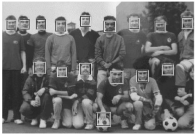
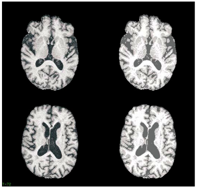
Estimation of Mixture Subspace Models
– Its Algebra, Statistics, and Compressed Sensing

Allen Y. Yang
<yang@eecs.berkeley.edu>

April 17, 2008, UPenn

Motivation

- Modern data in HD applications are often characterized as *multimodal, multivariate*: Subsets of the data are modeled by different distributions.

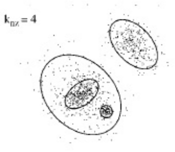
  - Face recognition
  
  - Natural image segmentation
  
  - MR image segmentation
  
  - MR video segmentation

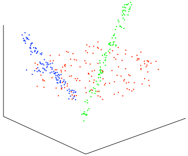
  trackers movie

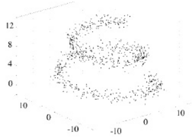
Estimation of Mixture Models

Keep these questions in mind:

- How to determine a class of models and the number of models?
- Robust to high noise and outliers?
- Purpose of segmentation and classification for higher-level applications?
  e.g., motion segmentation, image categorization, object recognition.



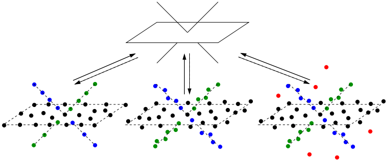




Outline

Pattern analysis in two domains

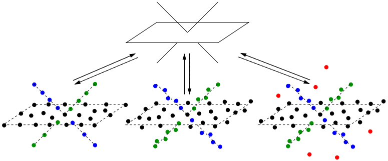
1. **Unsupervised** segmentation
   Segment samples drawn from $\mathcal{A} = V_1 \cup V_2 \cup \ldots \cup V_K$ in $\mathbb{R}^D$, and estimate subspace models.



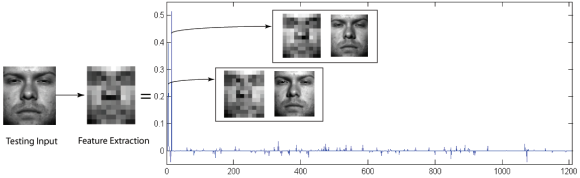
Outline

Pattern analysis in two domains

1. **Unsupervised** segmentation
   Segment samples drawn from $\mathcal{A} = V_1 \cup V_2 \cup \ldots \cup V_K$ in $\mathbb{R}^D$, and estimate subspace models.



2. **Supervised** recognition
   Assume training examples $\{A_1, \cdots, A_K\}$ for K subspaces. Given a test sample $\mathbf{y}$, determine its membership $\text{label}(\mathbf{y}) \in [1, 2, \cdots, K]$.



Outline

Literature Overview:

1. Unsupervised Segmentation:
   - PCA [Pearson 1901, Eckart-Young 1930, Hotelling 1933, Jolliffe 1986]
   - EM [Dempster 1977, McLachlan 1997]
   - RANSAC [Fischler 1981, Torr 1997, Schindler 2005]
2. Supervised Classification:
   - Nearest neighbors
   - Nearest subspaces [Kriegman 2003]
   - Support vector machines (SVMs) [Vapnik 1995, Cover 1965]

Outline

Literature Overview:

1. Unsupervised Segmentation:
   - PCA [Pearson 1901, Eckart-Young 1930, Hotelling 1933, Jolliffe 1986]
   - EM [Dempster 1977, McLachlan 1997]
   - RANSAC [Fischler 1981, Torr 1997, Schindler 2005]
2. Supervised Classification:
   - Nearest neighbors
   - Nearest subspaces [Kriegman 2003]
   - Support vector machines (SVMs) [Vapnik 1995, Cover 1965]

Presentation

1. References
   - *Generalized Principal Component Analysis*, (in press) **SIAM Review**, 2008.
   - *Image Segmentation using Mixture Subspace Models*, (in press) **CVIU**, 2008.
   - *Robust Face Recognition via Sparse Representation*, (in press) **PAMI**, 2008.
2. All MATLAB codes are free download at: http://www.eecs.berkeley.edu/~yang/

Mixture Subspace Segmentation

- Application: Motion Segmentation

1. Object features $\mathbf{p}_1, \ldots, \mathbf{p}_N \in \mathbb{R}^3$ are tracked in F frames.

2. Denote $\mathbf{m}_{ij}$ as image of p_i under 3-D affine projection:

$$\mathbf{m}_{ij} \doteq (x_{ij}, y_{ij})^T = A_j \mathbf{p}_i + \mathbf{b}_j \in \mathbb{R}^2, \quad j = 1, \ldots, F.$$

parking-lot movie

Mixture Subspace Segmentation

- Application: Motion Segmentation

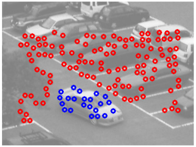
1. Object features $\mathbf{p}_1, \ldots, \mathbf{p}_N \in \mathbb{R}^3$ are tracked in F frames.
2. Denote $\mathbf{m}_{ij}$ as image of p_i under 3-D affine projection:

$$\mathbf{m}_{ij} \doteq (x_{ij}, y_{ij})^T = A_j \mathbf{p}_i + \mathbf{b}_j \in \mathbb{R}^2, \quad j = 1, \ldots, F.$$

3. For each p_i in space, represent all images in F frame

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{m}_{i1} \\ \vdots \\ \mathbf{m}_{iF} \end{bmatrix} \in \mathbb{R}^{2F}.$$

parking-lot movie



Challenge: Affine-Camera Motion Segmentation

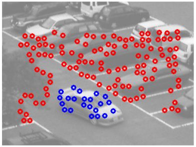
- Segment N points $\mathbf{x}_1, \cdots, \mathbf{x}_N$ into K groups that belong to different motions.

Mixture Subspace Segmentation

- Application: Motion Segmentation

1. Object features $\mathbf{p}_1, \ldots, \mathbf{p}_N \in \mathbb{R}^3$ are tracked in F frames.
2. Denote $\mathbf{m}_{ij}$ as image of p_i under 3-D affine projection:
$$\mathbf{m}_{ij} \doteq (x_{ij}, y_{ij})^T = A_j \mathbf{p}_i + \mathbf{b}_j \in \mathbb{R}^2, \quad j = 1, \ldots, F.$$
3. For each p_i in space, represent all images in F frame
$$\mathbf{x}_i = \begin{bmatrix} \mathbf{m}_{i1} \\ \vdots \\ \mathbf{m}_{iF} \end{bmatrix} \in \mathbb{R}^{2F}.$$

parking-lot movie



Challenge: Affine-Camera Motion Segmentation

- Segment N points $\mathbf{x}_1, \cdots, \mathbf{x}_N$ into K groups that belong to different motions.
- Each motion in fact satisfies a 4-D subspace model. Therefore it is a subspace segmentation problem.

