

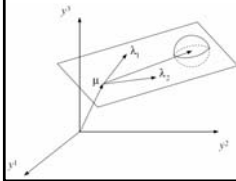
# Machine Learning

10-701/15-781, Fall 2006

## Dimensionality Reduction II Factor Analysis and Metric Learning

Eric Xing

Lecture 20, November 22, 2006  
Reading: Chap. ??, C.B book



1

## Outline

- Probabilistic PCA (brief)
- Factor Analysis (somewhat detail)
- ICA (will skip)
- Distance metric learning from very little side info (a very cool method)

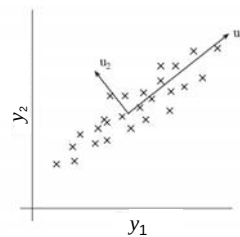


Eric Xing

2

## Recap of PCA

- Popular dimensionality reduction technique
- Project data onto directions of greatest variation



$$\begin{aligned}
 u &= \arg \max \frac{1}{m} \sum_{i=1}^m (\bar{y}_i^T u)^2 \\
 &= \arg \max u^T \left( \frac{1}{m} \sum_{i=1}^m \bar{y}_i \bar{y}_i^T \right) u \\
 &= \arg \max (u^T \text{Cov}(y) u)
 \end{aligned}$$

$$\begin{aligned}
 \bar{x}_i &= \begin{bmatrix} u_1^T \bar{y}_i \\ u_2^T \bar{y}_i \\ \vdots \\ u_q^T \bar{y}_i \end{bmatrix} = U_q^T \bar{y}_i \in \mathbb{R}^q \\
 \bar{y}_i &= U \bar{x}_i
 \end{aligned}$$

- Consequence:

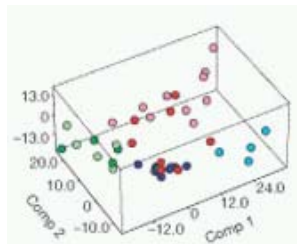
- $x_i$  are uncorrelated such that the covariance matrix  $\frac{1}{m} \sum_{i=1}^m \bar{x}_i \bar{x}_i^T$  is  $\begin{bmatrix} \gamma_1 & & \\ & \ddots & \\ & & \gamma_q \end{bmatrix}$
- Truncation error  $\Sigma_y = \sum_{k=1}^K \gamma_k (u_k u_k^T) \approx \sum_{k=1}^q \gamma_k (u_k u_k^T) = \Sigma_x$

Eric Xing

3

## Recap of PCA

- Popular dimensionality reduction technique
- Project data onto directions of greatest variation



Useful tool for visualising patterns and clusters within the data set, but ...

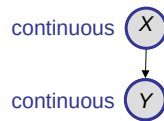
Need centering

Does not explicitly model data noise

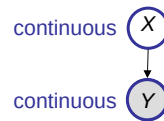
Eric Xing

4

## Probabilistic Interpretation?



regression



?

Eric Xing

5

## Probabilistic PCA



- PCA can be cast as a probabilistic model

$$y_n = \Lambda x_n + \mu + \varepsilon_n \quad \varepsilon_n \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$$

with  $q$ -dimensional latent variables  $x_n \sim \mathcal{N}(\mathbf{0}, I)$

- The resulting data distribution is

$$y_n \sim \mathcal{N}(\mu, \Lambda \Lambda^T + \sigma^2 I)$$

- Maximum likelihood solution is equivalent to PCA

$$\mu^{ML} = \frac{1}{N} \sum_n y_n \quad \Lambda^{ML} = U_q (\Gamma_q - \sigma^2 I)^{1/2}$$

Diagonal  $\Gamma_q$  contains the top  $q$  sample covariance eigen-values and  $U_q$  contains associated eigenvectors

Tipping and Bishop, *J. Royal Stat. Soc.* **6**, 611 (1999).

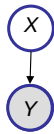
Eric Xing

6

# Factor analysis



- An unsupervised linear regression model

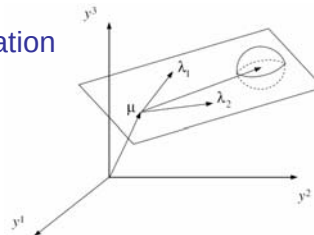


$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{0}, I)$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \mu + \Lambda \mathbf{x}, \Psi)$$

where  $\Lambda$  is called a **factor loading matrix**, and  $\Psi$  is diagonal.

- Geometric interpretation



- To generate data, first generate a point within the manifold then add noise. Coordinates of point are components of latent variable.

Eric Xing

7

# Relationship between PCA and FA



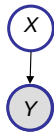
- **Probabilistic** PCA is equivalent to factor analysis with **equal** noise for every dimension, i.e.,  $\varepsilon_n \sim$  **isotropic Gaussian**  $\mathcal{N}(\mathbf{0}, \sigma^2 I)$
- In factor analysis  $\varepsilon_n \sim \mathcal{N}(\mathbf{0}, \Psi)$  for a diagonal covariance matrix  $\Psi$
- An iterative algorithm (eg. EM) is required to find parameters **if precisions are not known in advance**

Eric Xing

8

# Factor analysis

- An unsupervised linear regression model

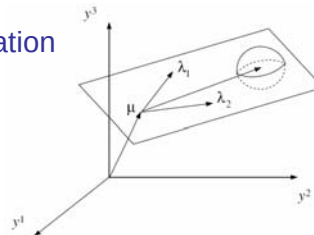


$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{0}, I)$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \mu + \Lambda \mathbf{x}, \Psi)$$

where  $\Lambda$  is called a factor loading matrix, and  $\Psi$  is diagonal.

- Geometric interpretation



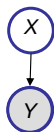
- To generate data, first generate a point within the manifold then add noise. Coordinates of point are components of latent variable.

