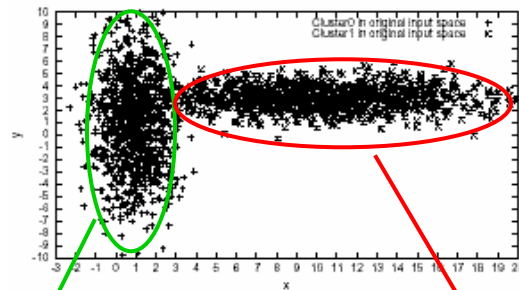
Local Dimensionality Reduction

- To capture the local correlations of data, a proper feature selection procedure should operate locally;
- A local operation allows to embed different distance measures in different regions;

The Idea

- We wish to *learn* from the data the relevant features for each cluster.
- Idea: *Soft* feature selection procedure
 - Assign (local) weights to features according to the local correlations of data along each dimension.

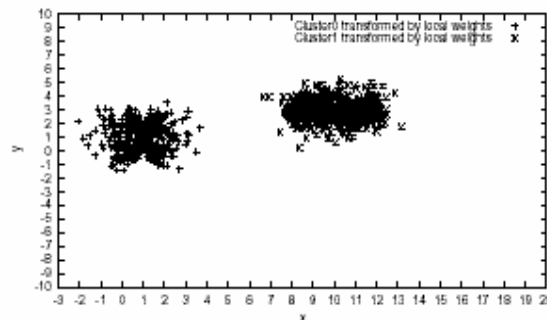
Locally Adaptive Clustering: Example



$$(w_{1x}, w_{1y}), w_{1x} > w_{1y}$$

$$(w_{2x}, w_{2y}), w_{2y} > w_{2x}$$

Locally Adaptive Clustering: Example



Within-cluster distances between points are computed using the respective local weights

Locally Adaptive Clustering (LAC)

- **Weighted cluster:** subset of data points, together with a weight vector, such that the points are closely clustered according to the corresponding weighted Euclidean distance;
- **Objective:** Find cluster centroids, and weight vectors.

LAC: Overall Approach

- **Initialization:** an initial set of centroids is chosen;
- **Iterative phase:** Compute weights within a locality of each centroid, and resulting clustering.
- Update centroids and iterate Until no change occurs.

LAC

- **Input parameter:**

k : the number of clusters

- **Initialization:**

$\mathbf{c}_1, \dots, \mathbf{c}_k$

$w_{ji} = \frac{1}{\sqrt{q}}$, for all centroids j and all features i

- **Partition the data:**

$S_j = \{x | j = \arg \min_l D_w(c_l, x)\}$

$$D_w = \sqrt{\sum_{i=1}^q w_{ji} (c_{ji} - x_i)^2}$$

LAC

- **Computing the weights:**

X_{ji} : average squared distance along dimension i of points in

S_j from \mathbf{c}_j

$$X_{ji} = \frac{1}{|S_j|} \sum_{x \in S_j} (c_{ji} - x_i)^2$$

$$w_{ji} = \frac{e^{-X_{ji}}}{\sum_l e^{-X_{jl}}}$$

Exponential weighting scheme

Result:

w_1, w_2, \dots, w_k

A weight vector for each cluster

LAC

- Forming new clusters:

Given the centroids and their associated weights, assign each point to the closest centroid j (with respect to the weighted Euclidean distance D_w) - the result is a new partition S_j

Update centroids:
$$c_j = \frac{\sum_x \mathbf{x} 1_{S_j(x)}}{\sum_x 1_{S_j(x)}}$$

Iterate, until convergence.

Convergence of LAC

The LAC algorithm converges to a local minimum of the error function:

$$E(C, W) = \sum_{j=1}^k \sum_{i=1}^q w_{ji} e^{-X_{ji}}$$

subject to the constraints $\sum_{i=1}^q w_{ji}^2 = 1 \quad \forall j$

$$C = [c_1 \cdots c_k] \quad W = [w_1 \cdots w_k]$$

EM-like convergence:

Hidden variables: assignments of points to centroids (S_j)

