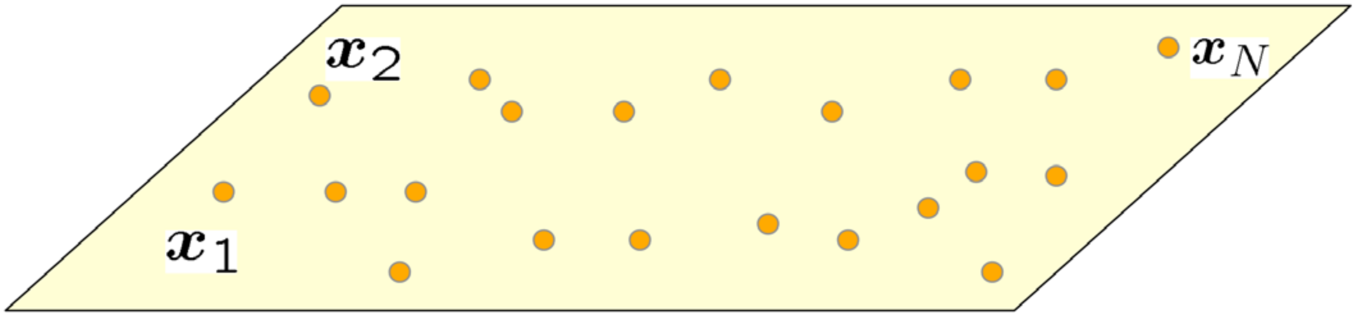
Part I
Generalized Principal Component Analysis

René Vidal
Center for Imaging Science
Institute for Computational Medicine
Johns Hopkins University

Principal Component Analysis (PCA)

- Given a set of points $x_1, x_2, \ldots, x_N$
  - Geometric PCA: find a subspace S passing through them
  - Statistical PCA: find projection directions that maximize the variance



- Solution (Beltrami'1873, Jordan'1874, Hotelling'33, Eckart-Householder-Young'36)

$$U \Sigma V^T = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_N] \in \mathbb{R}^{K \times N}$$

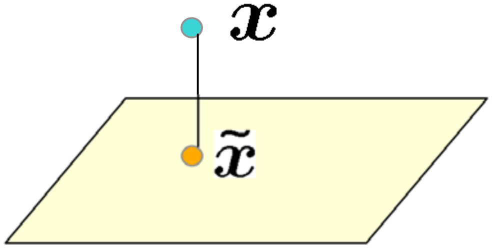
Basis for S

$$\boxed{\mathrm{dim}(S) = \mathrm{rank}(U)}$$

- Applications: data compression, regression, computer vision (eigenfaces), pattern recognition, genomics

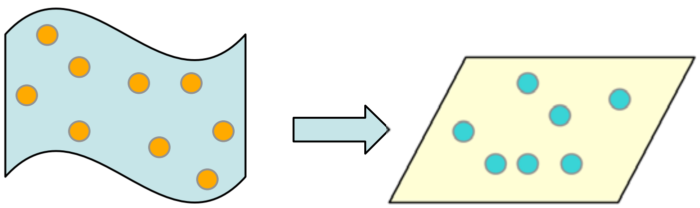
Extensions of PCA

- Higher order SVD (Tucker'66, Davis'02)
- Independent Component Analysis (Common '94)
- Probabilistic PCA (Tipping-Bishop '99)
  - Identify subspace from noisy data
  - Gaussian noise: standard PCA
  - Noise in exponential family (Collins et al.'01)

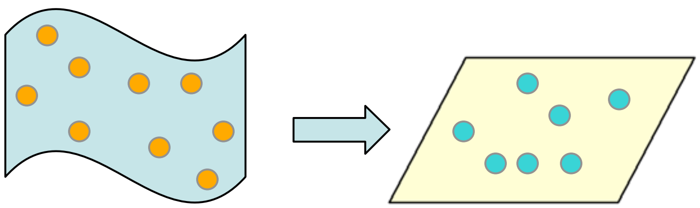


$$x = \tilde{x} + \text{noise}$$

- Nonlinear dimensionality reduction
  - Multidimensional scaling (Torgerson'58)
  - Locally linear embedding (Roweis-Saul '00)
  - Isomap (Tenenbaum '00)



- Nonlinear PCA (Scholkopf-Smola-Muller '98)
  - Identify nonlinear manifold by applying PCA to data embedded in high-dimensional space

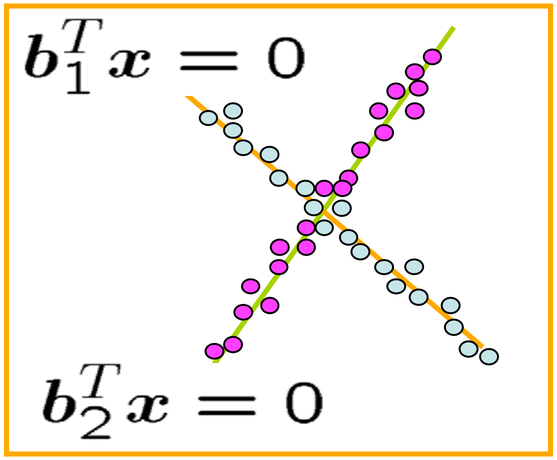


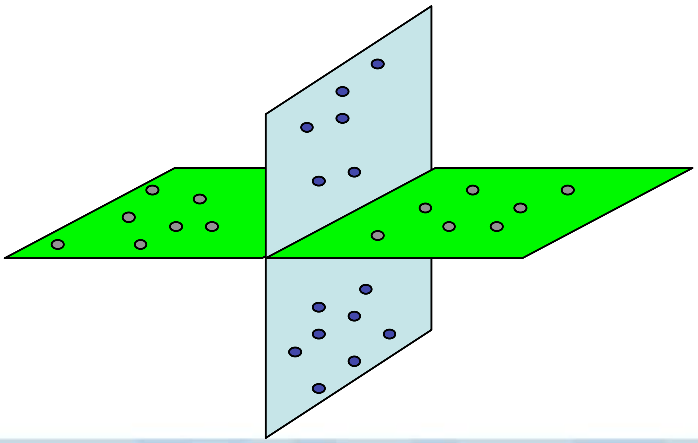
- Principal Curves and Principal Geodesic Analysis (Hastie-Stuetzle'89, Tishbirany '92, Fletcher '04)

Generalized Principal Component Analysis

- Given a set of points lying in multiple subspaces, identify
  - The number of subspaces and their dimensions
  - A basis for each subspace
  - The segmentation of the data points

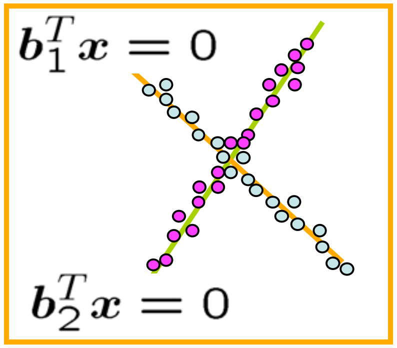
- "Chicken-and-egg" problem
  - Given segmentation, estimate subspaces
  - Given subspaces, segment the data





Prior work on subspace clustering

- Iterative algorithms:
  - K-subspace (Ho et al. '03),
  - RANSAC, subspace selection and growing (Leonardis et al. '02)
- Probabilistic approaches: learn the parameters of a mixture model using e.g. EM
  - Mixtures of PPCA: (Tipping-Bishop '99):
  - Multi-Stage Learning (Kanatani'04)