Eric Xing

9

# Marginal data distribution

- A marginal Gaussian (e.g.,  $p(\mathbf{x})$ ) times a conditional Gaussian (e.g.,  $p(\mathbf{y}|\mathbf{x})$ ) is a joint Gaussian
- Any marginal (e.g.,  $p(\mathbf{y})$ ) of a joint Gaussian (e.g.,  $p(\mathbf{x}, \mathbf{y})$ ) is also a Gaussian
  - Since the marginal is Gaussian, we can determine it by just computing its mean and variance. (Assume noise uncorrelated with data.)



$$\begin{aligned} E[\mathbf{Y}] &= E[\mu + \Lambda \mathbf{X} + \mathbf{W}] \quad \text{where } \mathbf{W} \sim \mathcal{N}(\mathbf{0}, \Psi) \\ &= \mu + \Lambda E[\mathbf{X}] + E[\mathbf{W}] \\ &= \mu + \mathbf{0} + \mathbf{0} = \mu \\ \text{Var}[\mathbf{Y}] &= E[(\mathbf{Y} - \mu)(\mathbf{Y} - \mu)^T] \\ &= E[(\mu + \Lambda \mathbf{X} + \mathbf{W} - \mu)(\mu + \Lambda \mathbf{X} + \mathbf{W} - \mu)^T] \\ &= E[(\Lambda \mathbf{X} + \mathbf{W})(\Lambda \mathbf{X} + \mathbf{W})^T] \\ &= \Lambda E[\mathbf{X} \mathbf{X}^T] \Lambda^T + E[\mathbf{W} \mathbf{W}^T] \\ &= \Lambda \Lambda^T + \Psi \end{aligned}$$

Eric Xing

10

# FA = Constrained-Covariance Gaussian

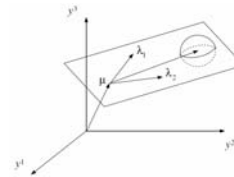


- Marginal density for factor analysis ( $\mathbf{y}$  is  $p$ -dim,  $\mathbf{x}$  is  $k$ -dim):

$$p(\mathbf{y} | \theta) = \mathcal{N}(\mathbf{y}; \mu, \Lambda \Lambda^T + \Psi)$$

- So the effective covariance is the low-rank outer product of two long skinny matrices plus a diagonal matrix:

$$\text{Cov}[\mathbf{y}] = \Lambda \Lambda^T + \Psi$$



- In other words, factor analysis is just a constrained Gaussian model. (If  $\Psi$  were not diagonal then we could model any Gaussian and it would be pointless.)

Eric Xing

11

# Review: A primer to multivariate Gaussian



- Multivariate Gaussian density:

$$p(\mathbf{x} | \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right\}$$

- A joint Gaussian:

$$p\left(\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \middle| \mu, \Sigma\right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \middle| \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

- How to write down  $p(\mathbf{x}_1)$ ,  $p(\mathbf{x}_1|\mathbf{x}_2)$  or  $p(\mathbf{x}_2|\mathbf{x}_1)$  using the block elements in  $\mu$  and  $\Sigma$ ?

- Formulas to remember:

$$p(\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_2 | \mathbf{m}_2^m, \mathbf{V}_2^m)$$

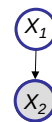
$$p(\mathbf{x}_1|\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_1 | \mathbf{m}_{1|2}, \mathbf{V}_{1|2})$$

$$\mathbf{m}_2^m = \mu_2$$

$$\mathbf{m}_{1|2} = \mu_1 + \Sigma_{12} \Sigma_{22}^{-1} (\mathbf{x}_2 - \mu_2)$$

$$\mathbf{V}_2^m = \Sigma_{22}$$

$$\mathbf{V}_{1|2} = \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}$$



Eric Xing

12

## Review: Some matrix algebra



- Trace and derivatives

$$\text{tr}[A] \stackrel{\text{def}}{=} \sum_i a_{ii}$$

- Cyclical permutations

$$\text{tr}[ABC] = \text{tr}[CAB] = \text{tr}[BCA]$$

- Derivatives

$$\frac{\partial}{\partial A} \text{tr}[BA] = B^T$$

$$\frac{\partial}{\partial A} \text{tr}[x^T A x] = \frac{\partial}{\partial A} \text{tr}[x x^T A] = x x^T$$

- Determinants and derivatives

$$\frac{\partial}{\partial A} \log|A| = A^{-T}$$

Eric Xing

13

## FA joint distribution



- Model

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{0}, I)$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \mu + \Lambda \mathbf{x}, \Psi)$$

- Covariance between  $\mathbf{x}$  and  $\mathbf{y}$

$$\begin{aligned} \text{Cov}[\mathbf{X}, \mathbf{Y}] &= E[(\mathbf{X} - \mathbf{0})(\mathbf{Y} - \mu)^T] = E[\mathbf{X}(\mu + \Lambda \mathbf{X} + \mathbf{W} - \mu)^T] \\ &= E[\mathbf{X} \mathbf{X}^T \Lambda^T + \mathbf{X} \mathbf{W}^T] \\ &= \Lambda^T \end{aligned}$$

- Hence the joint distribution of  $\mathbf{x}$  and  $\mathbf{y}$ :

$$p\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \middle| \begin{bmatrix} \mathbf{0} \\ \mu \end{bmatrix}, \begin{bmatrix} I & \Lambda^T \\ \Lambda & \Lambda \Lambda^T + \Psi \end{bmatrix}\right)$$

- Assume noise is uncorrelated with data or latent variables.