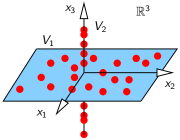
Generalized Principal Component Analysis

> If one wishes to shrink it, one must first expand it.
>
> – Lao Tzu

Generalized Principal Component Analysis

> If one wishes to shrink it, one must first expand it.
>
> – Lao Tzu

1. For a single subspace $V \subset \mathbb{R}^D$: $d \doteq \dim(V)$
   - $V_1^\perp : (x_3 = 0)$
   - $V_2^\perp : (x_1 = 0) \& (x_2 = 0)$

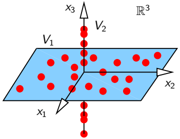


Generalized Principal Component Analysis

If one wishes to shrink it, one must first expand it.

– Lao Tzu

1. For a single subspace $V \subset \mathbb{R}^D$: $d \doteq \dim(V)$
   - $V_1^\perp : (x_3 = 0)$
   - $V_2^\perp : (x_1 = 0)\&(x_2 = 0)$



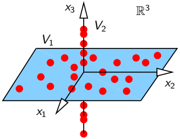
2. For $\mathcal{A} = V_1 \cup V_2$

$$\forall \mathbf{z} = (x_1, x_2, x_3)^T, \quad \mathbf{z} \in V_1 \cup V_2 \Leftrightarrow \{x_3 = 0\} | \{(x_1 = 0)\&(x_2 = 0)\}$$

Generalized Principal Component Analysis

> If one wishes to shrink it, one must first expand it.
>
> – Lao Tzu

1. For a single subspace $V \subset \mathbb{R}^D$: $d \doteq \dim(V)$
   - $V_1^\perp : (x_3 = 0)$
   - $V_2^\perp : (x_1 = 0)\&(x_2 = 0)$



2. For $\mathcal{A} = V_1 \cup V_2$

$$\forall \mathbf{z} = (x_1, x_2, x_3)^T, \quad \mathbf{z} \in V_1 \cup V_2 \Leftrightarrow \{x_3 = 0\} | \{(x_1 = 0)\&(x_2 = 0)\}$$

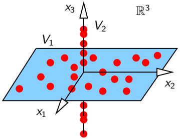
3. By De Morgan's law

$$\{x_3 = 0\} | \{(x_1 = 0)\&(x_2 = 0)\} \Leftrightarrow (x_1 x_3 = 0)\&(x_2 x_3 = 0) \Leftrightarrow \begin{cases} x_1 x_3 = 0 \\ x_2 x_3 = 0 \end{cases}$$

Generalized Principal Component Analysis

> If one wishes to shrink it, one must first expand it.
>
> – Lao Tzu

1. For a single subspace $V \subset \mathbb{R}^D$: $d \doteq \dim(V)$
   - $V_1^\perp : (x_3 = 0)$
   - $V_2^\perp : (x_1 = 0)\&(x_2 = 0)$



2. For $\mathcal{A} = V_1 \cup V_2$

$$\forall \mathbf{z} = (x_1, x_2, x_3)^T, \quad \mathbf{z} \in V_1 \cup V_2 \Leftrightarrow \{x_3 = 0\} | \{(x_1 = 0)\&(x_2 = 0)\}$$

3. By De Morgan's law

$$\{x_3 = 0\} | \{(x_1 = 0)\&(x_2 = 0)\} \Leftrightarrow (x_1 x_3 = 0)\&(x_2 x_3 = 0) \Leftrightarrow \begin{cases} x_1 x_3 = 0 \\ x_2 x_3 = 0 \end{cases}$$

4. **Vanishing polynomials**: $p_1 = x_1 x_3, p_2 = x_2 x_3$

Equivalence Relation

1. The **equivalence** between mixture subspaces and its vanishing polynomials
   - Given $p_1 = x_1x_3, p_2 = x_2x_3$, $V_1 \cup V_2$ uniquely determined.
   - Given $V_1 \cup V_2$, all vanishing polynomials of arbitrary degree generated by $p_1 = x_1x_3, p_2 = x_2x_3$.

Equivalence Relation

1. The **equivalence** between mixture subspaces and its vanishing polynomials
   - Given $p_1 = x_1 x_3, p_2 = x_2 x_3$, $V_1 \cup V_2$ uniquely determined.
   - Given $V_1 \cup V_2$, all vanishing polynomials of arbitrary degree generated by $p_1 = x_1 x_3, p_2 = x_2 x_3$.

A global signature

The set of linearly independent vanishing polynomials is a global signature for mixture K subspaces.

Equivalence Relation

1. The **equivalence** between mixture subspaces and its vanishing polynomials
   - Given $p_1 = x_1x_3, p_2 = x_2x_3$, $V_1 \cup V_2$ uniquely determined.
   - Given $V_1 \cup V_2$, all vanishing polynomials of arbitrary degree generated by $p_1 = x_1x_3, p_2 = x_2x_3$.

A global signature

The set of linearly independent vanishing polynomials is a global signature for mixture K subspaces.

2. Number of linearly independent vanishing polynomials [Derksen 2005]

$$h(K) = \sum_S (-1)^{|S|} \binom{K+D-1-c_S}{D-1-c_S},$$

where $S \subseteq \{1, \cdots, K\}$ is an index set and c denotes codimension.

Equivalence Relation

1. The **equivalence** between mixture subspaces and its vanishing polynomials
   - Given $p_1 = x_1x_3, p_2 = x_2x_3$, $V_1 \cup V_2$ uniquely determined.
   - Given $V_1 \cup V_2$, all vanishing polynomials of arbitrary degree generated by $p_1 = x_1x_3, p_2 = x_2x_3$.

A global signature

The set of linearly independent vanishing polynomials is a global signature for mixture K subspaces.

2. Number of linearly independent vanishing polynomials [Derksen 2005]

$$h(K) = \sum_S (-1)^{|S|} \binom{K+D-1-c_S}{D-1-c_S},$$

where $S \subseteq \{1, \cdots, K\}$ is an index set and c denotes codimension.