- Initialization
  - Geometric approaches: 2 planes in R^3 (Shizawa-Maze '91)
  - Factorization approaches: independent subspaces of equal dimension (Boult-Brown '91, Costeira-Kanade '98, Kanatani '01)
  - Spectral clustering based approaches: (Yan-Pollefeys'06)

Basic ideas behind GPCA

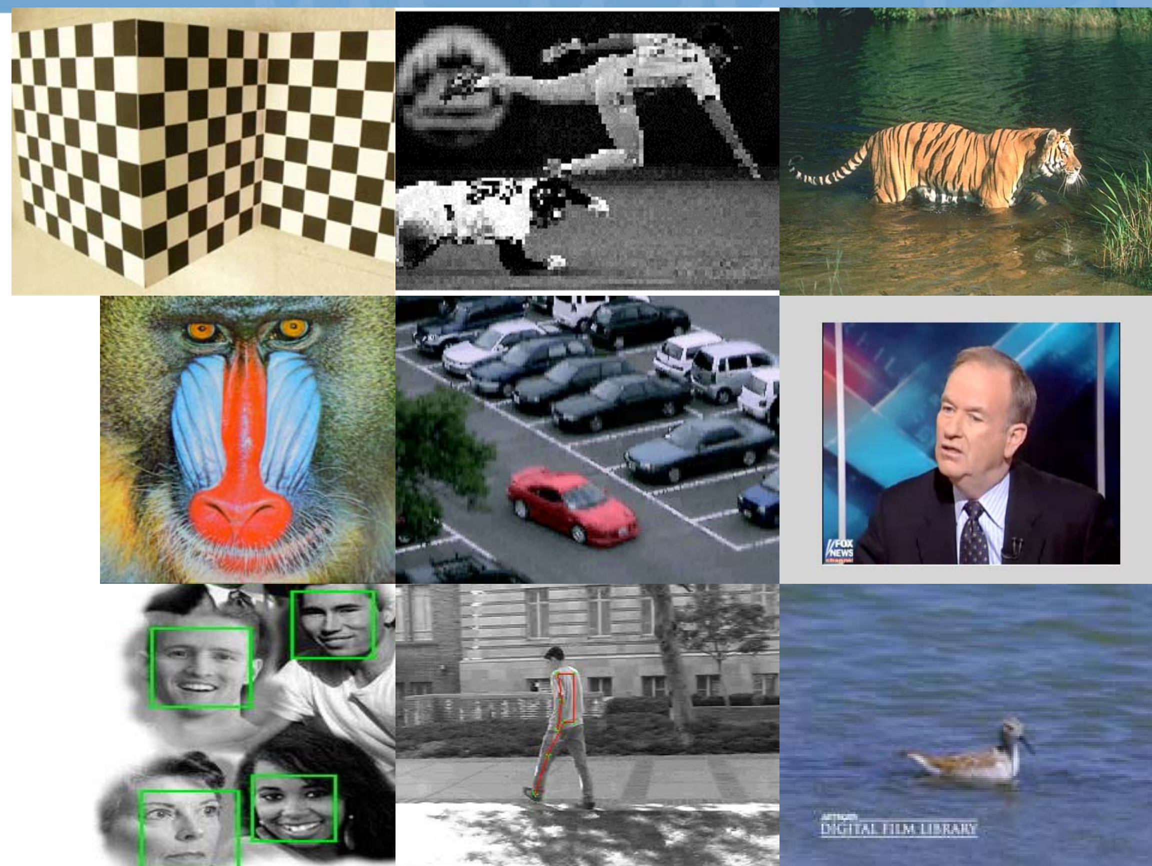
- Towards an analytic solution to subspace clustering
  - Can we estimate ALL models simultaneously using ALL data?
  - When can we do so analytically? In closed form?
  - Is there a formula for the number of models?

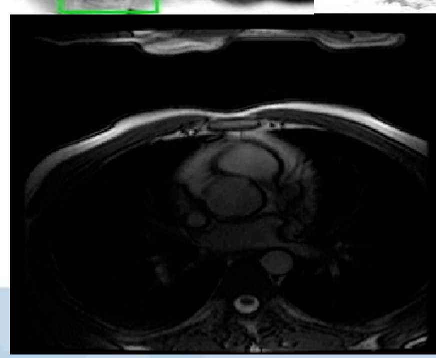
- Will consider the most general case
  - Subspaces of unknown and possibly different dimensions
  - Subspaces may intersect arbitrarily (not only at the origin)

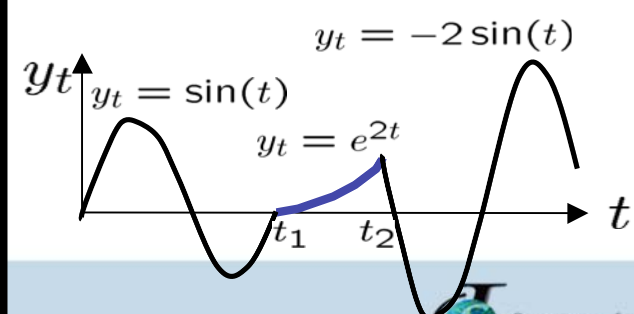
- GPCA is an algebraic geometric approach to data segmentation
  - Number of subspaces = degree of a polynomial
  - Subspace basis = derivatives of a polynomial
  - Subspace clustering is algebraically equivalent to
    - Polynomial fitting
    - Polynomial differentiation

Applications of GPCA in computer vision

- Geometry
  - Vanishing points
- Image compression
- Segmentation
  - Intensity (black-white)
  - Texture
  - Motion (2-D, 3-D)
  - Video (host-guest)
- Recognition
  - Faces (Eigenfaces)
    - Man - Woman
  - Human Gaits
  - Dynamic Textures
    - Water-bird
- Biomedical imaging
- Hybrid systems identification







Introductory example: algebraic clustering in 1D

$x = b_1 \qquad x = b_2$

$$x^2 - (b_1 + b_2)x + b_1 b_2 = 0$$

$$x = b_1 \text{ or } x = b_2$$

$$\underbrace{\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_N^2 & x_N & 1 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} 1 \\ -(b_1 + b_2) \\ b_1 b_2 \end{bmatrix}}_{\boldsymbol{c}} = 0$$

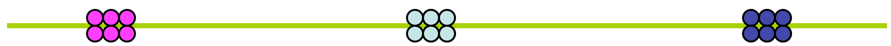
$$(x - b_1)(x - b_2) = 0$$

- Number of groups?

$$\text{rank}(P) = 1: \text{ one group only}$$

$$\text{rank}(P) = 2: \text{ two groups}$$

Introductory example: algebraic clustering in 1D



$$x = b_1 \text{ or } x = b_2 \cdots x = b_n$$

$$p_n(x) = (x-b_1)\cdots(x-b_n) = 0$$

$$p_n(x) = x^n + c_1 x^{n-1} + \cdots c_n = 0$$

$$p_n(x) = \begin{bmatrix} x^n & \cdots & x & 1 \end{bmatrix} \boldsymbol{c} = 0$$

$$P_n \boldsymbol{c} = \underbrace{\begin{bmatrix} x_1^n & \cdots & x_1 & 1 \\ x_2^n & \cdots & x_2 & 1 \\ \vdots & & \vdots & \vdots \\ x_N^n & \cdots & x_N & 1 \end{bmatrix}}_{P_n \in \mathbb{R}^{N \times (n+1}} \boldsymbol{c} = 0$$

