SUBSPACE MODELS

J. Elder

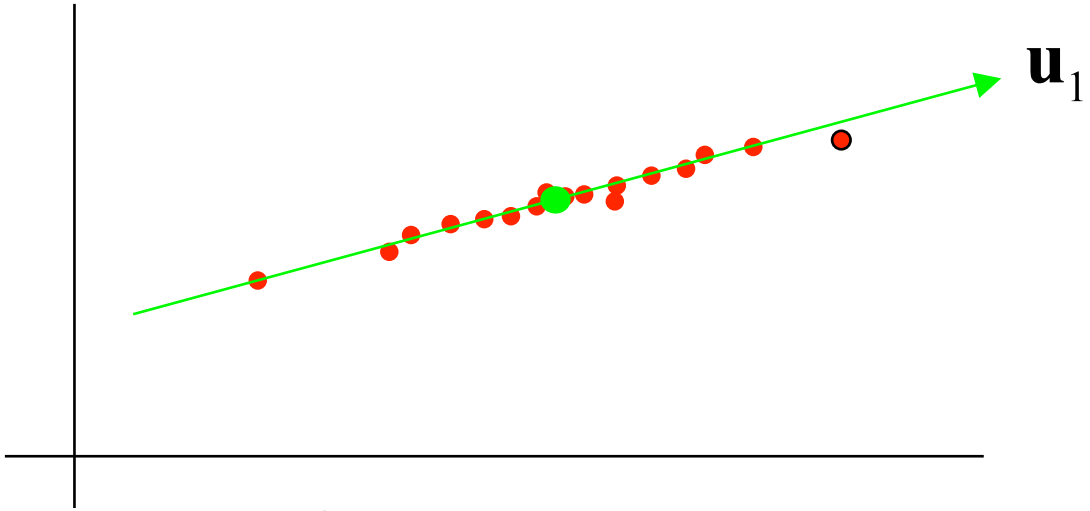
CSE 6390/PSYC 6225 Computational Modeling of Visual Perception

Credits

- Some of these slides were sourced and/or modified from Simon Prince, University College London

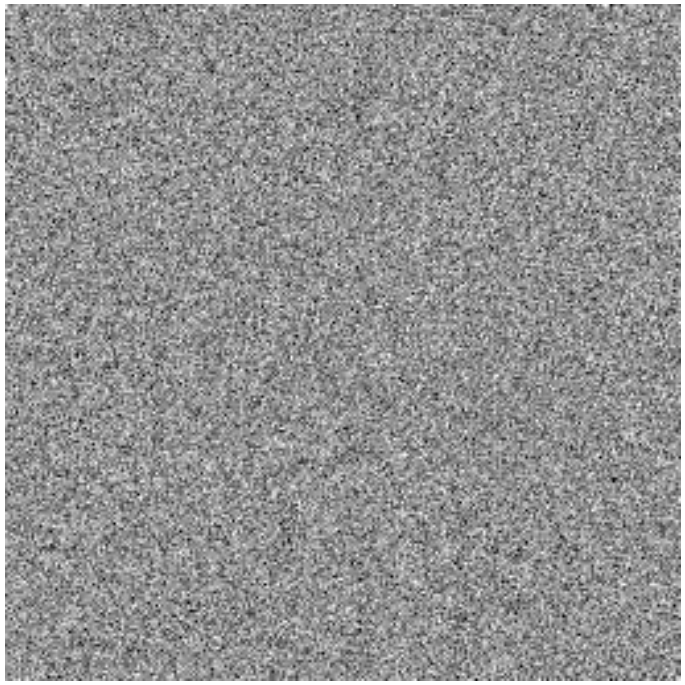
Subspace Models

- Natural images have high dimensionality *D*
  - e.g., for an 1800 x 1200 colour image, *D*≅6.5 million.
- There is typically insufficient training data to learn a probabilistic model in such a high-dimensional space.
- Fortunately, natural images actually live in a much smaller subspace, or manifold, of this high-dimensional space.



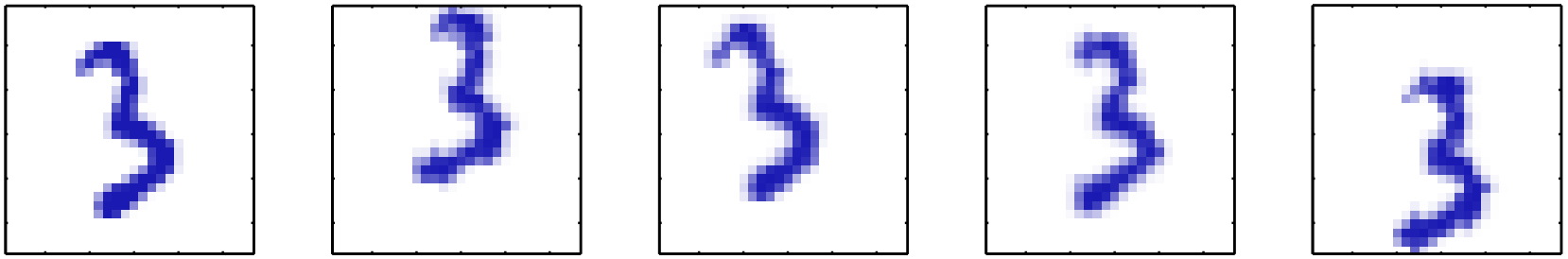
Subspace Models

- For example, you will have to wait a long time before a sample of white noise looks like a natural image.