3. Example: linearly independent 3rd degree vanishing polynomials for 3 mixture subspaces

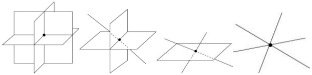


Figure: Four possible configurations in $\mathbb{R}^3$.

d_1	d_2	d_3	$h(3)$
2	2	2	1
2	2	1	2
2	1	1	4
1	1	1	7

Estimation of Vanishing Polynomials

1. **Veronese embedding**: Given N samples $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^3$,

$$
\begin{aligned}
L_2 &\doteq [\nu_2(\mathbf{x}_1), \ldots, \nu_2(\mathbf{x}_N)] \in \mathbb{R}^{M_2^{[3]} \times N} \\
&= \begin{bmatrix} \cdots & (x_1)^2 & \cdots \\ \cdots & (x_1 x_2) & \cdots \\ \cdots & (x_1 x_3) & \cdots \\ \cdots & (x_2)^2 & \cdots \\ \cdots & (x_2 x_3) & \cdots \\ \cdots & (x_3)^2 & \cdots \end{bmatrix}
\end{aligned}
$$

Estimation of Vanishing Polynomials

1. **Veronese embedding**: Given N samples $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^3$,

$$L_2 \doteq [\nu_2(\mathbf{x}_1), \ldots, \nu_2(\mathbf{x}_N)] \in \mathbb{R}^{M_2^{[3]} \times N} = \begin{bmatrix} \cdots & (x_1)^2 & \cdots \\ \cdots & (x_1 x_2) & \cdots \\ \cdots & (x_1 x_3) & \cdots \\ \cdots & (x_2)^2 & \cdots \\ \cdots & (x_2 x_3) & \cdots \\ \cdots & (x_3)^2 & \cdots \end{bmatrix}$$

2. The null space of L_2 is $\begin{array}{l} \mathbf{c}_1 = [0,0,1,0,0,0] \\ \mathbf{c}_2 = [0,0,0,0,1,0] \end{array} \Rightarrow \begin{array}{l} p_1 = \mathbf{c}_1 \nu_2(\mathbf{x}) = x_1 x_3 \\ p_2 = \mathbf{c}_2 \nu_2(\mathbf{x}) = x_2 x_3 \end{array}$

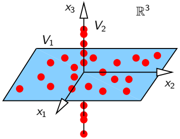


Figure: 2*nd*-degree vanishing polynomials: $p_1 = x_1 x_3$, $p_2 = x_2 x_3$.

Calculate Subspace Basis Vectors using Polynomial Derivatives

1. $V_1^\perp, \cdots, V_K^\perp$ recovered by the *derivatives*

$$\nabla_{\mathbf{x}} P = [\, \nabla_{\mathbf{x}} p_1 \;\; \nabla_{\mathbf{x}} p_2 \,] = \begin{bmatrix} x_3 & 0 \\ 0 & x_3 \\ x_1 & x_2 \end{bmatrix}.$$

2. Pick $\mathbf{z} = [1, 1, 0]^T \in V_1$, then $\nabla_{\mathbf{x}} P(\mathbf{z}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$.

   Pick $\mathbf{z} = [0, 0, 1]^T \in V_2$, then $\nabla_{\mathbf{x}} P(\mathbf{z}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$.

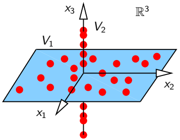


Figure: $P(\mathbf{x}) \doteq [p_1(\mathbf{x}) \;\; p_2(\mathbf{x})] = [x_1 x_3, x_2 x_3]$.

Calculate Subspace Basis Vectors using Polynomial Derivatives

1. $V_1^\perp, \cdots, V_K^\perp$ recovered by the *derivatives*

$$\nabla_{\mathbf{x}} P = [\, \nabla_{\mathbf{x}} p_1 \;\; \nabla_{\mathbf{x}} p_2 \,] = \begin{bmatrix} x_3 & 0 \\ 0 & x_3 \\ x_1 & x_2 \end{bmatrix}.$$

2. Pick $\mathbf{z} = [1, 1, 0]^T \in V_1$, then $\nabla_{\mathbf{x}} P(\mathbf{z}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$.

   Pick $\mathbf{z} = [0, 0, 1]^T \in V_2$, then $\nabla_{\mathbf{x}} P(\mathbf{z}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$.

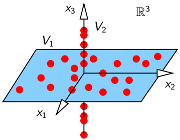
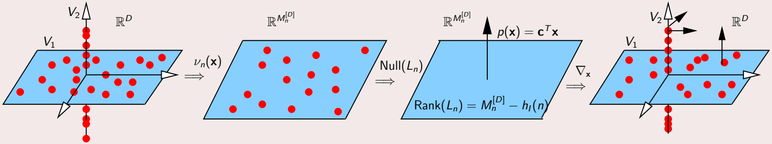


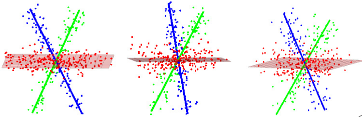
Figure: $P(\mathbf{x}) \doteq [p_1(\mathbf{x}) \;\; p_2(\mathbf{x})] = [x_1 x_3, x_2 x_3]$.

Diagram of GPCA



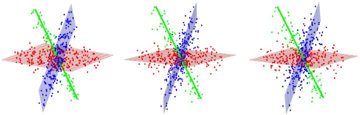
Stability and Robustness of GPCA

- GPCA is **stable** to moderate data noise



(a) .08 (b) .12 (c) .16

Figure: (2, 1, 1) with various noise-to-signal ratios



(a) .08 (b) .12 (c) .16

Figure: (2, 2, 1) with various noise-to-signal ratios

Stability and Robustness of GPCA

- GPCA is **stable** to moderate data noise

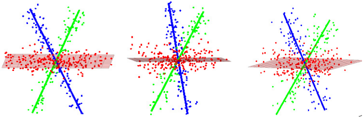

(a) .08 (b) .12 (c) .16

Figure: (2, 1, 1) with various noise-to-signal ratios

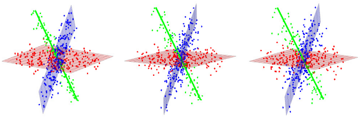
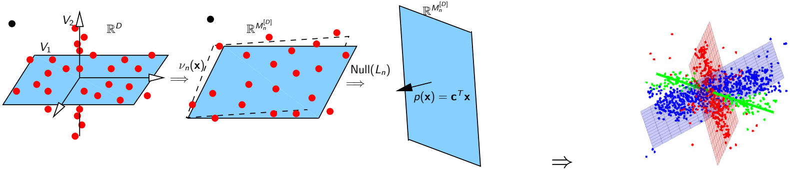