Eric Xing

14

## Inference in Factor Analysis



- Apply the Gaussian conditioning formulas to the joint distribution we derived above, where

$$\begin{aligned}\Sigma_{11} &= I \\ \Sigma_{12} &= \Sigma_{12}^T = \Lambda^T \\ \Sigma_{22} &= (\Lambda\Lambda^T + \Psi)\end{aligned}$$

we can now derive the posterior of the latent variable  $\mathbf{x}$  given observation  $\mathbf{y}$ ,  $p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x} | \mathbf{m}_{1|2}, \mathbf{V}_{1|2})$ , where

$$\begin{aligned}\mathbf{m}_{1|2} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{y} - \mu_2) & \mathbf{V}_{1|2} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \\ &= \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}(\mathbf{y} - \mu) & &= I - \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}\Lambda\end{aligned}$$

Applying the matrix inversion lemma  $(G)^{-1} = G^{-1} + G^{-1}F(G^{-1} + F)^{-1}F$

$$\Rightarrow \mathbf{V}_{1|2} = (I + \Lambda^T\Psi^{-1}\Lambda)^{-1} \quad \mathbf{m}_{1|2} = \mathbf{V}_{1|2}\Lambda^T\Psi^{-1}(\mathbf{y} - \mu)$$

- Here we only need to invert a matrix of size  $|\mathbf{x}| \times |\mathbf{x}|$ , instead of  $|\mathbf{y}| \times |\mathbf{y}|$ .

Eric Xing

15

## Geometric interpretation: inference is linear projection

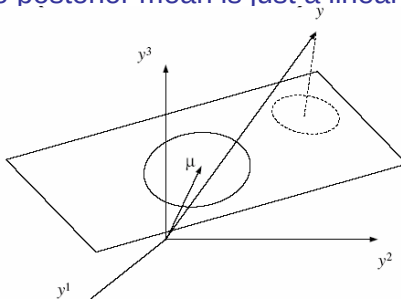


- The posterior is:

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}; \mathbf{m}_{1|2}, \mathbf{V}_{1|2})$$

$$\mathbf{V}_{1|2} = (I + \Lambda^T\Psi^{-1}\Lambda)^{-1} \quad \mathbf{m}_{1|2} = \mathbf{V}_{1|2}\Lambda^T\Psi^{-1}(\mathbf{y} - \mu)$$

- Posterior covariance does not depend on observed data  $\mathbf{y}$ !
- Computing the posterior mean is just a linear operation:



Eric Xing

16



## EM for Factor Analysis



- Incomplete data log likelihood function (marginal density of  $y$ )

$$\begin{aligned}\mathcal{L}(\theta, \mathcal{D}) &= -\frac{N}{2} \log |\Lambda \Lambda^T + \Psi| - \frac{1}{2} \sum_n (\mathbf{y}_n - \mu)^T (\Lambda \Lambda^T + \Psi)^{-1} (\mathbf{y}_n - \mu) \\ &= -\frac{N}{2} \log |\Lambda \Lambda^T + \Psi| - \frac{1}{2} \text{tr}[(\Lambda \Lambda^T + \Psi)^{-1} \mathbf{S}], \quad \text{where } \mathbf{S} = \sum_n (\mathbf{y}_n - \mu)(\mathbf{y}_n - \mu)^T\end{aligned}$$

- Estimating  $\mu$  is trivial:  $\hat{\mu}^{ML} = \frac{1}{N} \sum_n \mathbf{y}_n$
- Parameters  $\Lambda$  and  $\Psi$  are coupled nonlinearly in log-likelihood

- Complete log likelihood

$$\begin{aligned}\mathcal{L}_c(\theta, \mathcal{D}) &= \sum_n \log p(\mathbf{x}_n, \mathbf{y}_n) = \sum_n \log p(\mathbf{x}_n) + \log p(\mathbf{y}_n | \mathbf{x}_n) \\ &= -\frac{N}{2} \log |I| - \frac{1}{2} \sum_n \mathbf{x}_n^T \mathbf{x}_n - \frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_n (\mathbf{y}_n - \Lambda \mathbf{x}_n)^T \Psi^{-1} (\mathbf{y}_n - \Lambda \mathbf{x}_n) \\ &= -\frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_n \text{tr}[\mathbf{x}_n \mathbf{x}_n^T] - \frac{N}{2} \text{tr}[\mathbf{S} \Psi^{-1}], \quad \text{where } \mathbf{S} = \frac{1}{N} \sum_n (\mathbf{y}_n - \Lambda \mathbf{x}_n)(\mathbf{y}_n - \Lambda \mathbf{x}_n)^T\end{aligned}$$

Eric Xing

17

## E-step for Factor Analysis



- Compute  $\langle \mathcal{L}_c(\theta, \mathcal{D}) \rangle_{p(\mathbf{x}|\mathbf{y})}$

$$\begin{aligned}\langle \mathcal{L}_c(\theta, \mathcal{D}) \rangle &= -\frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_n \text{tr}[\langle \mathbf{x}_n \mathbf{x}_n^T \rangle] - \frac{N}{2} \text{tr}[\langle \mathbf{S} \rangle \Psi^{-1}] \\ \langle \mathbf{S} \rangle &= \frac{1}{N} \sum_n (\mathbf{y}_n \mathbf{y}_n^T - \mathbf{y}_n \langle \mathbf{x}_n^T \rangle \Lambda^T - \Lambda \langle \mathbf{x}_n^T \rangle \mathbf{y}_n^T + \Lambda \langle \mathbf{x}_n \mathbf{x}_n^T \rangle \Lambda^T) \\ \langle \mathbf{x}_n \rangle &= E[\mathbf{x}_n | \mathbf{y}_n] \\ \langle \mathbf{x}_n \mathbf{x}_n^T \rangle &= \text{Var}[\mathbf{x}_n | \mathbf{y}_n] + E[\mathbf{x}_n | \mathbf{y}_n] E[\mathbf{x}_n | \mathbf{y}_n]^T\end{aligned}$$