- How to compute *n*, *c*, *b*'s?
  - Number of clusters

$$n \doteq \min\{i : rank(P_i) = i\}$$

  - Cluster centers

    Roots of $p_n(x)$

  - Solution is unique if

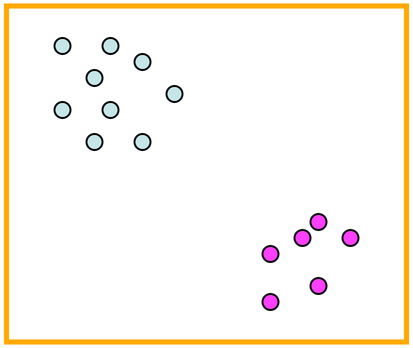
$$N_{points} \geq n_{groups}$$

  - Solution is closed form if

$$n_{groups} \leq 4$$

Introductory example: algebraic clustering in 2D

- What about dimension 2?



$$z = x + iy \in \mathbb{C} \qquad \underbrace{\begin{bmatrix} z_1^n & \cdots & z_1 & 1 \\ z_2^n & \cdots & z_2 & 1 \\ \vdots & & \vdots & \vdots \\ z_N^n & \cdots & z_N & 1 \end{bmatrix}}_{P_n \in \mathbb{C}^{N \times (n+1)}} \boldsymbol{c} = 0$$

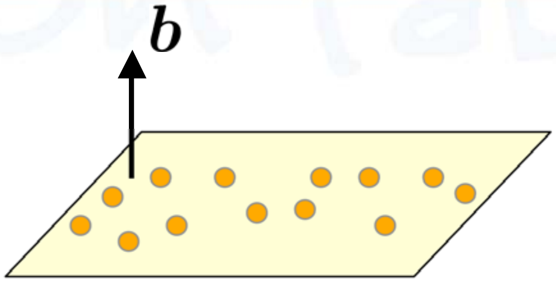
- What about higher dimensions?
  - Complex numbers in higher dimensions?
  - How to find roots of a polynomial of quaternions?

- Instead
  - Project data onto one or two dimensional space
  - Apply same algorithm to projected data

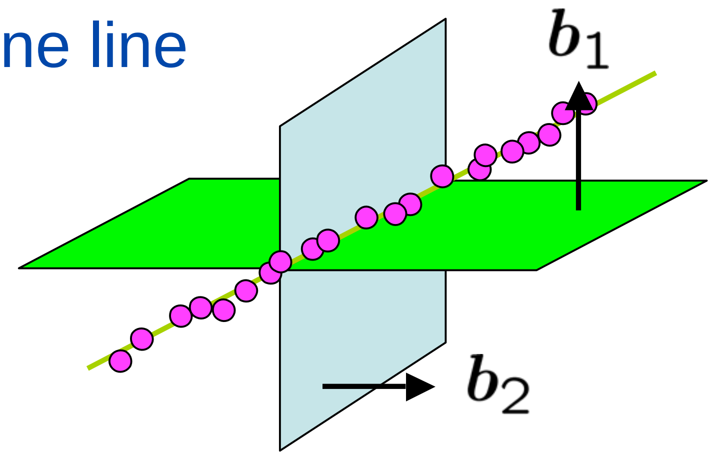
Representing one subspace

- One plane

$$\boldsymbol{b}^T\boldsymbol{x} = b_1x_1 + b_2x_2 + b_3x_3 = 0$$



- One line



$$\boldsymbol{b}_1^T\boldsymbol{x} = b_1x_1 + b_2x_2 + b_3x_3 = 0$$
$$\boldsymbol{b}_2^T\boldsymbol{x} = b_4x_1 + b_5x_2 + b_6x_3 = 0$$

- One subspace can be represented with
  - Set of linear equations
  - Set of polynomials of degree 1

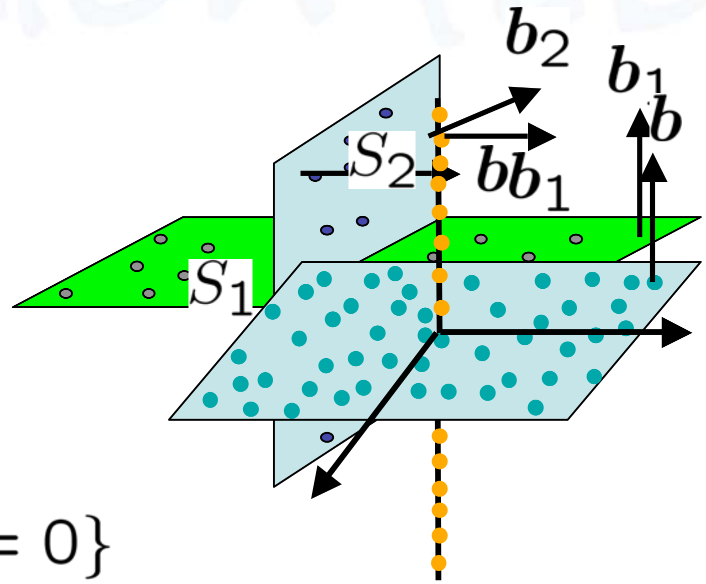
$$S = \{\boldsymbol{x} : B^T\boldsymbol{x} = 0\}$$

Representing *n* subspaces

- Two planes

$$(b_1^T x = 0) \ \boxed{\text{or}} \ (b_2^T x = 0)$$

$$\boxed{p_2(x) = (b_1^T x)(b_2^T x) = 0}$$



- One plane and one line
  - Plane: $S_1 = \{x : b^T x = 0\}$
  - Line: $S_2 = \{x : b_1^T x = b_2^T x = 0\}$

$$S_1 \cup S_2 = \{x : (b^T x = 0) \ \boxed{\text{or}} \ (b_1^T x = b_2^T x = 0\}$$

De Morgan's rule

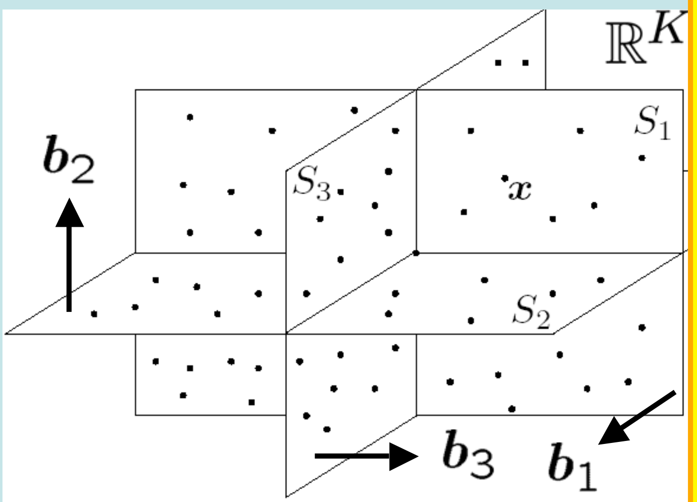
$$S_1 \cup S_2 = \{x : \boxed{(b^T x)(b_1^T x) = 0} \ \boxed{\text{and}} \ \boxed{(b^T x)(b_2^T x) = 0}\}$$

- A union of n subspaces can be represented with a set of homogeneous polynomials of degree n

Fitting polynomials to data points

- Polynomials can be written linearly in terms of the vector of coefficients by using polynomial embedding

$$(\boldsymbol{b}_1^T \boldsymbol{x})(\boldsymbol{b}_2^T \boldsymbol{x}) = c_1 x_1^2 + c_2 x_1 x_2 + c_3 x_2^2 = \boldsymbol{c}^T \nu_n(\boldsymbol{x}) = 0$$