(a) .08 (b) .12 (c) .16

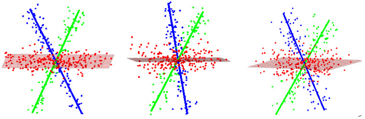
Figure: (2, 2, 1) with various noise-to-signal ratios

- GPCA is not robust to outliers: single outlier can arbitrarily perturb Null(L_n)



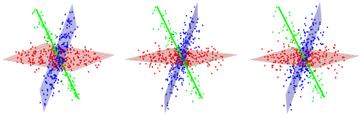
Stability and Robustness of GPCA

- GPCA is **stable** to moderate data noise



(a) .08 (b) .12 (c) .16

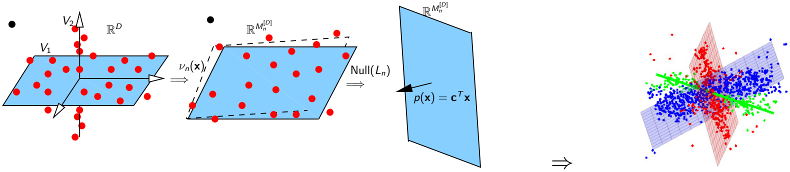
Figure: $(2, 1, 1)$ with various noise-to-signal ratios



(a) .08 (b) .12 (c) .16

Figure: $(2, 2, 1)$ with various noise-to-signal ratios

- GPCA is not robust to outliers: single outlier can arbitrarily perturb $\mathrm{Null}(L_n)$

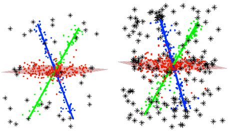


- **Breakdown point** of GPCA is 0%.

Robust GPCA

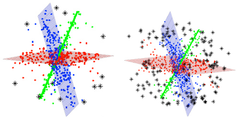
Breakdown of PCA is 0% $\Rightarrow$ Replace with robust PCA to estimate $\mathrm{Null}(L_n)$

- One plane and two lines.



(a) 12% (b) 32%

- Two planes and one line.



(a) 12% (b) 32%

Outlier Elimination

Figure: Elimination of outliers.

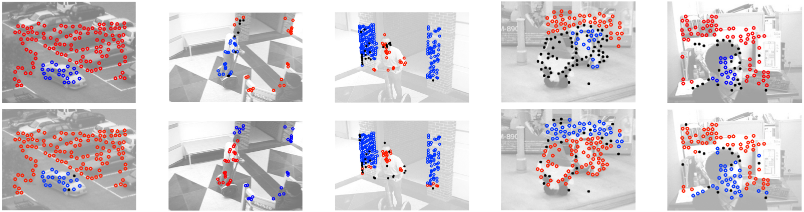
Reference: Jolliffe, **Principal Component Analysis**, Springer.

Affine Motion Segmentation

Sequences:

RANSAC:

RGPCA:



Reference: **Robust statistical estimation and segmentation of multiple subspaces**. *CVPR Workshop on 25 Years of RANDAC*, 2006.

Dynamic Texture

1. Dynamic texture as **ARMA model** for image $I(t)$ of D pixels

$$
\begin{aligned}
x(t+1) &= Ax(t) + Bu(t) \in \mathbb{R}^d \\
I(t) &= Cx(t) + w(t) \in \mathbb{R}^D
\end{aligned}
$$

Dynamic Texture

1. Dynamic texture as **ARMA model** for image $I(t)$ of D pixels

$$x(t+1) = Ax(t) + Bu(t) \in \mathbb{R}^d$$
$$I(t) = Cx(t) + w(t) \in \mathbb{R}^D$$

2. Subspace constraint on F frames, $t = 1, \cdots, F$

trackers movie

$$L = [I(1) \cdots I(F)] = C[x(1) \cdots x(F)] \in \mathbb{R}^{D \times F};$$
$$\Rightarrow \qquad \mathrm{Rank}(\mathbf{L}) = \mathbf{d}.$$

Dynamic Texture

1. Dynamic texture as **ARMA model** for image $I(t)$ of D pixels

$$x(t+1) = Ax(t) + Bu(t) \in \mathbb{R}^d$$
$$I(t) = Cx(t) + w(t) \in \mathbb{R}^D$$

2. Subspace constraint on F frames, $t = 1, \cdots, F$

trackers movie

$$L = [I(1) \cdots I(F)] = C[x(1) \cdots x(F)] \in \mathbb{R}^{D \times F};$$
$$\Rightarrow \quad \text{Rank}(\mathbf{L}) = \mathbf{d}.$$

Mixture of two dynamic texture regions

$$[i(1) \cdots i(F)] \in \text{Sys}(A_1, B_1, C_1) \quad \text{or} \quad [i(1) \cdots i(F)] \in \text{Sys}(A_2, B_2, C_2)$$

Dynamic Texture

1. Dynamic texture as **ARMA model** for image $I(t)$ of D pixels

$$x(t+1) = Ax(t) + Bu(t) \in \mathbb{R}^d$$
$$I(t) = Cx(t) + w(t) \in \mathbb{R}^D$$

2. Subspace constraint on F frames, $t = 1, \cdots, F$

trackers movie

$$L = [I(1) \cdots I(F)] = C[x(1) \cdots x(F)] \in \mathbb{R}^{D \times F};$$
$$\Rightarrow \qquad \mathrm{Rank}(\mathbf{L}) = \mathbf{d}.$$

Mixture of two dynamic texture regions

$$[i(1) \cdots i(F)] \in \mathrm{Sys}(A_1, B_1, C_1) \quad \text{or} \quad [i(1) \cdots i(F)] \in \mathrm{Sys}(A_2, B_2, C_2)$$

Reference: Ravichandran, et al., **Segmenting a beating heart using polysegment and spatial GPCA**, Int Sym Biomedical Imaging, 2006.

GPCA Website: http://perception.csl.uiuc.edu/gpca/

Generalized Principal Component Analysis

Welcome

Introduction

Sample Code

Applications

Publications

Sponsors

JHU

Berkeley

NSF

About GPCA

In many scientific and engineering problems, the data of interest can be viewed as drawn from a mixture of geometric or statistical models instead of a single one. Such data are often referred to in different contexts as ``mixed," or ``multi-modal," or ``multi-model," or ``heterogeneous," or ``hybrid." For instances, a natural image normally consists of multiple regions of different texture, a video sequence may contains multiple independently moving objects, and a hybrid dynamical system may arbitrarily switch among different subsystems.

