

Linear Dimensionality Reduction

Piyush Rai

Machine Learning - CS5350/CS6350
November 03, 2009

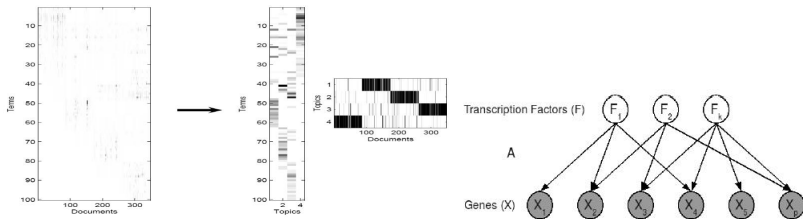
- ▶ Dimensionality Reduction
- ▶ Principal Component Analysis (PCA)
 - ▶ Maximum Variance Formulation
 - ▶ Minimum Error Formulation
 - ▶ Applications of PCA
 - ▶ PCA in High Dimensions
- ▶ Probabilistic Principal Component Analysis (PPCA)
 - ▶ Latent Variable Model
 - ▶ Maximum Likelihood Solution
 - ▶ Expectation Maximization (EM) for PPCA
 - ▶ Factor Analysis (related to PPCA)
- ▶ *Supervised* Dimensionality Reduction

What is Dimensionality Reduction?

- ▶ Most interesting datasets have very large number of features
 - ▶ Text documents, microarray gene-expressions, images, etc
 - ▶ Each datapoint is a vector in some high-dimensional space
- ▶ Want compact (yet *useful*) representations of data
 - ▶ *Without losing much information*
- ▶ Possible because data often lies on, or close to, a low(er) dimensional *subspace*
 - ▶ Subspace can be a linear or a nonlinear manifold
- ▶ Important: Different from *Feature Selection*
 - ▶ It's rather **Feature Extraction**

Why do Dimensionality Reduction?

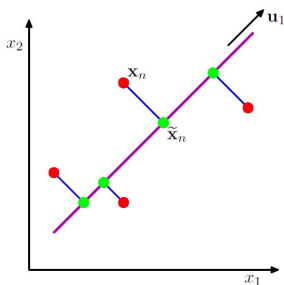
- ▶ Understanding structure underlying high-dimensional data
 - ▶ Scientific Visualization
 - ▶ Thematic or Causal Structure (e.g., Topic Models, Factor Analysis)



- ▶ Better generalization performance (dealing with overfitting)
- ▶ Making learning algorithms more efficient (space/time)

Principal Component Analysis

- ▶ A linear dimensionality reduction technique
- ▶ Finds the best **linear subspace** underlying the data that
 - ▶ Captures **maximum variance** of the projected data
 - ▶ Results in **minimum projection error**
 - ▶ *Note: Both the above conditions are equivalent !*



- ▶ PCA computes an orthogonal projection of data onto this subspace, e.g., $\tilde{\mathbf{x}} = \mathbf{u}_1^T \mathbf{x}$

PCA: Maximum Variance Formulation

- ▶ Given: dataset of observations $\{\mathbf{x}_n\}$, $n = 1, \dots, N$
- ▶ $\mathbf{x}_n \in \mathbb{R}^D$: Euclidean variable with dimensionality D
- ▶ **Goal:** Project data onto a linear subspace of dimensionality $M < D$
- ▶ .. *while maximizing the variance of projected data*
- ▶ Assume M is given

Maximum Variance Formulation (Contd.)

- ▶ Consider projection onto a one-dim. subspace ($M = 1$)
- ▶ Define *direction* of subspace as a D -dimensional vector \mathbf{u}_1
- ▶ Choose \mathbf{u}_1 to be a unit vector: $\mathbf{u}_1^T \mathbf{u}_1 = 1$
- ▶ Projection of \mathbf{x}_n onto \mathbf{u}_1 : $\mathbf{u}_1^T \mathbf{x}_n$
- ▶ Mean of projected data: $\mathbf{u}_1^T \bar{\mathbf{x}}$, where $\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n$
- ▶ Variance of projected data:

$$\frac{1}{N} \sum_{n=1}^N \{\mathbf{u}_1^T \mathbf{x}_n - \mathbf{u}_1^T \bar{\mathbf{x}}\}^2 = \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$$

- ▶ \mathbf{S} is the data covariance matrix:

$$\mathbf{S} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T$$

Maximum Variance Formulation (Contd.)