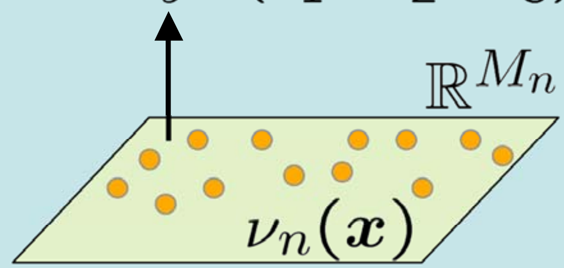


Veronese map

$$\nu_n : \mathbb{R}^K \mapsto \mathbb{R}^{M_n}$$

$$\nu_2 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{pmatrix}$$

$$\boldsymbol{c} = Sym(\boldsymbol{b}_1 \otimes \boldsymbol{b}_2 \otimes \boldsymbol{b}_3)$$



$$M_n = \binom{n + K - 1}{n}$$

- Coefficients of the polynomials can be computed from nullspace of embedded data
  - Solve using least squares
  - N = #data points

$$L_n \boldsymbol{c} = \begin{bmatrix} \nu_n(\boldsymbol{x}_1)^T \\ \vdots \\ \nu_n(\boldsymbol{x}_N)^T \end{bmatrix} \boldsymbol{c} = 0$$

Finding a basis for each subspace

- Case of hyperplanes:
  - Only one polynomial
  - Number of subspaces
  - Basis are normal vectors

$$\boldsymbol{c}^T \nu_n(\boldsymbol{x}) = (\boldsymbol{b}_1^T \boldsymbol{x}) \cdots (\boldsymbol{b}_n^T \boldsymbol{x})$$

$$n = \min\{i : \text{rank}(L_i) = M_i - 1\}$$

$$\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots \boldsymbol{b}_n$$

$$\boldsymbol{c} \in \mathbb{R}^{M_n}$$

$$\boldsymbol{b}_1 \quad \boldsymbol{b}_2 \quad \ldots \quad \boldsymbol{b}_n$$

Polynomial Factorization (GPCA-PFA) [CVPR 2003]
- Find roots of polynomial of degree n in one variable
- Solve $K-2$ linear systems in n variables
- Solution obtained in closed form for $n \leq 4$

- Problems
  - Computing roots may be sensitive to noise
  - The estimated polynomial may not perfectly factor with noisy
  - Cannot be applied to subspaces of different dimensions
    - Polynomials are estimated up to change of basis, hence they may not factor, even with perfect data

Finding a basis for each subspace

$$\boldsymbol{c} \in \mathbb{R}^{M_n}$$

$$\boldsymbol{b}_1 \quad \boldsymbol{b}_2 \quad \ldots \quad \boldsymbol{b}_n$$

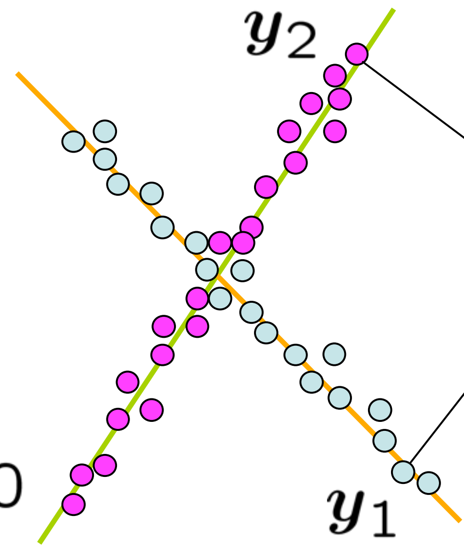
Polynomial Differentiation (GPCA-PDA) [CVPR'04]

$$\boldsymbol{b}_i = Dp_n(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{y}_i} \qquad \boldsymbol{y}_i \in S_i$$

$$p_n(\boldsymbol{x}) = 0$$

$$\boldsymbol{b}_1^T \boldsymbol{x} = 0$$

$$\boldsymbol{b}_2^T \boldsymbol{x} = 0$$



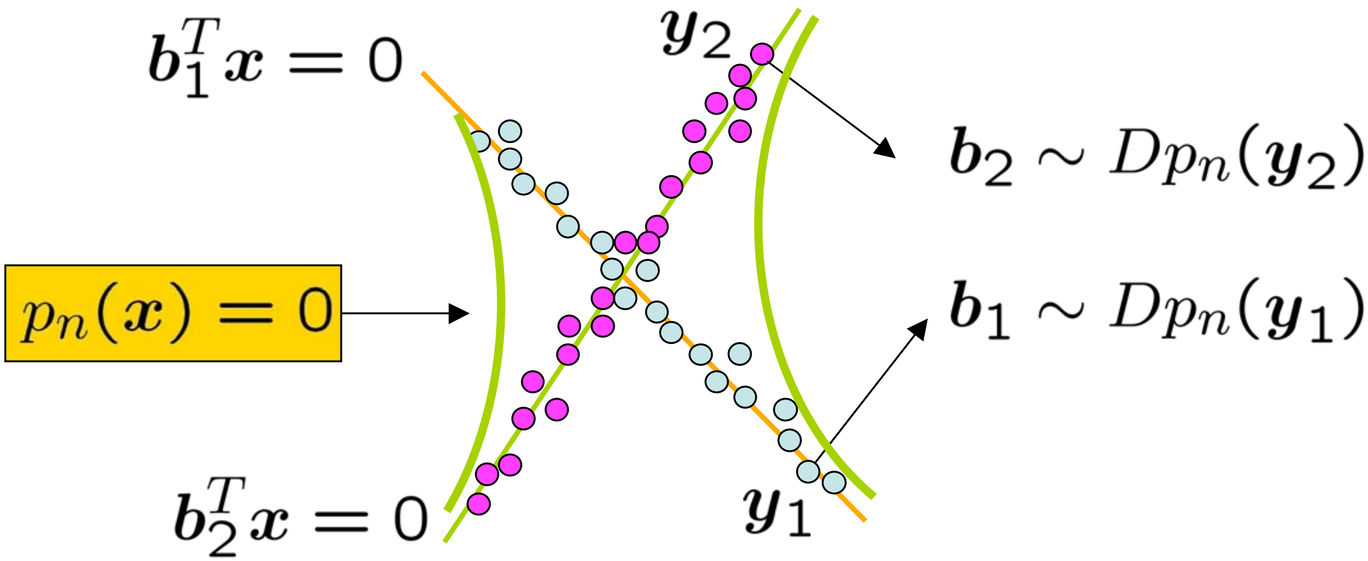
$$\boldsymbol{b}_2 \sim Dp_n(\boldsymbol{y}_2)$$

$$\boldsymbol{b}_1 \sim Dp_n(\boldsymbol{y}_1)$$

- To learn a mixture of subspaces we just need one positive example per class

Choosing one point per subspace

- With noise and outliers
  - Polynomials may not be a perfect union of subspaces



  - Normals can estimated correctly by choosing points optimally
- Distance to closest subspace without knowing segmentation?

$$\|x - \tilde{x}\| = \sqrt{\frac{|p_n(x)|}{\|Dp_n(x)\|}} + O\left(\|x - \tilde{x}\|^2\right)$$