Generalized Principal Component Analysis (GPCA) is a general method for modeling and segmenting such mixed data using a collection of subspaces, also known in mathematics as a subspace arrangement. By introducing certain new algebraic models and techniques into data clustering, traditionally a statistical problem, GPCA offers a new spectrum of algorithms for data modeling and clustering that are in many aspects more efficient and effective than (or complementary to) traditional methods (e.g. Expectation Maximization and K-Means).

The goal of this site is to promote the use of the GPCA algorithm to improve segmentation performance in many application domains. Tutorials and sample code are provided to help researchers and practitioners decide if the algorithm can be applied to their application domain, and to help get their implementation set up quickly and correctly.

Browsing through the links on the left, you will find a brief overview of the fundamental concepts behind GPCA in the *Introduction* section; numerical implementations of several variations of the GPCA algorithm in the *Sample Code* section; examples of real applications in the areas of computer vision, image processing; and system identification in the *Applications* section; and finally all the related literature in the *Publications* section.

Website Credits

This site is jointly developed and maintained by the research groups of

- Professor Yi Ma of the Electrical & Computer Engineering Department at the University of Illinois at Urbana-Champaign
- Professor Rene Vidal of the Biomedical Engineering Department at the Johns Hopkins University
- Professor Kun Huang of the Biomedical Informatics Department at the Ohio State University

Summary

Advantages:

- A global algebraic framework. Solution is not iterative.
- If the subspace models are known, likely outperforms other classical solutions.
- Robust to noise and outliers.

Summary

Advantages:

- A global algebraic framework. Solution is not iterative.
- If the subspace models are known, likely outperforms other classical solutions.
- Robust to noise and outliers.

Limitations:

- Need to provide correct subspace number.
- High-dimensional polynomial space brings high computation complexity.

Summary

Advantages:

- A global algebraic framework. Solution is not iterative.
- If the subspace models are known, likely outperforms other classical solutions.
- Robust to noise and outliers.

Limitations:

- Need to provide correct subspace number.
- High-dimensional polynomial space brings high computation complexity.

Next Section:

- Novel segmentation method that operates on original data space, and simultaneously estimate subspace number.

Image Segmentation based on Texture

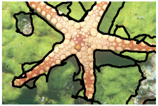
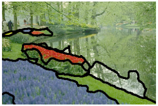
    

Image Segmentation based on Texture

1. If the subspace number and dimensions are not given, segmentation is naturally **ambiguous**.

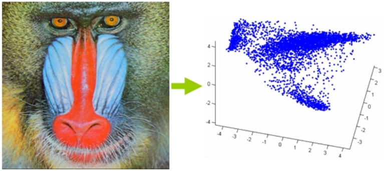
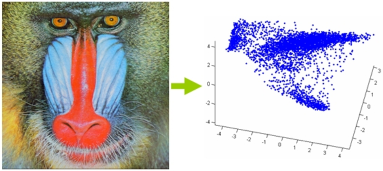


Image Segmentation based on Texture

1. If the subspace number and dimensions are not given, segmentation is naturally **ambiguous**.



2. **Minimum Description Length** (MDL): Given $V = (v_1, \cdots, v_N) \in \mathbb{R}^{D \times N}$, a lossy coding function maps the vectors to a binary sequence

$$1, 0, 0, 1, 1|0, 0, 1, 0, \cdots$$

such that V can be recovered up to a distortion $\mathbb{E}[\|v_i - \hat{v}_i\|^2] \leq \epsilon^2$.

Optimal segmentation $V = V_1 \cup \cdots \cup V_K$ produces the shortest coding length.

Lossy MDL for Mixture Subspace Models

1. Coding length for *i*-**th Gaussian model** of N_i samples

$$L(V_i) = (N_i + D)\frac{1}{2}\log_2 \det(I + \frac{D}{\epsilon^2 N_i} V_i V_i^T) + \frac{D}{2}\log_2 \det(1 + \frac{1}{\epsilon^2}\mu_i \mu_i^T) + N_i(-\log_2(N_i/N)).$$

Lossy MDL for Mixture Subspace Models

1. Coding length for ***i*-th Gaussian model** of N_i samples

$$L(V_i) = (N_i + D)\frac{1}{2}\log_2 \det(I + \frac{D}{\epsilon^2 N_i} V_i V_i^T) + \frac{D}{2}\log_2 \det(1 + \frac{1}{\epsilon^2}\mu_i \mu_i^T) + N_i(-\log_2(N_i/N)).$$

2. **Total coding length** for mixture K Gaussians

$$L^s(V_1, \cdots, V_K) = \sum_{i=1,\cdots,K} L(V_i)$$

Lossy MDL for Mixture Subspace Models

1. Coding length for ***i*-th Gaussian model** of N_i samples

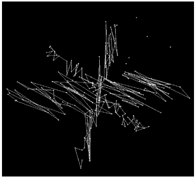
$$L(V_i) = (N_i + D)\frac{1}{2}\log_2 \det(I + \frac{D}{\epsilon^2 N_i} V_i V_i^T) + \frac{D}{2}\log_2 \det(1 + \frac{1}{\epsilon^2}\mu_i \mu_i^T) + N_i(-\log_2(N_i/N)).$$

2. **Total coding length** for mixture K Gaussians

$$L^s(V_1, \cdots, V_K) = \sum_{i=1,\cdots,K} L(V_i)$$

3. Minimization/segmentation via **region-merging optimization**

animation



Simulation

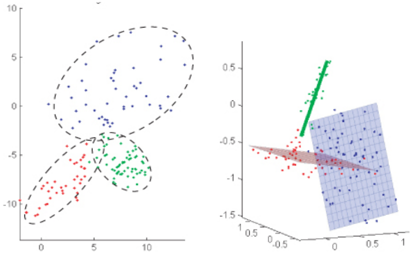


Figure: General mixture Gaussians and degenerate subspaces.

Simulation

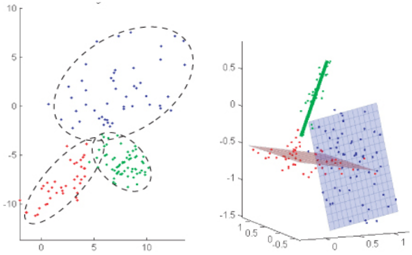


Figure: General mixture Gaussians and degenerate subspaces.