- ▶ Want subspace that maximizes projected data variance
- ▶ i.e., maximize $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$ w.r.t. \mathbf{u}_1

Maximum Variance Formulation (Contd.)

- ▶ Want subspace that maximizes projected data variance
- ▶ i.e., maximize $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$ w.r.t. \mathbf{u}_1
- ▶ Don't want the trivial solution $\|\mathbf{u}_1\| \rightarrow \infty$
- ▶ Constraint: $\mathbf{u}_1^T \mathbf{u}_1 = 1$
- ▶ Lagrangian: $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 + \lambda_1(1 - \mathbf{u}_1^T \mathbf{u}_1)$
- ▶ Stationary Point: $\mathbf{S} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1$
 - ▶ \mathbf{u}_1 must be an eigenvector of \mathbf{S}
- ▶ *Value* of projected data variance: $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 = \lambda_1$
- ▶ Maximum variance captured when we set \mathbf{u}_1 equal to the eigenvector associated with **largest** eigenvalue λ_1

Maximum Variance Formulation (Contd.)

- ▶ Additional principal components defined incrementally
- ▶ Choose each new direction as one that maximizes variance of projected data **and** is orthogonal to those already considered
- ▶ M dimensional subspace is defined by M eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_M$ of data covariance matrix \mathbf{S} corresponding to M largest eigenvalues $\lambda_1, \dots, \lambda_M$
- ▶ Proof by induction
- ▶ Eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_M$ form an orthonormal set: $\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}$

PCA Algorithm

- ▶ Compute mean $\bar{\mathbf{x}}$ and covariance \mathbf{S} of data
- ▶ Do eigen decomposition of \mathbf{S} to find M largest eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_M$ corresponding to M largest eigenvalues $\lambda_1, \dots, \lambda_M$
- ▶ Full eigen decomposition of $D \times D$ matrix takes $O(D^3)$ time
- ▶ Efficient techniques exist if only M eigenvectors are needed

PCA: Minimum Error Formulation

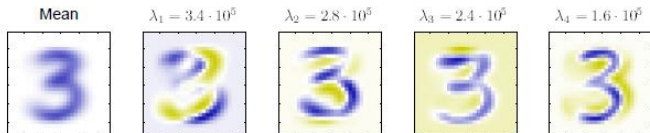
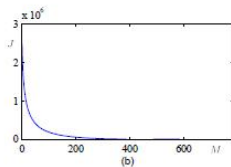
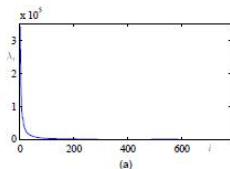
- ▶ Minimize distortion error $J = \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2$ of the projected data ($\tilde{\mathbf{x}}_n$ is projection of \mathbf{x}_n)

$$J = \sum_{i=M+1}^D \mathbf{u}_i^T \mathbf{S} \mathbf{u}_i$$

- ▶ Minimization w.r.t. \mathbf{u}_i gives solution: $\mathbf{S} \mathbf{u}_i = \lambda_i \mathbf{u}_i$
- ▶ Orthonormal \mathbf{u}_i give distortion measure $J = \sum_{i=M+1}^D \lambda_i$
- ▶ Minimum distortion achieved by having the **discarded** eigenvectors to be those corresponding to $D - M$ **smallest** eigenvalues
- ▶ Thus the principal subspace consists of the M **retained** eigenvectors corresponding to M **largest** eigenvalues

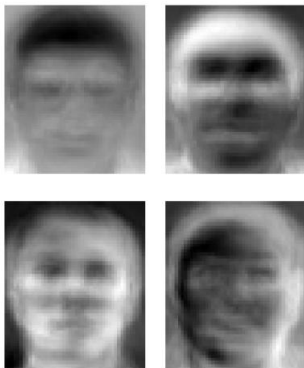
PCA Example: Data Compression (1)

- ▶ Offline digits dataset
- ▶ Eigenvalue spectrum (figure a)
- ▶ Total distortion: sum of discarded eigenvalues (figure b)