- Recall that we have derived:

$$\begin{aligned}\mathbf{V}_{1|2} &= (I + \Lambda^T \Psi^{-1} \Lambda)^{-1} & \mathbf{m}_{1|2} &= \mathbf{V}_{1|2} \Lambda^T \Psi^{-1} (\mathbf{y} - \mu) \\ \Rightarrow \langle \mathbf{x}_n \rangle &= \mathbf{m}_{\mathbf{x}_n | \mathbf{y}_n} = \mathbf{V}_{1|2} \Lambda^T \Psi^{-1} (\mathbf{y}_n - \mu) & \text{and} & \quad \langle \mathbf{x}_n \mathbf{x}_n^T \rangle = \mathbf{V}_{1|2} + \mathbf{m}_{\mathbf{x}_n | \mathbf{y}_n} \mathbf{m}_{\mathbf{x}_n | \mathbf{y}_n}^T\end{aligned}$$

Eric Xing

18

## M-step for Factor Analysis

- Take the derivatives of the expected complete log likelihood wrt. parameters.
  - Using the trace and determinant derivative rules:

$$\begin{aligned}\frac{\partial}{\partial \Psi^{-1}} \langle \ell_c \rangle &= \frac{\partial}{\partial \Psi^{-1}} \left( -\frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_n \text{tr}[\langle X_n X_n^T \rangle] - \frac{N}{2} \text{tr}[\langle S \rangle \Psi^{-1}] \right) \\ &= \frac{N}{2} \Psi - \frac{N}{2} \langle S \rangle \quad \Rightarrow \quad \Psi^{t+1} = \langle S \rangle\end{aligned}$$

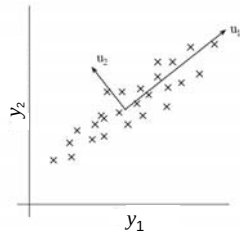
$$\begin{aligned}\frac{\partial}{\partial \Lambda} \langle \ell_c \rangle &= \frac{\partial}{\partial \Lambda} \left( -\frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_n \text{tr}[\langle X_n X_n^T \rangle] - \frac{N}{2} \text{tr}[\langle S \rangle \Psi^{-1}] \right) = -\frac{N}{2} \Psi^{-1} \frac{\partial}{\partial \Lambda} \langle S \rangle \\ &= -\frac{N}{2} \Psi^{-1} \frac{\partial}{\partial \Lambda} \left( \frac{1}{N} \sum_n (\gamma_n \gamma_n^T - \gamma_n \langle X_n^T \rangle \Lambda^T - \Lambda \langle X_n^T \rangle \gamma_n^T + \Lambda \langle X_n X_n^T \rangle \Lambda^T) \right) \\ &= \Psi^{-1} \sum_n \gamma_n \langle X_n^T \rangle - \Psi^{-1} \Lambda \sum_n \langle X_n X_n^T \rangle \quad \Rightarrow \quad \Lambda^{t+1} = \left( \sum_n \gamma_n \langle X_n^T \rangle \right) \left( \sum_n \langle X_n X_n^T \rangle \right)^{-1}\end{aligned}$$

Eric Xing

19

## Comparison of PCA and FA

- PCA

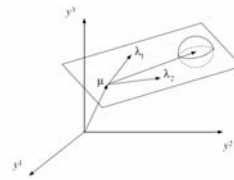


$$y_n = U x_n$$

$$u = \arg \max (u^T \text{Cov}(y) u)$$

$$\bar{x}_i = \begin{bmatrix} u_1^T \bar{y}_i \\ u_2^T \bar{y}_i \\ \vdots \\ u_k^T \bar{y}_i \end{bmatrix} = U_q^T \bar{y}_i \in \mathbb{R}^q$$

- FA



$$y_n = \Lambda x_n + \mu + \varepsilon_n$$

$$\varepsilon_n \sim \mathcal{N}(\mathbf{0}, \Psi)$$

$$\langle X_n \rangle = \mathbf{m}_{x_n|y_n} = \mathbf{V}_{12} \Lambda^T \Psi^{-1} (y_n - \mu)$$

$$\text{and} \quad \langle X_n X_n^T \rangle = \mathbf{V}_{12} + \mathbf{m}_{x_n|y_n} \mathbf{m}_{x_n|y_n}^T$$

$$\Lambda^{t+1} = \left( \sum_n \gamma_n \langle X_n^T \rangle \right) \left( \sum_n \langle X_n X_n^T \rangle \right)^{-1}$$

Eric Xing

20

# Comparison of PCA and FA

## • PCA

$$u = \arg \max (u^T \text{Cov}(y) u)$$

$$\bar{x}_i = \begin{bmatrix} u_1^T \bar{y}_i \\ u_2^T \bar{y}_i \\ \vdots \\ u_k^T \bar{y}_i \end{bmatrix} = U_q^T \bar{y}_i \in \mathbb{R}^q$$

- SVD on a  $K \times K$  matrix
- Covariant under rotation:  $Ay$
- Principle axis can be found incrementally

## • FA

$$\langle X_n \rangle = \mathbf{m}_{x_n|y_n} = \mathbf{V}_{12} \Lambda^T \Psi^{-1} (y_n - \mu)$$

$$\text{and } \langle X_n X_n^T \rangle = \mathbf{V}_{12} + \mathbf{m}_{x_n|y_n} \mathbf{m}_{x_n|y_n}^T$$

$$\Lambda^{*+1} = \left( \sum_n y_n \langle X_n^T \rangle \right) \left( \sum_n \langle X_n X_n^T \rangle \right)^{-1}$$