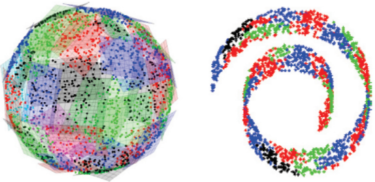
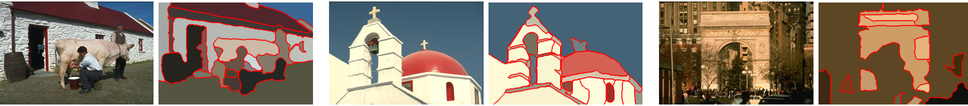


Figure: Linear fitting of nonlinear manifolds.

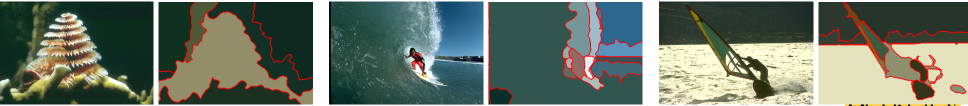
Natural Image Segmentation

(a) Nature



(b) Urban

(c) Portraits



(d) Water

www.eecs.berkeley.edu/~yang/software/lossy_segmentation/

Unsupervised Segmentation of Natural Images via Lossy Data Compression

Allen Y. Yang, John Wright, Yi Ma, and Shankar Sastry

ABSTRACT:

We cast natural-image segmentation as a problem of clustering texure features as multivariate mixed data. We model the distribution of the texture features using a mixture of Gaussian distributions. Unlike most existing clustering methods, we allow the mixture components to be degenerate or nearly-degenerate. We contend that this assumption is particularly important for mid-level image segmentation, where degeneracy is typically introduced by using a common feature representation for different textures in an image. We show that such a mixture distribution can be effectively segmented by a simple agglomerative clustering algorithm derived from a lossy data compression approach. Using either 2D texture filter banks or simple fixed-size windows as texture features, the algorithm effectively segments an image by minimizing the overall coding length of the feature vectors. We conduct comprehensive experiments to measure the performance of the algorithm in terms of visual evaluation and a variety of quantitative indices for image segmentation. The algorithm compares favorably against other well-known image segmentation methods on the Berkeley image database.

- **Publications:**

  Allen Y. Yang, John Wright, Yi Ma, and Shankar Sastry. *Unsupervised segmentation of natural images via lossy data compression*. To appear in CVIU 2007. [PDF]

Face Recognition: *"Where amazing happens"*

Face Recognition: *"Where amazing happens"*

Face Recognition: *"Where amazing happens"*

Face Recognition: *"Where amazing happens"*

Face Recognition: *"Where amazing happens"*

Figure: Steve Nash, Kevin Garnett, Jason Kidd, Yao Ming.

Problem Formulation

1. **Notation**
   - Training: For K classes, collect training samples $\{\mathbf{v}_{1,1}, \cdots, \mathbf{v}_{1,n_1}\}, \cdots, \{\mathbf{v}_{K,1}, \cdots, \mathbf{v}_{K,n_K}\} \in \mathbb{R}^D$.
   - Test: Present a new $\mathbf{y} \in \mathbb{R}^D$, solve for label$(\mathbf{y}) \in [1, 2, \cdots, K]$.

Problem Formulation

1. **Notation**
   - Training: For K classes, collect training samples $\{\mathbf{v}_{1,1}, \cdots, \mathbf{v}_{1,n_1}\}, \cdots, \{\mathbf{v}_{K,1}, \cdots, \mathbf{v}_{K,n_K}\} \in \mathbb{R}^D$.
   - Test: Present a new $\mathbf{y} \in \mathbb{R}^D$, solve for label($\mathbf{y}$) $\in [1, 2, \cdots, K]$.
2. **Data representation** in (long) vector form via stacking

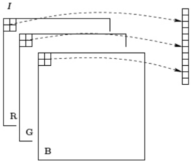


Figure: Assume 3-channel 640×480 image, $D = 3 \cdot 640 \cdot 480$.

Problem Formulation

1. **Notation**
   - Training: For K classes, collect training samples $\{\mathbf{v}_{1,1}, \cdots, \mathbf{v}_{1,n_1}\}, \cdots, \{\mathbf{v}_{K,1}, \cdots, \mathbf{v}_{K,n_K}\} \in \mathbb{R}^D$.
   - Test: Present a new $\mathbf{y} \in \mathbb{R}^D$, solve for label($\mathbf{y}$) $\in [1, 2, \cdots, K]$.
2. **Data representation** in (long) vector form via stacking

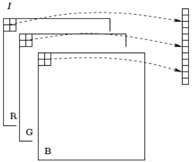
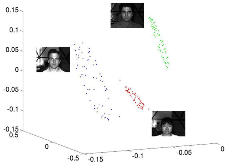


Figure: Assume 3-channel 640×480 image, $D = 3 \cdot 640 \cdot 480$.

3. **Mixture subspace model** for face recognition [Belhumeur et al. 1997, Basri & Jocobs 2003]



Sparse Representation

Sparsity

A signal is sparse if most of its coefficients are (approximately) zero.

Sparse Representation

Sparsity

A signal is sparse if most of its coefficients are (approximately) zero.

1. Sparsity in frequency domain

Figure: 2-D DCT transform.

2. Sparsity in spatial domain

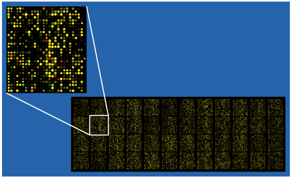
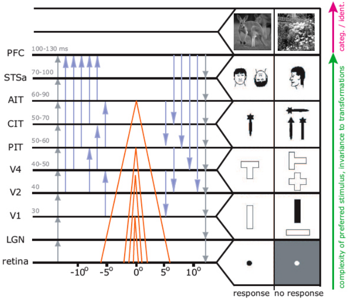
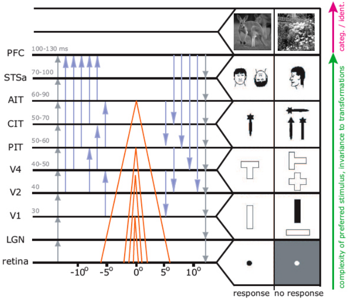


Figure: Gene microarray data.