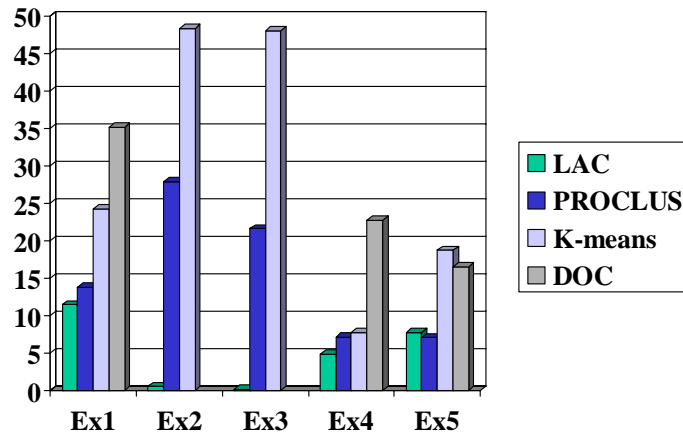
E-step: find the values of S_j given w_{ji}, c_{ji}

M-step: find w_{ji}, c_{ji} that minimize $E(C, W)$ given current estimates S_j .

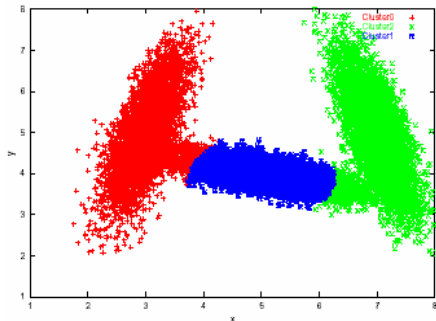
Simulated Data

- ❖ **Example 1:** $q=2$, $k=3$. Multivariate Gaussian clusters. Mean vectors and std dev: (2,0) and (4,1); (10,0) and (1,4); (18,0) and (4,1).
- ❖ **Example 2:** $q=30$, $k=2$. Multivariate Gaussian clusters. Mean vectors and std dev: (1,...,1) and (10,5,...,10,5); (2,1,...,1) and (5,10,...,5,10).
- ❖ **Example 3:** $q=50$, $k=2$. Multivariate Gaussian clusters. Mean vectors and std dev: (1,...,1) and (20,10,...,20,10); (2,1,...,1) and (10,20,...,10,20).
- ❖ **Example 4:** $q=2$, $k=2$. Multivariate Gaussian clusters, off-axis oriented.
- ❖ **Example 5:** $q=2$, $k=3$. Multivariate Gaussian clusters, off-axis oriented.