$$\Psi^{*+1} = \langle S \rangle$$

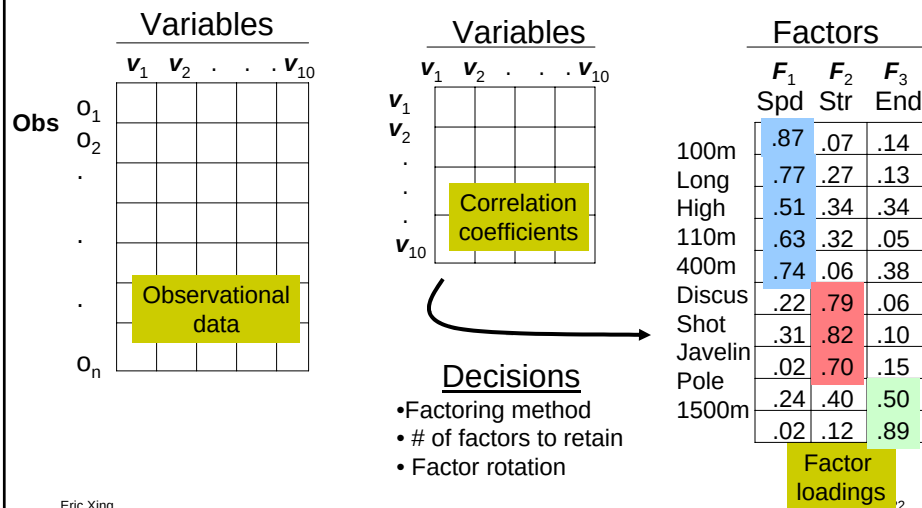
- Invert a  $q \times q$  matrix
- Covariant under rescaling:  $\text{diag}(\alpha)y$
- Neither of the factors found by a two-factor model is necessarily the same as that found by a single factor model, and ...

Eric Xing

21

# Example:

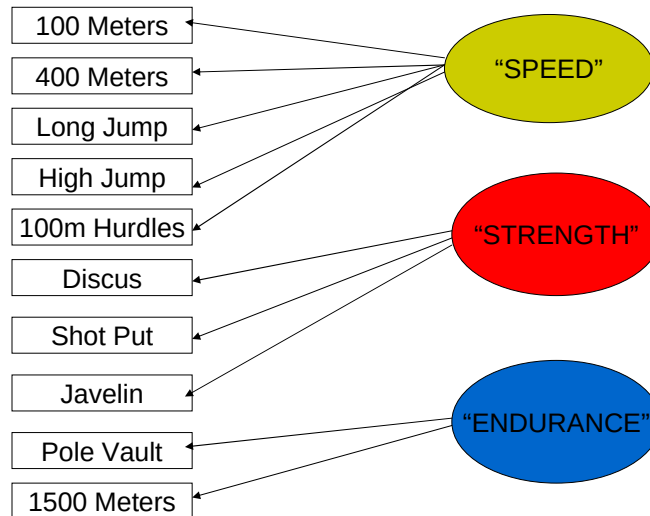
Original data matrix  $\rightarrow$  Correlation matrix  $\rightarrow$  Factor matrix



Eric Xing

22

## Decathlon example



Eric Xing

23

## Model Invariance and Identifiability

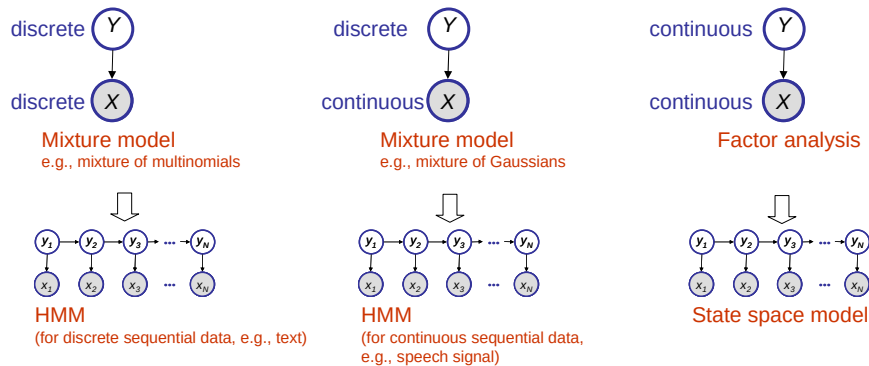
- There is *degeneracy* in the FA model.
- Since  $\Lambda$  only appears as outer product  $\Lambda\Lambda^T$ , the model is invariant to rotation and axis flips of the latent space.
- We can replace  $\Lambda$  with  $\Lambda Q$  for any orthonormal matrix  $Q$  and the model remains the same:  $(\Lambda Q)(\Lambda Q)^T = \Lambda(QQ^T)\Lambda^T = \Lambda\Lambda^T$ .
- This means that there is no "one best" setting of the parameters. An infinite number of parameters all give the ML score!
- Such models are called *un-identifiable* since two people both fitting ML parameters to the identical data will not be guaranteed to identify the same parameters.

Eric Xing

24

## Why FA

- Latent trajectories

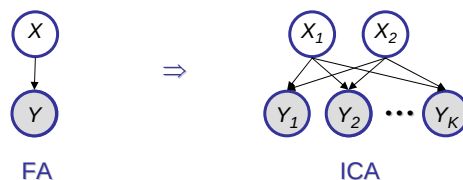


Eric Xing

25

## Independent Components Analysis (ICA)

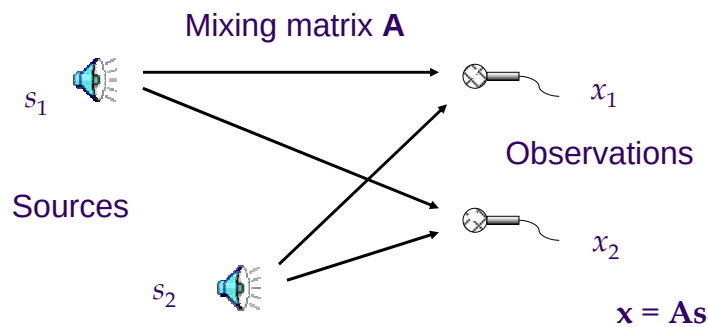
- ICA is similar to FA, except it assumes the latent source has non-Gaussian density.
- Hence ICA can extract higher order moments (not just second order).
- It is commonly used to solve blind source separation (cocktail party problem).