GPCA for hyperplane segmentation

- Coefficients of the polynomial can be computed from null space of embedded data matrix
  - Solve using least squares
  - N = #data points

$$\boxed{L_n \boldsymbol{c} = \begin{bmatrix} \nu_n(\boldsymbol{x}_1)^T \\ \vdots \\ \nu_n(\boldsymbol{x}_N)^T \end{bmatrix} \boldsymbol{c} = 0}$$

- Number of subspaces can be computed from the rank of embedded data matrix

$$\boxed{n = \min\{i : \operatorname{rank}(L_i) = M_i - 1\}}$$

- Normal to the subspaces $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots \boldsymbol{b}_n$ can be computed from the derivatives of the polynomial

$$\begin{tikzcd} & \boldsymbol{c} \in \mathbb{R}^{M_n} \arrow[dl] \arrow[d] \arrow[dr] & \\ \boldsymbol{b}_1 & \boldsymbol{b}_2 \quad \ldots & \boldsymbol{b}_n \end{tikzcd}$$

$$\boxed{\boldsymbol{b}_i = Dp_n(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{y}_i} \qquad \boldsymbol{y}_i \in S_i}$$

GPCA for subspaces of different dimensions

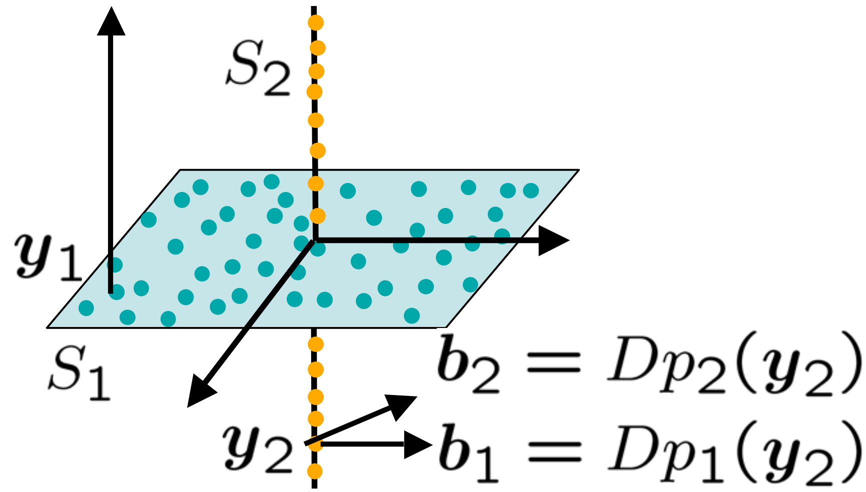
- There are multiple polynomials fitting the data

$$p_1(\boldsymbol{x}) = (\boldsymbol{b}^T\boldsymbol{x})(\boldsymbol{b}_1^T\boldsymbol{x}) = 0$$

$$p_2(\boldsymbol{x}) = (\boldsymbol{b}^T\boldsymbol{x})(\boldsymbol{b}_2^T\boldsymbol{x}) = 0$$

- The derivative of each polynomial gives a different normal vector

$$\boldsymbol{b} = Dp_1(\boldsymbol{y}_1) = Dp_2(\boldsymbol{y}_1)$$



- Can obtain a basis for the subspace by applying PCA to normal vectors

$$\{B_i = PCA(DP_n(\boldsymbol{y}_i))\}_{i=1}^n$$

GPCA for subspaces of different dimensions

- Apply polynomial embedding to projected data

$$\boxed{L_n = [\nu_n(\boldsymbol{x}^1), \ldots, \nu_n(\boldsymbol{x}^N)]^T \in \mathbb{R}^{N \times M_n}}$$

- Obtain multiple subspace model by polynomial fitting

$$\boxed{P_n(\boldsymbol{x}) \doteq [p_{n1}(\boldsymbol{x}), \ldots, p_{n,m_n}(\boldsymbol{x})] \in \mathbb{R}^{1 \times m_n}}$$

  – Solve $L_n \boldsymbol{c} = 0$ to obtain $\{\boldsymbol{c}_{n\ell}\}_{\ell=1}^{m_i} \in \mathrm{null}(L_n)$,
  – Need to know number of subspaces

- Obtain bases & dimensions by polynomial differentiation

$$\boxed{\begin{array}{ll} B_i = PCA(DP_n(\boldsymbol{y}_i)) & i = 1, \ldots, n \\ k_i = K - \mathrm{rank}\big(DP_n(\boldsymbol{y}_i)\big) & i = 1, \ldots, n \end{array}}$$

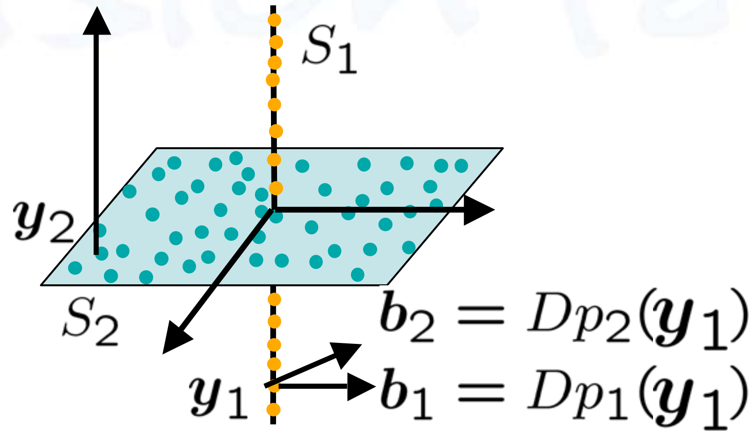
- Optimally choose one point per subspace using distance

$$\boxed{\|\boldsymbol{x}-\tilde{\boldsymbol{x}}\| = \sqrt{P_n(\boldsymbol{x})\Big(DP_n(\boldsymbol{x})^T DP_n(\boldsymbol{x})\Big)^{\dagger} P_n(\boldsymbol{x})^T} + O\Big(\|\boldsymbol{x}-\tilde{\boldsymbol{x}}\|^2\Big)}$$

An example

- Given data lying in the union of the two subspaces

$$S_1 = \{\boldsymbol{x} : x_1 = x_2 = 0\}$$
$$S_2 = \{\boldsymbol{x} : x_3 = 0\}$$



- We can write the union as

$$\begin{aligned} S_1 \cup S_2 &= \{\boldsymbol{x} : (x_1 = x_2 = 0) \vee (x_3 = 0)\} \\ &= \{\boldsymbol{x} : (x_1 = 0 \vee x_3 = 0) \wedge (x_2 = 0 \vee x_3 = 0)\} \\ &= \{\boldsymbol{x} : (x_1 x_3 = 0) \wedge (x_2 x_3 = 0)\}. \end{aligned}$$

- Therefore, the union can be represented with the two polynomials

$$p_1(\boldsymbol{x}) = x_1 x_3 \qquad p_2(\boldsymbol{x}) = x_2 x_3$$

An example

- Can compute polynomials from

$$\begin{bmatrix} x_1^2 & x_1x_2 & x_1x_3 & x_2^2 & x_2x_3 & x_3^2 \\ 0 & 0 & 0 & 0 & 0 & * \\ * & * & 0 & * & 0 & 0 \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_6 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$p_1(\boldsymbol{x}) = x_1x_3 \qquad p_2(\boldsymbol{x}) = x_2x_3$$