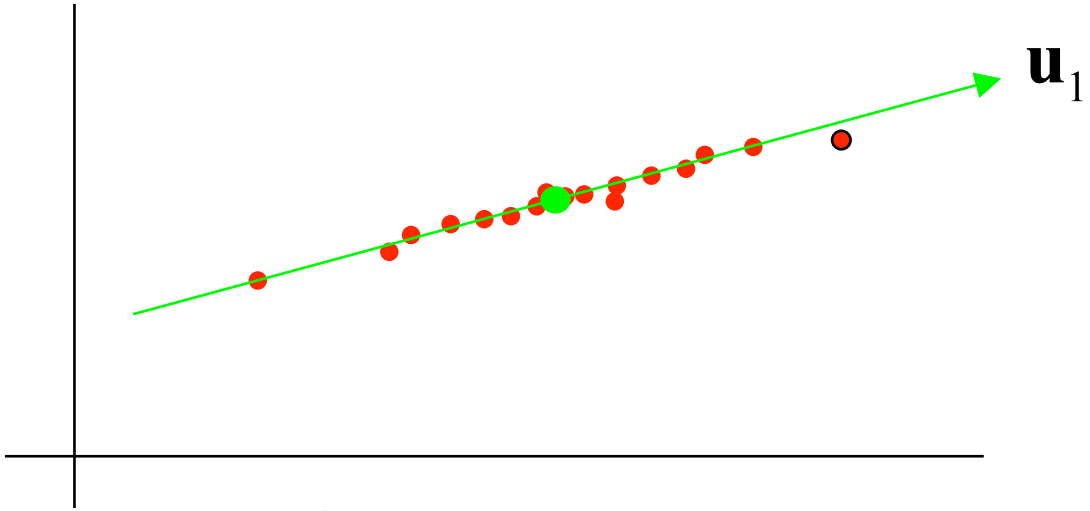
Subspace Models

- e.g., standard transformations (e.g., translations, rotations, scalings) of objects produce images populating a low-dimensional manifold embedded in this high-dimensional space



Subspace Models

- The goal of subspace methods is to discover the low-dimensional subspace in which the data lie and exploit the lower-dimensionality to allow efficient and detailed modeling.



Subspace Models

- We will mainly consider linear subspaces
  - A line if $D=2$
  - A line or a plane if $D=3$
  - A hyperplane of dimensionality $[1,...,D-1]$ for higher D
- But we will also consider some methods to deal with nonlinear manifolds.

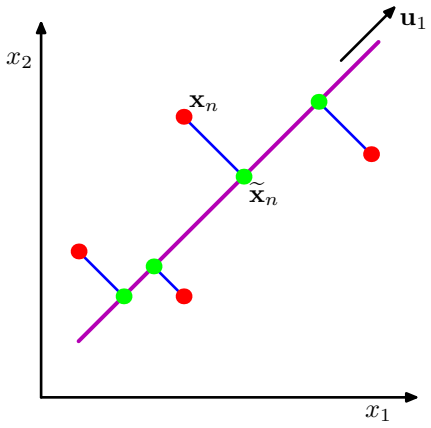
PRINCIPAL COMPONENT ANALYSIS

J. Elder

CSE 6390/PSYC 6225 Computational Modeling of Visual Perception

Principal Component Analysis

- PCA finds the linear subspace that
  - maximizes the explained variance
  - equivalently, minimizes the unexplained variance
- PCA can be applied to any multidimensional dataset
  - (data do not have to be Gaussian)



Maximum Variance Formulation

Observations $\{\mathbf{x}_n\}, n = 1,\ldots N$

Observation $\mathbf{x}_n$ is a high-dimensional vector of dimension D

Let $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_n$ be the sample mean and $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^t$ be the sample covariance

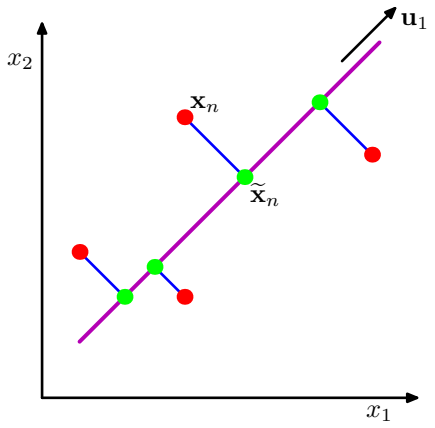
Goal: Project the data onto subspace of dimension $M < D$

Consider a direction in the data space given by unit vector $\mathbf{u}_1$.

Now imagine projecting all of the data onto this unit vector.

The mean of the projected data is $\mathbf{u}_1^t\bar{\mathbf{x}}$.