Eric Xing

26

## The simple “Cocktail Party” Problem



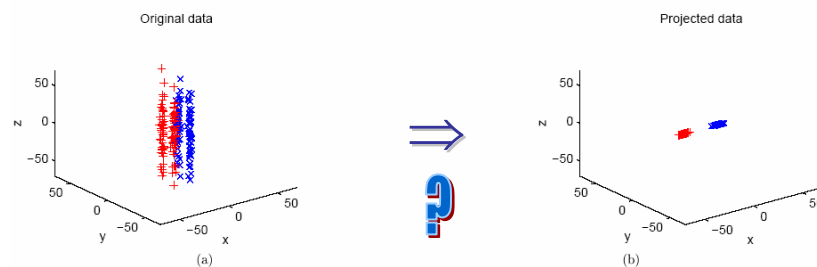
$n$  sources,  $m=n$  observations

We skip more details and next introduce a more interesting new algorithm!

Eric Xing

27

## Semi-supervised Metric Learning



Xing et al, NIPS 2003

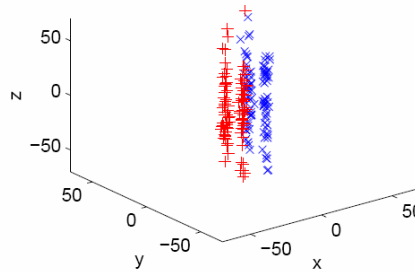
Eric Xing

28

## What is a good metric?



- What is a good metric over the input space for learning and data-mining



- How to convey metrics sensible to a human user (e.g., dividing traffic along highway lanes rather than between overpasses, categorizing documents according to writing style rather than topic) to a computer data-miner using a systematic mechanism?

Eric Xing

29

## Issues in learning a metric



- Data distribution is self-informing (E.g., lies in a sub-manifold)
  - Learning metric by finding an embedding of data in some space.
    - Con: does not reflect (changing) human subjectiveness.
- Explicitly labeled dataset offers clue for critical features
  - Supervised learning
    - Con: needs sizable homogeneous training sets.
- What about side information? (E.g., x and y look (or read) similar ...)
  - Providing small amount of qualitative and less structured side information is often much easier than stating explicitly a metric (what should be the metric for writing style?) or labeling a large set of training data.
- Can we learn a distance metric more informative than Euclidean distance using a small amount of side information?

Eric Xing

30

## Distance Metric Learning



Side information:

Suppose for some set of points  $\{x_i\}_{i=1}^m \subseteq \mathbb{R}^n$ , we are given:

$\mathcal{S} : (x_i, x_j) \in \mathcal{S}$  if  $x_i$  and  $x_j$  are similar

$\mathcal{D} : (x_i, x_j) \in \mathcal{D}$  if  $x_i$  and  $x_j$  are dissimilar

Distance metric learning:

Learn a distance metric of the form

$$d(x, y) = d_A(x, y) = \|x - y\|_A = \sqrt{(x - y)^T A (x - y)},$$

such that pairs of points  $(x_i, x_j)$  in  $\mathcal{S}$  have small squared distance.

- In general,  $A$  parameterizes a family of Mahalanobis distances over  $\mathbb{R}^n$ .
- Learning  $A$  is equivalent to finding a rescaling of a data:  $x \rightarrow A^{1/2}x$ .

Eric Xing

31

## Optimal Distance Metric



- Learning an optimal distance metric with respect to the side-information leads to the following optimization problem:

$$\min_A \sum_{(x_i, x_j) \in \mathcal{S}} \|x_i - x_j\|_A^2 \quad (1)$$

$$\text{s.t. } \sum_{(x_i, x_j) \in \mathcal{D}} \|x_i - x_j\|_A \geq 1, \quad (2)$$

$$A \geq 0. \quad (3)$$

- This optimization problem is **convex**. Local-minima-free algorithms exist.
- Xing et al 2003 provided an efficient **gradient descent + iterative constraint-projection method**

Eric Xing

32



## Examples of learned distance metrics



- Distance metrics learned on three-cluster artificial data:

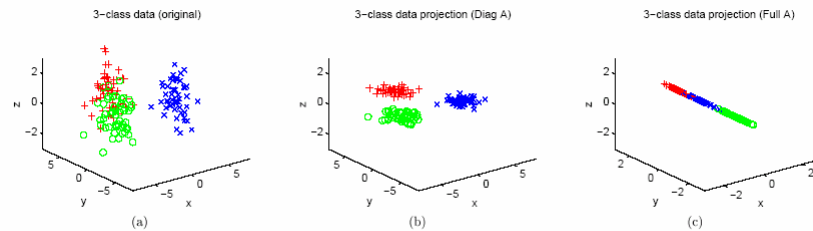


Figure 2: (a) Original data. (b) Rescaling corresponding to learned diagonal  $A$ . (c) Rescaling corresponding to full  $A$ .

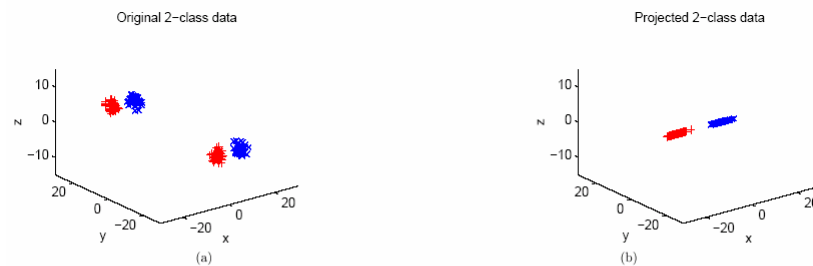
Eric Xing

33

## Application to Clustering



- Artificial Data I: a difficult two-class dataset



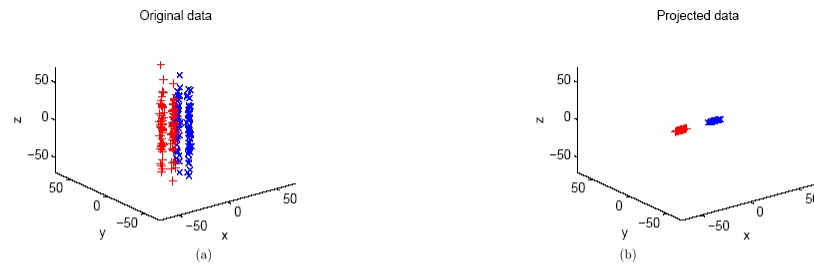
1. K-means: Accuracy = 0.4975
2. Constrained K-means: Accuracy = 0.5060
3. K-means + metric: Accuracy = 1
4. Constrained K-means + metric: Accuracy = 1