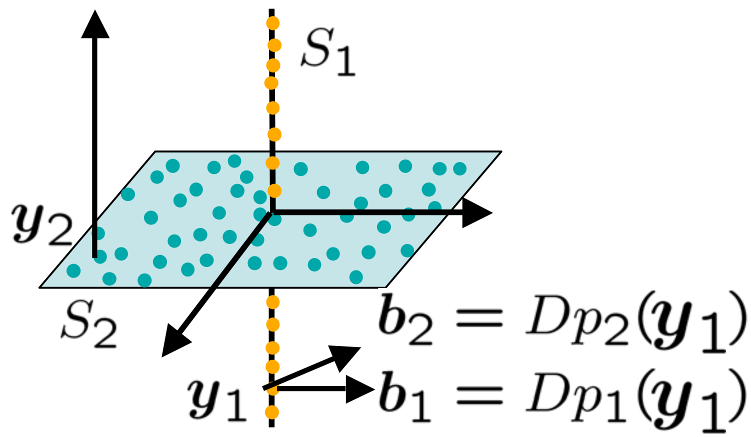
$$S_1 = \{\boldsymbol{x} : x_1 = x_2 = 0\}$$

$$S_2 = \{\boldsymbol{x} : x_3 = 0\}$$

- Can compute normals from

$$[\nabla p_1(\boldsymbol{x}) \ \nabla p_2(\boldsymbol{x})] = \begin{bmatrix} x_3 & 0 \\ 0 & x_3 \\ x_1 & x_2 \end{bmatrix} \implies$$

$$B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \text{ and } B_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$$



Dealing with high-dimensional data

- Minimum number of points
  - K = dimension of ambient space
  - n = number of subspaces

$$M_n(K) = \binom{n+K-1}{n}$$

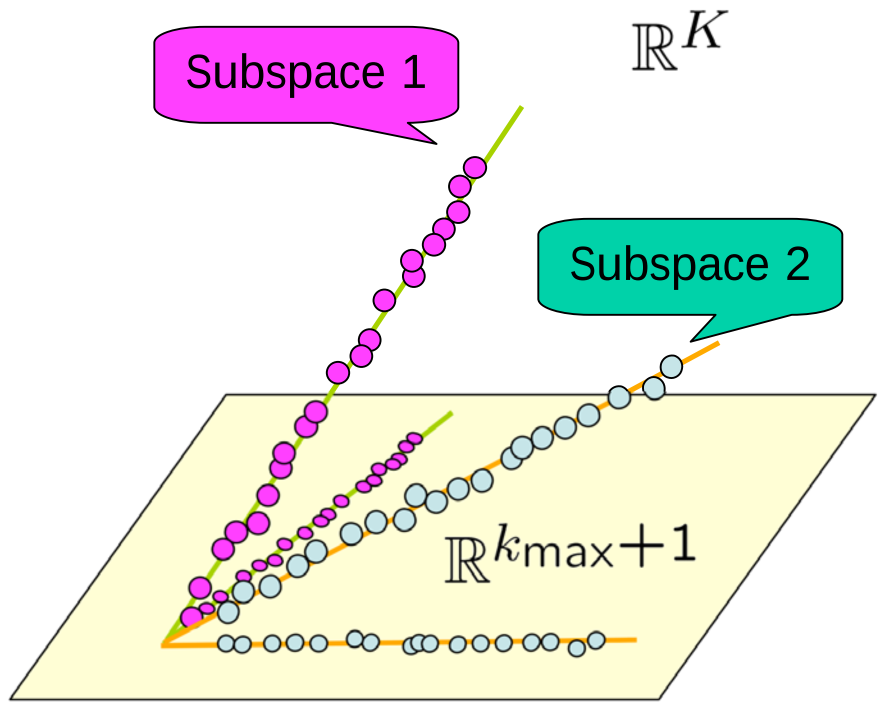
- In practice the dimension of each subspace ki is much smaller than K

$$k_i << K$$

  - Number and dimension of the subspaces is preserved by a linear projection onto a subspace of dimension

$$\max\{k_i\} + 1 << K$$

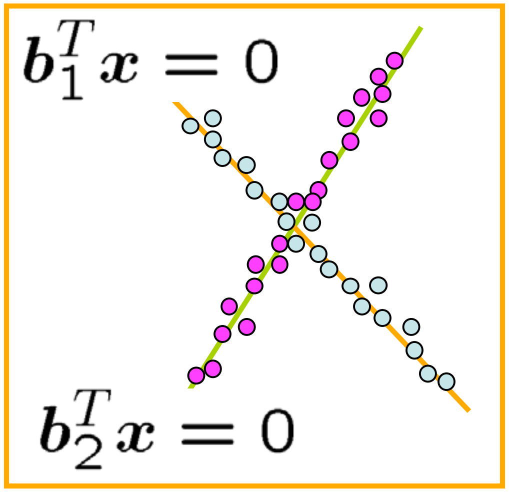
  - Can remove outliers by robustly fitting the subspace



- Open problem: how to choose projection?
  - PCA?

GPCA with spectral clustering

- Spectral clustering
  - Build a similarity matrix between pairs of points
  - Use eigenvectors to cluster data
- How to define a similarity for subspaces?
  - Want points in the same subspace to be close
  - Want points in different subspace to be far