PCA Example: Data Compression (2)

- ▶ Face dataset
- ▶ Eigenfaces: Eigenvectors of face dataset
- ▶ Each face image can be represented as a linear combination of a small set of eigenfaces



PCA for High Dimensional Data

- ▶ In many cases, number of data points is smaller than data dimensionality ($N < D$)
- ▶ At least, $D - N + 1$ eigenvalues are zero
- ▶ $D \times D$ covariance matrix $\mathbf{S} = \frac{1}{N}\mathbf{X}^T\mathbf{X}$ (assuming centered data): **Eigen decomposition for high D can be expensive !**
- ▶ \mathbf{S} has $D - N + 1$ eigenvalues of value zero and $N - 1$ eigenvalues as the $N \times N$ matrix $\frac{1}{N}\mathbf{X}\mathbf{X}^T$
- ▶ Eigen decomposition of $\frac{1}{N}\mathbf{X}\mathbf{X}^T$ is easier !
- ▶ If \mathbf{v}_i are eigenvectors of $\frac{1}{N}\mathbf{X}\mathbf{X}^T$ then the normalized eigenvectors \mathbf{u}_i of \mathbf{S} are given by:

$$\mathbf{u}_i = \frac{1}{(N\lambda_i)^{1/2}}\mathbf{X}^T\mathbf{v}_i$$

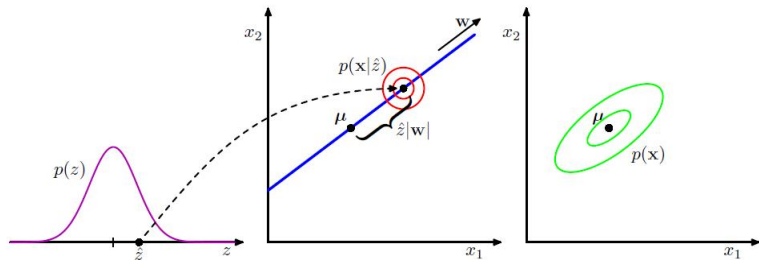
Probabilistic PCA

- ▶ A probabilistic formulation of PCA
- ▶ Based on a latent variable model (LVM) of the data
- ▶ PCA solution recovered as the MLE solution of the LVM
- ▶ Several advantages over standard PCA
 - ▶ Can use EM algorithm if only a few eigenvectors required (thus can be computationally more efficient)
 - ▶ Probabilistic framework can deal with missing data
 - ▶ More complex probabilistic models can be formulated
 - ▶ Allows a Bayesian treatment of PCA
 - ▶ And many others :)

Probabilistic PCA (Contd.)

- ▶ Introduce an M dimensional **latent** variable \mathbf{z}
 - ▶ \mathbf{z} corresponds to the principal subspace of PCA
- ▶ A linear-Gaussian generative model of data: $\mathbf{x} = \mathbf{W}\mathbf{z} + \mu + \epsilon$
 - ▶ All marginal and conditional distributions are Gaussian
 - ▶ \mathbf{W} is $D \times M$, μ is a D -dimensional vector
- ▶ Gaussian prior distribution on latent variable \mathbf{z}
 - ▶ $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, I)$
- ▶ Gaussian conditional distribution on observed variable \mathbf{x}
 - ▶ $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \mu, \sigma^2 I)$
- ▶ Columns of \mathbf{W} span a linear subspace of data
 - ▶ This subspace corresponds to the principal subspace

Probabilistic PCA, Pictorially



- ▶ \mathbf{W} specifies a set of directions, μ is the mean of the data, and the latent variable vector \mathbf{z} tells how much we should move from the mean along each direction
- ▶ \mathbf{x} generated by adding spherical noise ϵ to that location

Probabilistic PCA (Contd.)

- ▶ Observed data (marginal) likelihood

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

- ▶ Both $p(\mathbf{x}|\mathbf{z})$ and $p(\mathbf{z})$ are Gaussian
 - ▶ $p(\mathbf{x})$ is Gaussian too !