Eric Xing

34

## Application to Clustering

- Artificial Data II: two-class data with strong irrelevant feature



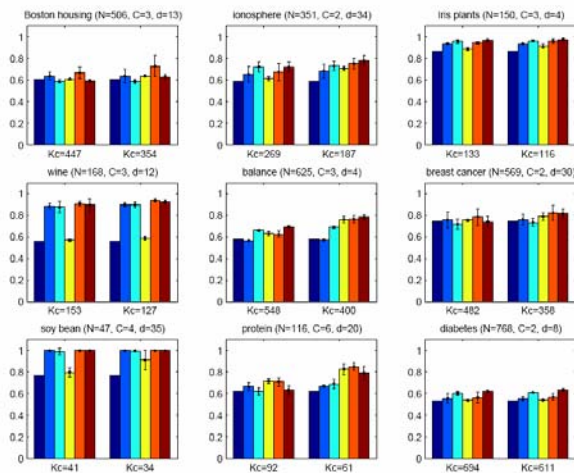
1. K-means: Accuracy = 0.4993
2. Constrained K-means: Accuracy = 0.5701
3. K-means + metric: Accuracy = 1
4. Constrained K-means + metric: Accuracy = 1

Eric Xing

35

## Application to Clustering

- 9 datasets from the UC Irvine repository



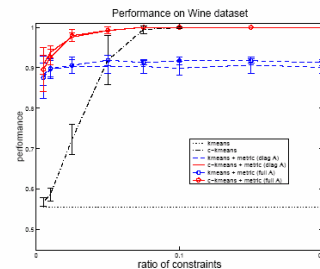
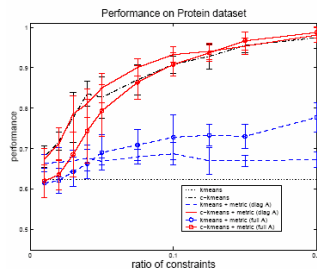
Eric Xing

36

## Accuracy vs. amount of side-information



- Two typical examples of how the quality of the clusters found increases with the amount of side-information.



Eric Xing

37

## Take home message



- Distance metric learning is an important problem in machine learning and data mining.
- A good distance metric can be learned from small amount of side-information in the form of similarity and dissimilarity constraints from data by solving a convex optimization problem.
- The learned distance metric can identify the most significant direction(s) in feature space that separates data well, effectively doing implicit Feature Selection.
- The learned distance metric can be used to improve clustering performance.

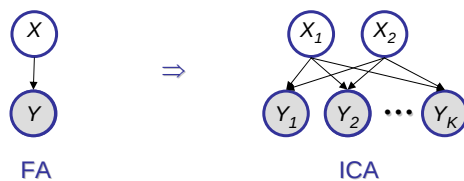
Eric Xing

38

# Independent Components Analysis (ICA)



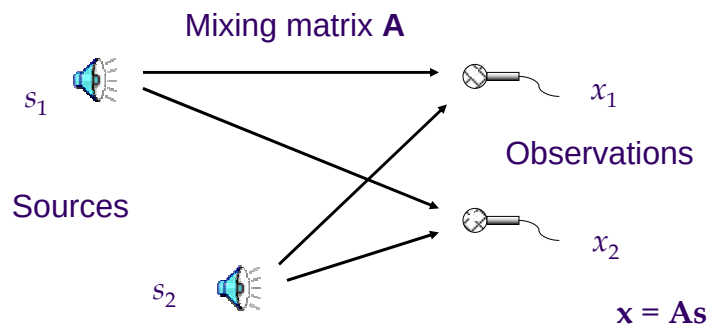
- ICA is similar to FA, except it assumes the latent source has non-Gaussian density.
- Hence ICA can extract higher order moments (not just second order).
- It is commonly used to solve blind source separation (cocktail party problem).