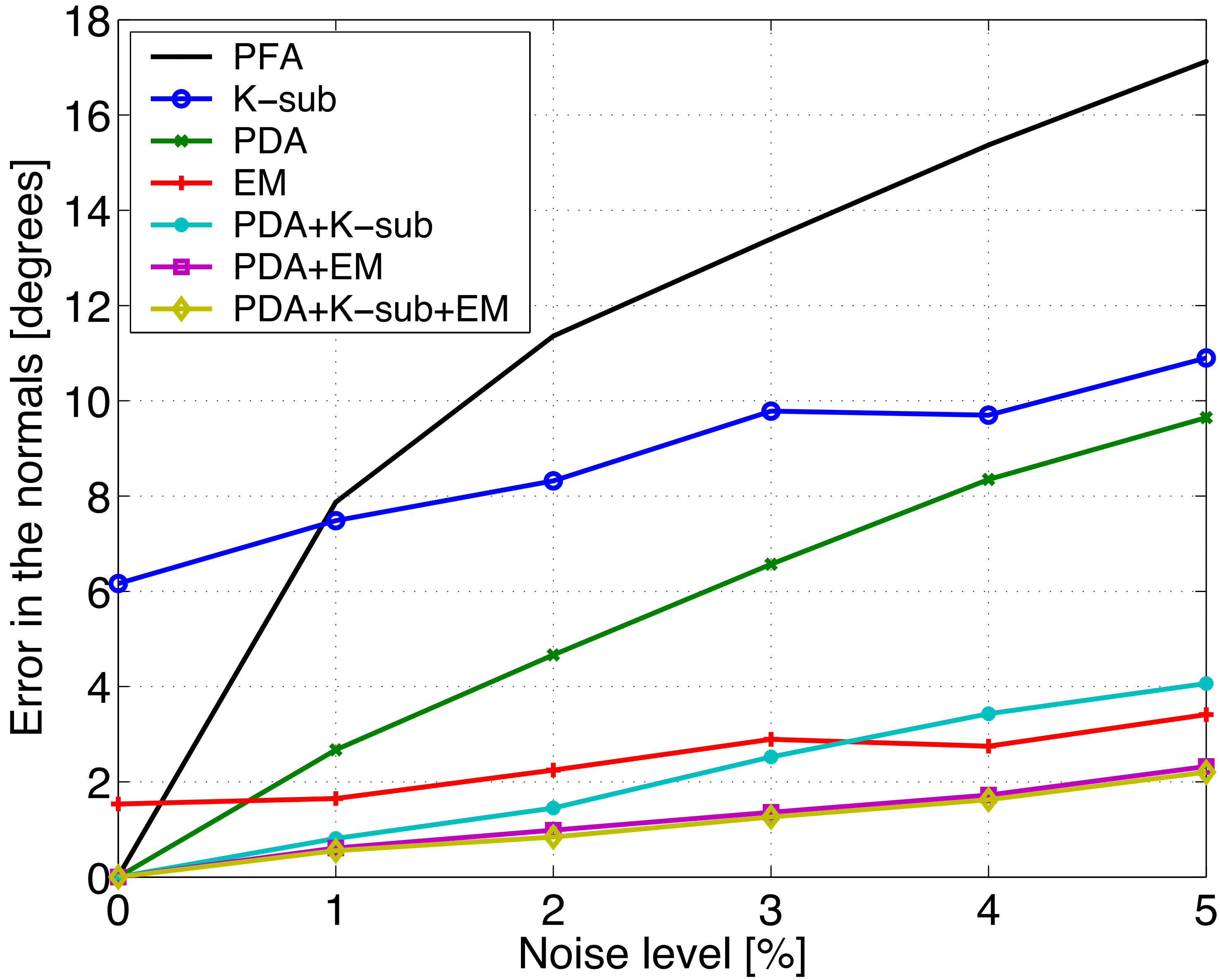


- Use GPCA to get basis

$$B_i = PCA(DP_n(\boldsymbol{y}_i))$$
$$B_j = PCA(DP_n(\boldsymbol{y}_j))$$

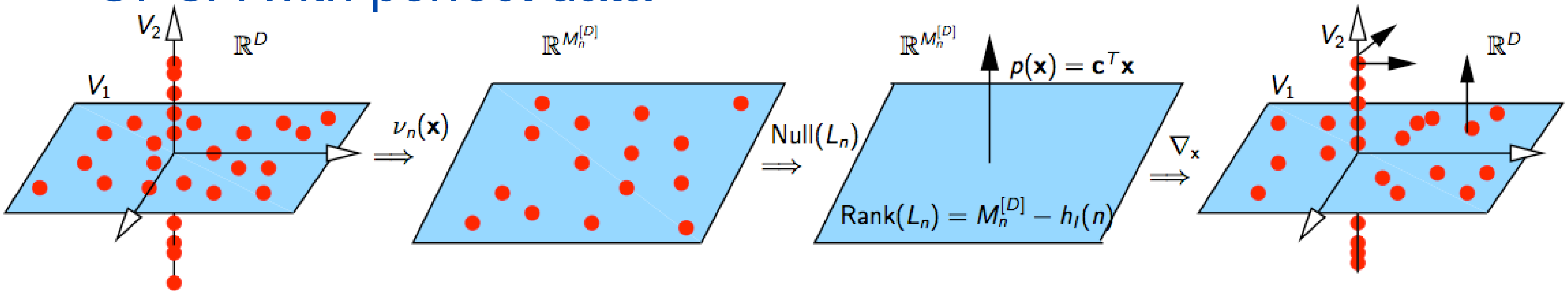
- Distance: subspace angles $\mathcal{D}_{ij} \doteq \langle B_i, B_j \rangle$

Comparison of PFA, PDA, K-sub, EM

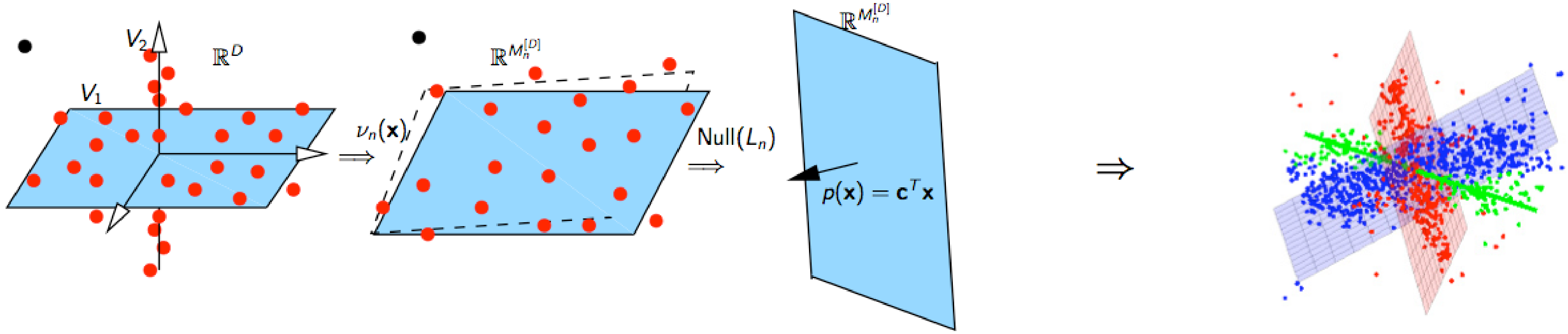


Dealing with outliers

- GPCA with perfect data



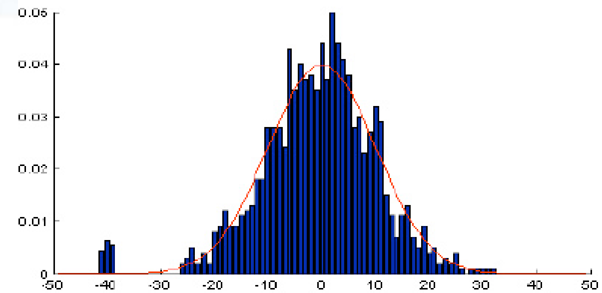
- GPCA with outliers



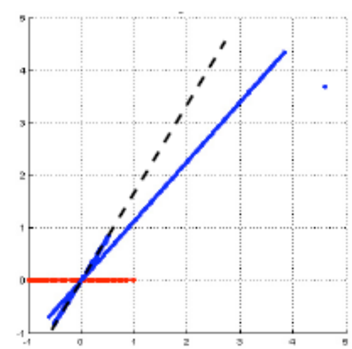
- GPCA fails because PCA fails $\Longrightarrow$ seek a robust estimate of $\mathrm{Null}(L_n)$ where $L_n = [\nu_n(\mathbf{x}_1), \ldots, \nu_n(\mathbf{x}_N)]$.

Three approaches to tackle outliers

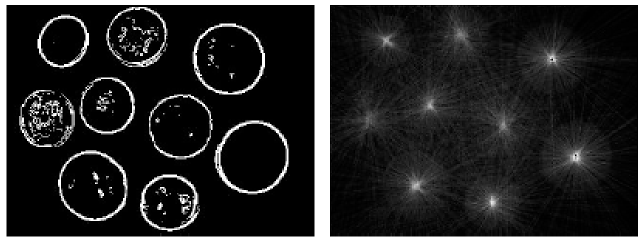
- Probability-based: small-probability samples
  - Probability plots: [Healy 1968, Cox 1968]
  - PCs: [Rao 1964, Ganadesikan & Kettenring 1972]
  - M-estimators: [Huber 1981, Camplbell 1980]
  - Multivariate-trimming (MVT): [Ganadesikan & Kettenring 1972]



- Influence-based: large influence on model parameters
  - Parameter difference with and without a sample: [Hampel et al. 1986, Critchley 1985]



- Consensus-based: not consistent with models of high consensus.
  - Hough: [Ballard 1981, Lowe 1999]
  - RANSAC: [Fischler & Bolles 1981, Torr 1997]
  - Least Median Estimate (LME): [Rousseeuw 1984, Steward 1999]



Center for IMAGING SCIENCE

Robust GPCA

STEP 1: Given the outlier percentage $\alpha\%$, robustify PCA:

- Influence function:
  1. Compute null space $C = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m\}$ for $L_n = [\nu_n(\mathbf{x}_1) \cdots \nu_n(\mathbf{x}_N)]$.
  2. For $\mathbf{x}_i$, compute $C^{(i)}$ for $L_n^{(i)} = [\nu_n(\mathbf{x}_1) \cdots \hat{i} \cdots \nu_n(\mathbf{x}_N)]$.
  3. $I(\mathbf{x}_i) \doteq \langle C, C^{(i)} \rangle$.
  4. Reject top $\alpha\%$ samples with highest influence.