- ▶ $p(\mathbf{x}) = \mathcal{N}or(\mathbf{x}|\mu, \mathbf{C})$

$$\text{where } \mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}$$

- ▶ $p(\mathbf{z}|\mathbf{x}) = \mathcal{N}or(\mathbf{z}|\mathbf{M}^{-1}\mathbf{W}^T(\mathbf{x} - \mu), \sigma^2\mathbf{M})$

$$\text{where } \mathbf{M} = \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}$$

- ▶ Want to find model parameters \mathbf{W} , μ , and σ^2
- ▶ Can find parameters by maximum likelihood

Maximum Likelihood Parameter Estimation

- ▶ Log-likelihood function

$$\ln p(\mathbf{X}|\mu, \mathbf{W}, \sigma^2) = \sum_{n=1}^N \ln p(\mathbf{x}_n|\mu, \mathbf{W}, \sigma^2)$$

$$= -\frac{ND}{2} \ln(2\pi) - \frac{N}{2} \ln \det \mathbf{C} - \frac{1}{2} \sum_{n=1}^N (\mathbf{x}_n - \mu)^T \mathbf{C}^{-1} (\mathbf{x}_n - \mu)$$

Maximum Likelihood Parameter Estimation

- ▶ Log-likelihood function

$$\ln p(\mathbf{X}|\mu, \mathbf{W}, \sigma^2) = \sum_{n=1}^N \ln p(\mathbf{x}_n|\mu, \mathbf{W}, \sigma^2)$$

$$= -\frac{ND}{2} \ln(2\pi) - \frac{N}{2} \ln \det \mathbf{C} - \frac{1}{2} \sum_{n=1}^N (\mathbf{x}_n - \mu)^T \mathbf{C}^{-1} (\mathbf{x}_n - \mu)$$

- ▶ Setting the derivative w.r.t. μ to zero yields $\mu = \bar{\mathbf{x}}$.
Back-substituting:

- ▶ $\ln p(\mathbf{X}|\mu, \mathbf{W}, \sigma^2) = -\frac{N}{2} \{ D \ln(2\pi) + \ln \det \mathbf{C} + \text{Tr}(\mathbf{C}^{-1} \mathbf{S}) \}$

Maximum Likelihood Parameter Estimation

- ▶ Log-likelihood function

$$\begin{aligned}\ln p(\mathbf{X}|\mu, \mathbf{W}, \sigma^2) &= \sum_{n=1}^N \ln p(\mathbf{x}_n|\mu, \mathbf{W}, \sigma^2) \\ &= -\frac{ND}{2} \ln(2\pi) - \frac{N}{2} \ln \det \mathbf{C} - \frac{1}{2} \sum_{n=1}^N (\mathbf{x}_n - \mu)^T \mathbf{C}^{-1} (\mathbf{x}_n - \mu)\end{aligned}$$

- ▶ Setting the derivative w.r.t. μ to zero yields $\mu = \bar{\mathbf{x}}$.
Back-substituting:

- ▶ $\ln p(\mathbf{X}|\mu, \mathbf{W}, \sigma^2) = -\frac{N}{2} \{ D \ln(2\pi) + \ln \det \mathbf{C} + \text{Tr}(\mathbf{C}^{-1} \mathbf{S}) \}$

- ▶ Maximum likelihood solutions:

$$\mathbf{W}_{ML} = \mathbf{U}_M (\mathbf{L}_M - \sigma^2 \mathbf{I})^{\frac{1}{2}} \mathbf{R} \quad \text{and} \quad \sigma_{ML}^2 = \frac{1}{D-M} \sum_{i=M+1}^D \lambda_i$$

Maximum Likelihood Parameter Estimation

- ▶ Log-likelihood function

$$\begin{aligned}\ln p(\mathbf{X}|\mu, \mathbf{W}, \sigma^2) &= \sum_{n=1}^N \ln p(\mathbf{x}_n|\mu, \mathbf{W}, \sigma^2) \\ &= -\frac{ND}{2} \ln(2\pi) - \frac{N}{2} \ln \det \mathbf{C} - \frac{1}{2} \sum_{n=1}^N (\mathbf{x}_n - \mu)^T \mathbf{C}^{-1} (\mathbf{x}_n - \mu)\end{aligned}$$

- ▶ Setting the derivative w.r.t. μ to zero yields $\mu = \bar{\mathbf{x}}$.
Back-substituting:

- ▶ $\ln p(\mathbf{X}|\mu, \mathbf{W}, \sigma^2) = -\frac{N}{2} \{ D \ln(2\pi) + \ln \det \mathbf{C} + \text{Tr}(\mathbf{C}^{-1} \mathbf{S}) \}$

- ▶ **Maximum likelihood solutions:**

$$\mathbf{W}_{ML} = \mathbf{U}_M (\mathbf{L}_M - \sigma^2 \mathbf{I})^{\frac{1}{2}} \mathbf{R} \quad \text{and} \quad \sigma_{ML}^2 = \frac{1}{D-M} \sum_{i=M+1}^D \lambda_i$$

- ▶ \mathbf{U}_M : $D \times M$ matrix whose columns are top M e.vecs of \mathbf{S}
- ▶ \mathbf{L}_M : $M \times M$ diag. matrix with the corresponding e.vals λ_i as elements
- ▶ \mathbf{R} : arbitrary $M \times M$ rotation matrix

Latent Space Mapping and Equivalence with Classical PCA

- ▶ Posterior mean: $\mathbb{E}[\mathbf{z}|\mathbf{x}] = \mathbf{M}^{-1}\mathbf{W}_{ML}^T(\mathbf{x} - \bar{\mathbf{x}})$
- ▶ Taking the limit $\sigma^2 \rightarrow 0$, the posterior mean reduces to:
$$(\mathbf{W}_{ML}^T\mathbf{W}_{ML})^{-1}\mathbf{W}_{ML}^T(\mathbf{x} - \bar{\mathbf{x}})$$
- ▶ An orthogonal projection of the data onto the latent space (like standard PCA)

EM Algorithm for PCA

- ▶ Traditional approaches to PCA all work with $D \times D$ sample covariance matrix
 - ▶ Can be expensive for large D
- ▶ EM based iterative approach can give computational benefits
 - ▶ Also allows dealing with missing data in a principled way

EM Algorithm for PCA (Contd.)

- ▶ Write complete data likelihood as

$$\ln p(\mathbf{X}, \mathbf{Z} | \mu, \mathbf{W}, \sigma^2) = \sum_{n=1}^N \{ \ln p(\mathbf{x}_n | \mathbf{z}_n) + \ln p(\mathbf{z}_n) \}$$

EM Algorithm for PCA (Contd.)

- ▶ Write complete data likelihood as

$$\ln p(\mathbf{X}, \mathbf{Z} | \mu, \mathbf{W}, \sigma^2) = \sum_{n=1}^N \{ \ln p(\mathbf{x}_n | \mathbf{z}_n) + \ln p(\mathbf{z}_n) \}$$

- ▶ E Step:

$$\begin{aligned} \mathbb{E}[\ln p(\mathbf{X}, \mathbf{Z} | \mu, \mathbf{W}, \sigma^2)] &= - \sum_{n=1}^N \left\{ \frac{D}{2} \ln(2\pi\sigma^2) + \frac{1}{2} \text{Tr}(\mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T]) \right. \\ &+ \frac{1}{2\sigma^2} \|\mathbf{x}_n - \mu\|^2 - \frac{1}{\sigma^2} \mathbb{E}[\mathbf{z}_n]^T \mathbf{W}^T (\mathbf{x}_n - \mu) \\ &\left. + \frac{1}{2\sigma^2} \text{Tr}(\mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T] \mathbf{W}^T \mathbf{W}) \right\} \end{aligned}$$

EM Algorithm for PCA (Contd.)

- ▶ Write complete data likelihood as

$$\ln p(\mathbf{X}, \mathbf{Z} | \mu, \mathbf{W}, \sigma^2) = \sum_{n=1}^N \{ \ln p(\mathbf{x}_n | \mathbf{z}_n) + \ln p(\mathbf{z}_n) \}$$