Eric Xing

39

# The simple “Cocktail Party” Problem

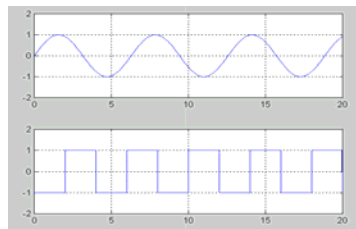


$n$  sources,  $m=n$  observations

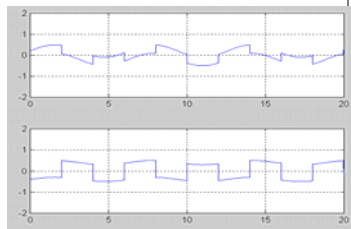
Eric Xing

40

## Motivation



Two Independent Sources



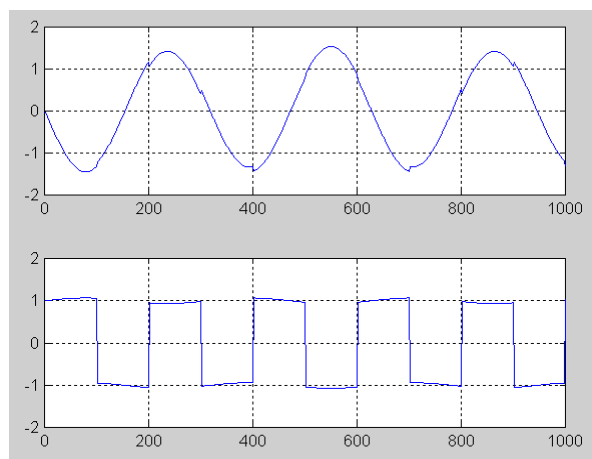
Mixture at two Mics

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t)$$

$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t)$$

$a_{ij}$  ... Depend on the distances of the microphones from the speakers

## Motivation



Get the Independent Signals out of the Mixture

# Blind Source Separation

- Suppose that there are  $k$  unknown independent sources

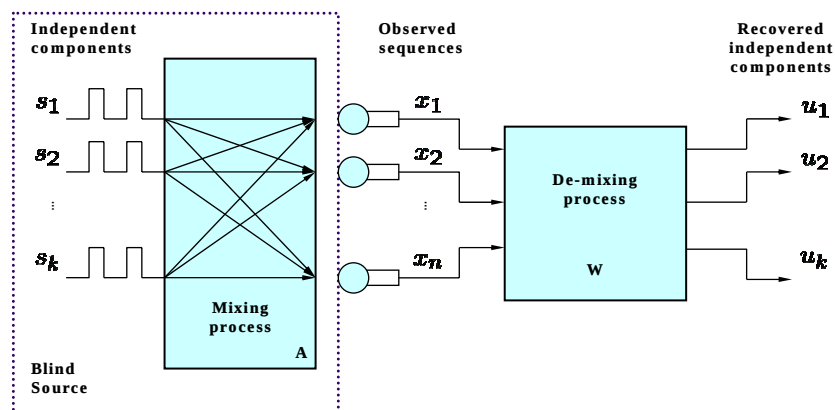
$$\mathbf{s}(t) = [s_1(t), \dots, s_k(t)]^T \quad \text{with} \quad E[\mathbf{s}(t)] = 1$$

- A data vector  $\mathbf{x}(t)$  is observed at each time point  $t$ , such that

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$$

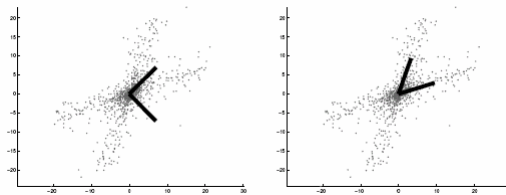
where  $\mathbf{A}$  is a  $n \times k$  full rank scalar matrix

# Blind source separation



## ICA versus PCA (and FA)

- Similarity
  - Feature extraction
  - Dimension reduction
- Difference
  - PCA uses up to second order moment of the data to produce uncorrelated components
  - ICA strives to generate components as independent as possible



Eric Xing

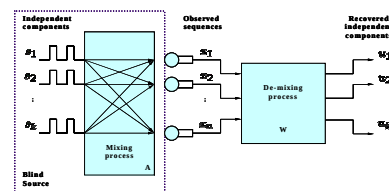
45

## Problem formulation

- The goal of ICA is to find a linear mapping  $\mathbf{W}$  such that the unmixed sequences  $\mathbf{u}$

$$\mathbf{u}(t) = \mathbf{W}\mathbf{x}(t) = \mathbf{W}\mathbf{A}\mathbf{s}(t)$$

are maximally statistically independent



- Find some

$$\mathbf{V} = \mathbf{W}\mathbf{A} = \mathbf{P}\mathbf{C}$$

where  $\mathbf{C}$  is a diagonal matrix and  $\mathbf{P}$  is a permutation matrix.

Eric Xing

46

## Principle of ICA: Nongaussianity



- The fundamental restriction in ICA is that the independent components must be nongaussian for ICA to be possible.
- This is because gaussianity is invariant under orthogonal transformation and hence make the matrix  $\mathbf{A}$  not identifiable for gaussian independent components.

## Measures of nongaussianity (1)



- Kurtosis
  - $\text{Kurt}(y) = E\{y^4\} - 3(E\{y^2\})^2$
  - Kurtosis can be very sensitive to outliers, when its value has to be estimate from a measured sample.
- Mutual information
- Negative Entropy



## FastICA Preprocessing



- Centering:
  - Make the x-s mean 0 variables
- Whitening
  - Transform the observed vector  $\mathbf{x}$  linearly so that it has unit variance:

$$\mathbf{E}\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\} = \mathbf{I}$$

- One can show that:

$$\tilde{\mathbf{x}} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{x} = \tilde{\mathbf{A}}\mathbf{s}$$

where  $\mathbf{E}\{\mathbf{x}\mathbf{x}^T\} = \mathbf{E}\mathbf{D}\mathbf{E}^T$

Eric Xing

49

## FastICA algorithm



- Initialize the weight matrix  $\mathbf{W}$
- Iteration:
  - $$\mathbf{W}^+ = \mathbf{W} + \text{diag}(\alpha_i)[\text{diag}(\beta_i) + \mathbf{E}\{\mathbf{g}(\mathbf{u}\mathbf{u}^T)\}]\mathbf{W}$$

where

$$\beta_i = -\mathbf{E}\{u_i g(u_i)\}, \alpha_i = -1/(\beta_i - \mathbf{E}\{g(u_i)'\})$$
  - Repeat until convergence  $\mathbf{W}^\infty$
- The ICAs are the components of  $\mathbf{W}^\infty\mathbf{x}(t)$

Eric Xing

50

## Summary



- There has been a wide discussion about the application of Independence Component Analysis (ICA) in Signal Processing, Neural Computation and Finance.
- First introduced as a novel tool to separate blind sources in a mixed signal.
- The Basic idea of ICA is to reconstruct from observation sequences the hypothesized independent original sequences.