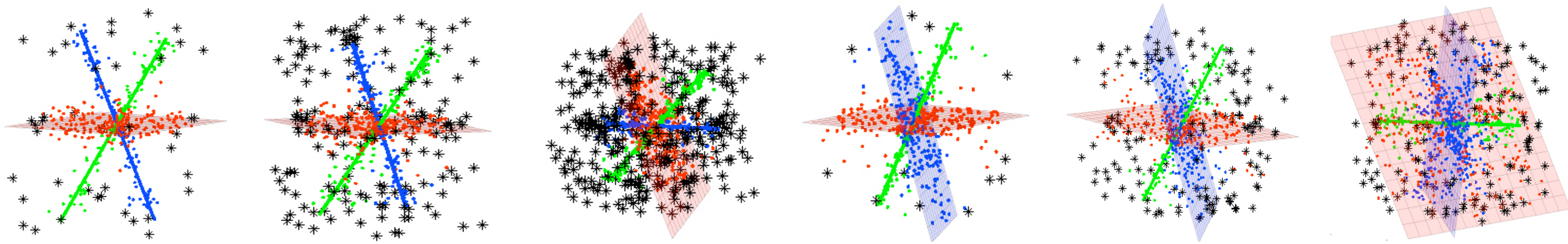
- Multivariate-trimming (MVT):
  Assuming a Gaussian distribution, samples with large *Mahalanobis* distance more likely to be outliers.
  1. Compute a robust mean $\bar{u}$. $\mathbf{v}_i = \mathbf{u}_i - \bar{u}$. $\mathbf{u}_i, \mathbf{v}_i \in \mathbb{R}^{M_n^{[D]}}$
  2. Initialize $\Sigma_0 = I_{M_n^{[D]} \times M_n^{[D]}}$.
  3. In kth iteration, sort $\mathbf{v}_1, \ldots, \mathbf{v}_N$ by the *Mahalanobis* distance: $d_i = \mathbf{v}_i^T \Sigma_{k-1}^{-1} \mathbf{v}_i$.
  4. Update Σ_k from $(100 - \alpha)\%$ samples with smallest distances.
  5. Iteration stops when $\|\Sigma_{k-1} - \Sigma_k\|$ is small.

Center for IMAGING SCIENCE

Robust GPCA

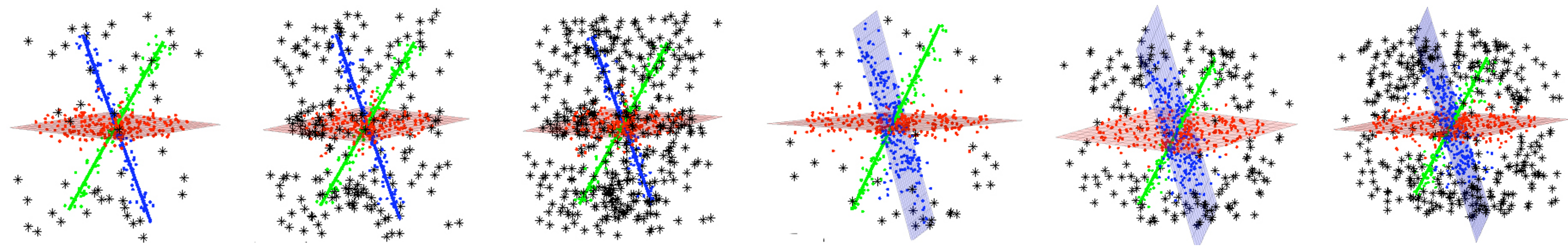
Simulation on Robust GPCA (parameters fixed at τ = 0.3rad and σ = 0.4

- RGPCA – Influence



(e) 12% (f) 32% (g) 48% (h) 12% (i) 32% (j) 48%

- RGPCA - MVT

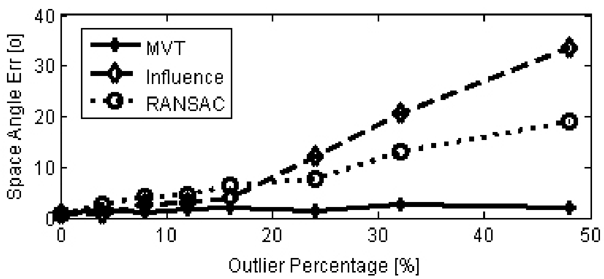


(k) 12% (l) 32% (m) 48% (n) 12% (o) 32% (p) 48%

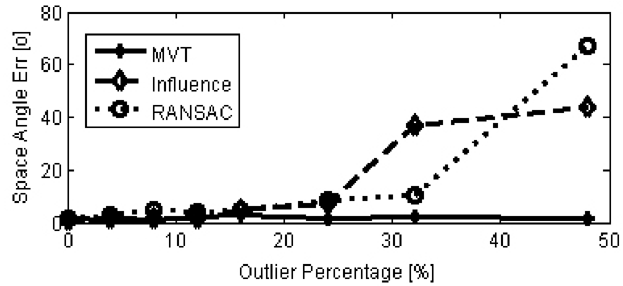
Robust GPCA

Comparison with RANSAC

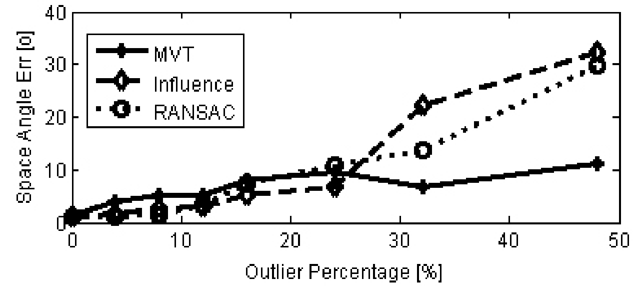
- Accuracy



(q) (2,2,1) in $\Re^3$



(r) (4,2,2,1) in $\Re^5$



(s) (5,5,5) in $\Re^6$

- Speed

Table: Average time of RANSAC and RGPCA with 24% outliers.

Arrangement	(2,2,1) in $\Re^3$	(4,2,2,1) in $\Re^5$	(5,5,5) in $\Re^6$
RANSAC	44s	5.1min	3.4min
MVT	46s	23min	8min
Influence	3min	58min	146min

Summary

- GPCA: algorithm for clustering subspaces
  - Deals with unknown and possibly different dimensions
  - Deals with arbitrary intersections among the subspaces
- Our approach is based on
  - Projecting data onto a low-dimensional subspace
  - Fitting polynomials to projected subspaces
  - Differentiating polynomials to obtain a basis
- Applications in image processing and computer vision
  - Image segmentation: intensity and texture
  - Image compression
  - Face recognition under varying illumination

For more information,

Vision, Dynamics and Learning Lab

@

Johns Hopkins University

http://www.vision.jhu.edu

Thank You!