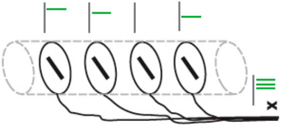
- **Sparsity in human visual cortex** [Olshausen & Field 1997, Serre & Poggio 2006]



- **Sparsity in human visual cortex** [Olshausen & Field 1997, Serre & Poggio 2006]

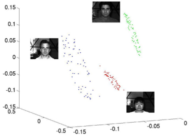


1. **Feed-forward**: No iterative feedback loop.
2. **Redundancy**: Average 80-200 neurons for each feature representation.
3. **Recognition**: Information exchange between stages is not about individual neurons, but rather how many neurons as a group fire together.



Classification of Mixture Subspace Model

1. Assume $\mathbf{y}$ belongs to Class i

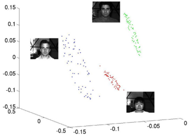


$$
\begin{aligned}
\mathbf{y} &= \alpha_{i,1}\mathbf{v}_{i,1} + \alpha_{i,2}\mathbf{v}_{i,2} + \cdots + \alpha_{i,n_1}\mathbf{v}_{i,n_i}, \\
&= A_i \alpha_i,
\end{aligned}
$$

where $A_i = [\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \cdots, \mathbf{v}_{i,n_i}]$.

Classification of Mixture Subspace Model

1. Assume $\mathbf{y}$ belongs to Class i



$$\begin{aligned} \mathbf{y} &= \alpha_{i,1}\mathbf{v}_{i,1} + \alpha_{i,2}\mathbf{v}_{i,2} + \cdots + \alpha_{i,n_1}\mathbf{v}_{i,n_i}, \\ &= A_i\alpha_i, \end{aligned}$$

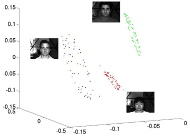
where $A_i = [\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \cdots, \mathbf{v}_{i,n_i}]$.

2. Nevertheless, Class i is the **unknown** variable we need to solve:

Sparse representation $\mathbf{y} = [A_1, A_2, \cdots, A_K] \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_K \end{bmatrix} = A\mathbf{x}.$

Classification of Mixture Subspace Model

1. Assume $\mathbf{y}$ belongs to Class i



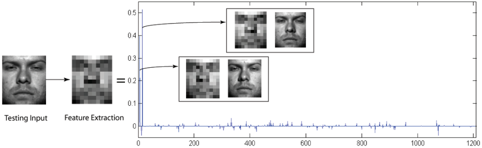
$$\begin{aligned} \mathbf{y} &= \alpha_{i,1}\mathbf{v}_{i,1} + \alpha_{i,2}\mathbf{v}_{i,2} + \cdots + \alpha_{i,n_1}\mathbf{v}_{i,n_i}, \\ &= A_i \alpha_i, \end{aligned}$$

where $A_i = [\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \cdots, \mathbf{v}_{i,n_i}]$.

2. Nevertheless, Class i is the **unknown** variable we need to solve:

Sparse representation $\quad \mathbf{y} = [A_1, A_2, \cdots, A_K] \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_K \end{bmatrix} = A\mathbf{x}.$

3. $\mathbf{x}_0 = [\,0 \cdots 0 \;\alpha_i^T\; 0 \cdots 0\,]^T \in \mathbb{R}^n.$



Sparse representation $\mathbf{x}_0$ encodes membership!

ℓ^1-Minimization

1. Ideal solution: ℓ^0-Minimization

$$(P_0) \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ s.t. } \mathbf{y} = A\mathbf{x}.$$

$\|\cdot\|_0$ simply counts the number of nonzero terms.
However, generally ℓ^0-minimization is *NP-hard*.

ℓ^1-Minimization

1. Ideal solution: ℓ^0-Minimization

$$(P_0) \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ s.t. } \mathbf{y} = A\mathbf{x}.$$

   $\|\cdot\|_0$ simply counts the number of nonzero terms.
   However, generally ℓ^0-minimization is *NP-hard*.

2. **Compressed sensing**: Under mild condition, ℓ^0-minimization is equivalent to

$$(P_1) \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ s.t. } \mathbf{y} = A\mathbf{x},$$

   where $\|\mathbf{x}\|_1 = |x_1| + |x_2| + \cdots + |x_n|$.

ℓ^1-Minimization

1. Ideal solution: ℓ^0-Minimization

$$(P_0) \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ s.t. } \mathbf{y} = A\mathbf{x}.$$

   $\|\cdot\|_0$ simply counts the number of nonzero terms.
   However, generally ℓ^0-minimization is *NP-hard*.

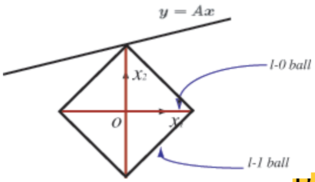
2. **Compressed sensing**: Under mild condition, ℓ^0-minimization is equivalent to

$$(P_1) \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ s.t. } \mathbf{y} = A\mathbf{x},$$

   where $\|\mathbf{x}\|_1 = |x_1| + |x_2| + \cdots + |x_n|$.

3. ℓ^1-Ball

- ℓ^1-Minimization is convex.
- Solution equal to ℓ^0-minimization.



ℓ^1-Minimization Routines

- **Matching pursuit** [Mallat 1993]
  1. Find most correlated vector $\mathbf{v}_i$ in A with $\mathbf{y}$: $i = \arg\max \langle \mathbf{y}, \mathbf{v}_j \rangle$.
  2. $A \leftarrow A^{(i)}$, $x_i \leftarrow \langle \mathbf{y}, \mathbf{v}_i \rangle$, $\mathbf{y} \leftarrow \mathbf{y} - x_i \mathbf{v}_i$.
  3. Repeat until $\|\mathbf{y}\| < \epsilon$.
- **Basis pursuit** [Chen 1998]
  1. Start with number of sparse coefficients $m = 1$.
  2. Select m linearly independent vectors B_m in A as a basis
  $$\mathbf{x}_m = B_m^\dagger \mathbf{y}.$$
  3. Repeat swapping one basis vector in B_m with another vector not in B_m if improve $\|\mathbf{y} - B_m \mathbf{x}_m\|$.
  4. If $\|\mathbf{y} - B_m \mathbf{x}_m\|_2 < \epsilon$, stop; Otherwise, $m \leftarrow m + 1$, repeat Step 2.
- **Quadratic solvers**: $\mathbf{y} = A\mathbf{x}_0 + \mathbf{z} \in \mathbb{R}^d$, where $\|\mathbf{z}\|_2 < \epsilon$
$$\mathbf{x}^* = \arg\min\{\|\mathbf{x}\|_1 + \lambda \|\mathbf{y} - A\mathbf{x}\|_2\}$$
[LASSO, Second-order cone programming]: Much more expensive.