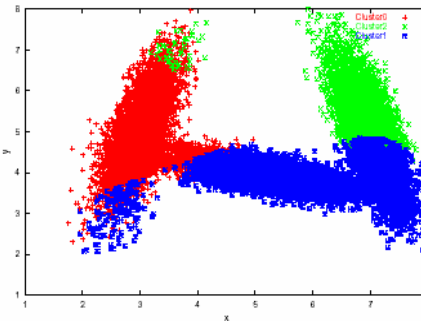
Error Rates for Simulated Data



Example 5



lac Error rate: 7.7%



K-means

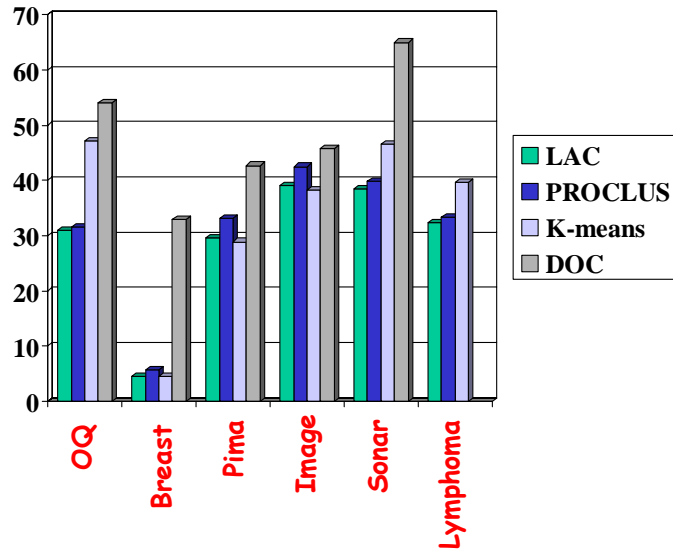
Error rate: 18.7%

Cluster	w_1	w_2
C0	0.92	0.08
C1	0.44	0.56
C2	0.94	0.06

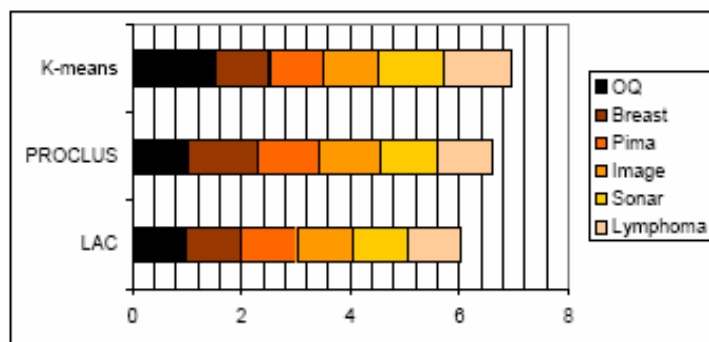
Real Data

- ❖ **OQ**: $q=16$, $k=2$. 1536 records.
- ❖ **Breast**: $q=9$, $k=2$. 683 records.
- ❖ **Pima**: $q=8$, $k=2$. 768 records.
- ❖ **Image**: $q=16$, $k=15$. 640 records.
- ❖ **Sonar**: $q=60$, $k=2$. 208 records.
- ❖ **Lymphoma**: $q=4026$, $k=9$. 96 records.

Error Rates for Real Data



Performance Distributions over Real Data



$$b_m = e_m / \min_{1 \leq k \leq 3} e_k$$

LAC: Subspace clustering of Microarray data

- **Aim:** Cluster genes according to their expression levels across different conditions.
- We can apply LAC to the gene vectors.
- By analyzing the distribution of weight values within each identified cluster, we can determine the correlations between genes and conditions.

Results with Microarray data

	<i>LAC</i>	<i>PROCLUS</i>
<i>k</i> = 3		
<i>C0</i> (size, score)	1220×5, 11.98	1635×4, 9.41
<i>dimensions</i>	9,13,14,19,22	7,8,9,13
<i>C1</i> (size, score)	1052×5, 1.07	1399×6, 48.18
<i>dimensions</i>	7,8,9,13,18	7,8,9,13,19,22
<i>C2</i> (size, score)	954×4, 5.32	192×5, 2.33
<i>dimensions</i>	12,13,16,18	2,7,10,19,22
<i>k</i> = 4		
<i>C0</i> (size, score)	1701×5, 4.52	1249×5, 3.90
<i>dimensions</i>	7,8,9,19,22	7,8,9,13,22
<i>C1</i> (size, score)	1255×5, 3.75	1229×6, 42.74
<i>dimensions</i>	7,8,9,13,22	7,8,9,13,19,22
<i>C2</i> (size, score)	162 outliers	730×4, 15.94
<i>dimensions</i>	-	7,8,9,13
<i>C3</i> (size, score)	108 outliers	18×5, 3.97
<i>dimensions</i>	-	6,11,14,16,21

$$Score(I, J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{i\cdot} - a_{\cdot j} + a_{\cdot\cdot})^2$$

Mean squared residue score
(Cheng and Church, 00)

Goal: find clusters with low mean squared residue score.

Biological processes annotated in one cluster generated by the LAC algorithm

There exists a number of cell cycle genes. The terms for cell cycle regulation all score high. As with all cancers, BRCA1-BRCA2-related tumors involve the loss of control over cell growth and proliferation. Thus, the presence of strong cell-cycle components in the clustering is expected.

Biological process	z-score
DNA damage checkpoint	7.4
nucleocytoplasmic transport	7.4
meiotic recombination	7.4
asymmetric cytokinesis	7.4
purine base biosynthesis	7.4
GMP biosynthesis	5.1
rRNA processing	5.1
glutamine metabolism	5.1
establishment and/or	5.1
maintenance of cell polarity	
gametogenesis	5.1
DNA replication	4.6
cell cycle arrest	4.4
central nervous system	4.4
development	
purine nucleotide	4.1
biosynthesis	
mRNA splicing	4.1
cell cycle	3.5
negative regulation of cell	3.4
proliferation	
induction of apoptosis by	2.8
intracellular signals	
oncogenesis	2.6
G1/S transition of mitotic	2.5
cell cycle	
protein kinase cascade	2.5
glycogen metabolism	2.3
regulation of cell cycle	2.1

Conclusions

- The output of LAC is twofold.
 - It provides a partition of data, so that the points in each set of the partition constitute a cluster.
- Each cluster is associated with a weight vector, whose values capture the relevance of features within the corresponding cluster.
- We experimentally demonstrate the gain in performance our method achieves, using both synthetic and real data sets.
- Our results show the feasibility of the proposed technique to perform simultaneous clustering of genes and conditions in microarray data.

Ευχαριστώ!