- ▶ E Step:

$$\begin{aligned} \mathbb{E}[\ln p(\mathbf{X}, \mathbf{Z} | \mu, \mathbf{W}, \sigma^2)] &= - \sum_{n=1}^N \left\{ \frac{D}{2} \ln(2\pi\sigma^2) + \frac{1}{2} \text{Tr}(\mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T]) \right. \\ &+ \frac{1}{2\sigma^2} \|\mathbf{x}_n - \mu\|^2 - \frac{1}{\sigma^2} \mathbb{E}[\mathbf{z}_n]^T \mathbf{W}^T (\mathbf{x}_n - \mu) \\ &\left. + \frac{1}{2\sigma^2} \text{Tr}(\mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T] \mathbf{W}^T \mathbf{W}) \right\} \end{aligned}$$

- ▶ Expected likelihood depends only on the sufficient statistics of the Gaussian (computed using old parameter estimates):

$$\begin{aligned} \mathbb{E}[\mathbf{z}_n] &= \mathbf{M}^{-1} \mathbf{W}^T (\mathbf{x}_n - \bar{\mathbf{x}}) \\ \mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T] &= \sigma^2 \mathbf{M}^{-1} + \mathbb{E}[\mathbf{z}_n] \mathbb{E}[\mathbf{z}_n]^T \\ &\text{(using } \mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T] = \text{cov}[\mathbf{z}_n] + \mathbb{E}[\mathbf{z}_n] \mathbb{E}[\mathbf{z}_n]^T) \end{aligned}$$

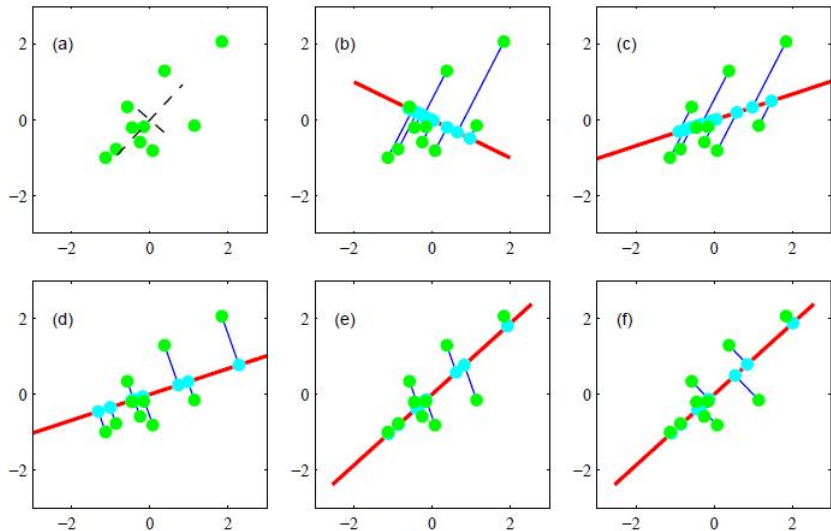
EM Algorithm for PCA (Contd.)

- ▶ M Step: Maximize expected (complete) data likelihood w.r.t. parameters \mathbf{W} and σ^2

$$\mathbf{W}_{new} = [\sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}}) \mathbb{E}[\mathbf{z}_n]^T] [\sum_{n=1}^N \mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T]]^{-1}$$
$$\sigma_{new}^2 = \frac{1}{ND} \sum_{n=1}^N \{ \|\mathbf{x}_n - \bar{\mathbf{x}}\|^2 - 2 \mathbb{E}[\mathbf{z}_n]^T \mathbf{W}_{new}^T (\mathbf{x}_n - \bar{\mathbf{x}}) + \text{Tr}(\mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T] \mathbf{W}_{new}^T \mathbf{W}_{new}) \}$$

- ▶ Computationally more efficient than standard PCA for large-scale applications
- ▶ Also possible to do online processing
- ▶ Can treat missing data as random variables and compute them in E step

EM for PCA, Pictorially



Factor Analysis

- ▶ Closely related to Probabilistic PCA
- ▶ Except that the conditional distribution has a diagonal, not isotropic covariance

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \mu, \Psi)$$

- ▶ Parameter estimation by EM algorithm (similar to Probabilistic PCA)

Supervised Dimensionality Reduction

- ▶ Dimensionality reduction with label information (ultimate goal is classification)
- ▶ PCA does not take into account label information
 - ▶ Only chooses directions of maximum variance
- ▶ Fisher Discriminant Analysis (FDA) does !
- ▶ PCA: magenta line, FDA: green line