Matlab Toolboxes for ℓ^1-Minimization

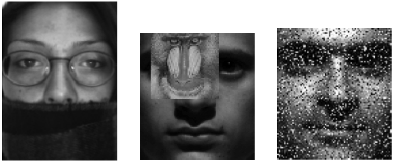
- **ℓ^1-Magic** by Candes
- **SparseLab** by Donoho
- **cvx** by Boyd

Partial Features on Extended Yale B Database

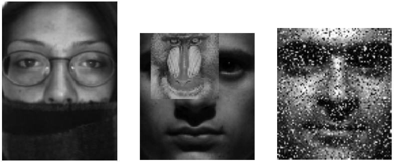
Features	Nose	Right Eye	Mouth & Chin
Dimension	4,270	5,040	12,936
SRC [%]	87.3	93.7	98.3
nearest-neighbor [%]	49.2	68.8	72.7
nearest-subspace [%]	83.7	78.6	94.4
Linear SVM [%]	70.8	85.8	95.3

SRC: sparse-representation classifier

Occlusion Compensation

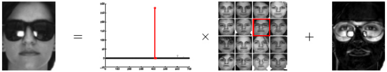


Occlusion Compensation



1. Sparse representation + sparse error

$$\mathbf{y} = A\mathbf{x} + \mathbf{e}$$



2. Occlusion compensation

$$\mathbf{y} = \begin{pmatrix} A & | & I \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{e} \end{pmatrix} = B\mathbf{w}$$

AR Database: 100 subjects, illumination, expression, occlusion

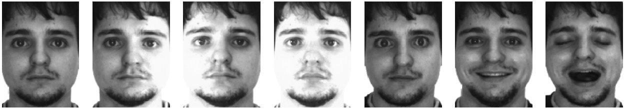


Figure: Training samples for Subject 1.

AR Database: 100 subjects, illumination, expression, occlusion

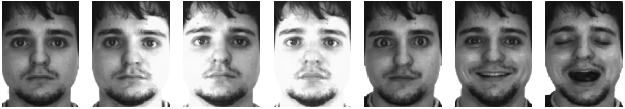
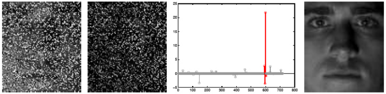
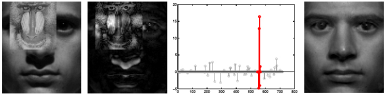


Figure: Training samples for Subject 1.



(a) random corruption



(b) occlusion

AR Database: 100 subjects, illumination, expression, occlusion

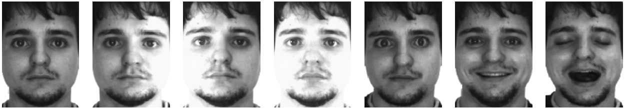
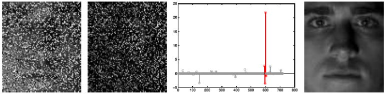
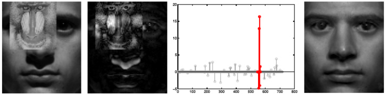


Figure: Training samples for Subject 1.



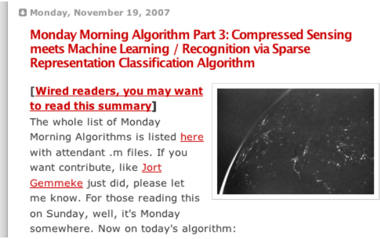
(a) random corruption



(b) occlusion

 sunglasses	 scarves
97.5%	93.5%

Websites



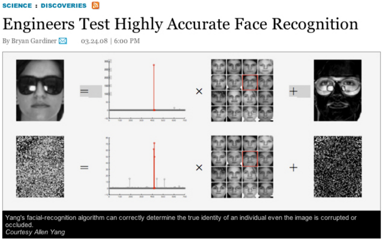
Monday, November 19, 2007

Monday Morning Algorithm Part 3: Compressed Sensing meets Machine Learning / Recognition via Sparse Representation Classification Algorithm

[**Wired readers, you may want to read this summary**]

The whole list of Monday Morning Algorithms is listed here with attendant .m files. If you want contribute, like Jort Gemmeke just did, please let me know. For those reading this on Sunday, well, it's Monday somewhere. Now on today's algorithm:

Figure: Blog: Monday Morning Algorithm



SCIENCE : DISCOVERIES

Engineers Test Highly Accurate Face Recognition

By Bryan Gardiner 03.24.08 | 6:00 PM

Yang's facial-recognition algorithm can correctly determine the true identity of an individual even the image is corrupted or occluded.
Courtesy Allen Yang

Figure: News: Wired.com

Also cited:

- Rice University Compressive Sensing Resources

Conclusion

Estimation of Mixture Subspace Models

1. GPCA
2. Minimum Lossy Coding Length
3. Sparse Representation & ℓ^1-Minimization

Conclusion

Estimation of Mixture Subspace Models

1. GPCA
2. Minimum Lossy Coding Length
3. Sparse Representation & ℓ^1-Minimization

References

- Generalized Principal Component Analysis, **SIAM Review**, 2008.
- Image Segmentation using Mixture Subspace Models, **CVIU**, 2008.
- Robust Face Recognition via Sparse Representation, **PAMI**, 2008.

Conclusion

Estimation of Mixture Subspace Models

1. GPCA
2. Minimum Lossy Coding Length
3. Sparse Representation & ℓ^1-Minimization

References

- Generalized Principal Component Analysis, **SIAM Review**, 2008.
- Image Segmentation using Mixture Subspace Models, **CVIU**, 2008.
- Robust Face Recognition via Sparse Representation, **PAMI**, 2008.

Confluence of Algebra and Statistics

In estimation of mixture (subspace) models,

- Algebra makes statistical algorithms well-conditioned;
- Statistics makes algebraic algorithms robust.

Acknowledgments

Collaborators

- **Berkeley**: Shankar Sastry, Ruzena Bajcsy
- **UIUC**: Yi Ma, Robert Fossum
- **UMich**: Harm Derksen

Funding Support

- ARO MURI: Heterogeneous Sensor Networks (HSNs)
- NSF TRUST Center