The variance of the projected data is $\frac{1}{N}\sum_{i=1}^{N}(\mathbf{u}_1^t\mathbf{x}_n - \mathbf{u}_1^t\bar{\mathbf{x}})^2 = \mathbf{u}_1^t\mathbf{S}\mathbf{u}_1$



Maximum Variance Formulation

We want to select the unit vector $\mathbf{u}_1$ that maximizes the projected variance $\mathbf{u}_1^t \mathbf{S} \mathbf{u}_1$

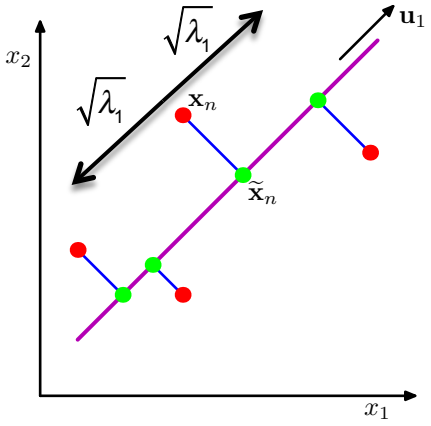
To do this, we use a Lagrange multiplier λ_1 to maintain the constraint that $\mathbf{u}_1$ be a unit vector.

Thus we seek to maximize $\mathbf{u}_1^t \mathbf{S} \mathbf{u}_1 + \lambda_1 \left(1 - \mathbf{u}_1^t \mathbf{u}_1\right)$

Setting the derivative with respect to $\mathbf{u}_1$ to 0, we have $S\mathbf{u}_1 = \lambda_1 \mathbf{u}_1$

Thus $\mathbf{u}_1$ is an eigenvector of $\mathbf{S}$.

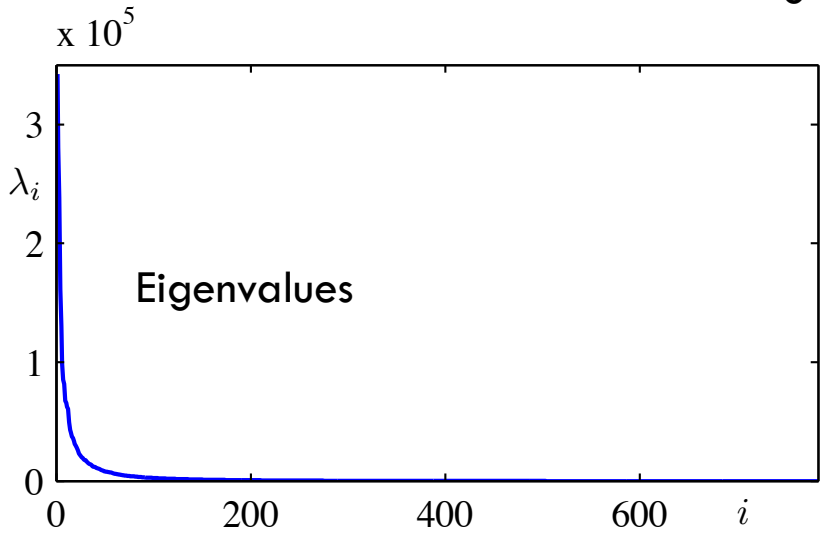
Left-multiplying by $\mathbf{u}_1^t$, we see that the projected variance $\mathbf{u}_1^t \mathbf{S} \mathbf{u}_1 = \lambda_1$.

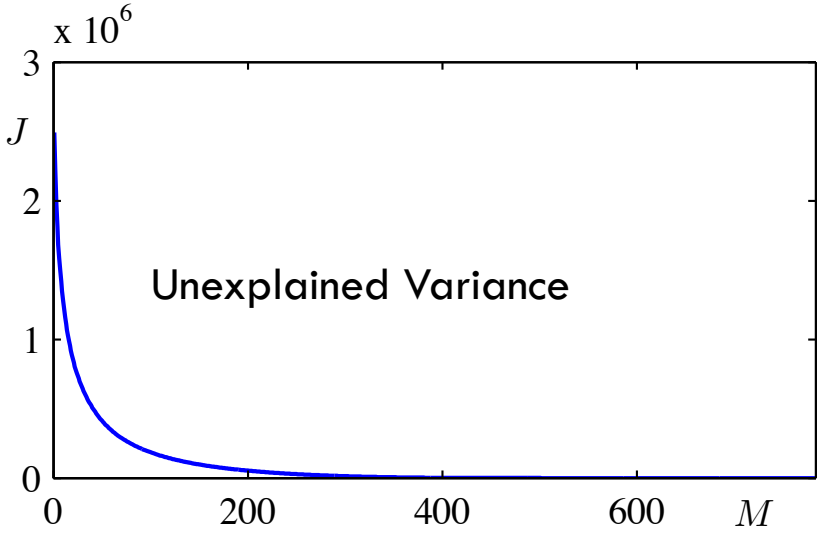


Dimensionality Reduction

- The next direction $\mathbf{u}_2$ can be chosen by maximizing projected variance in the *D-1* dimensional subspace orthogonal to $\mathbf{u}_1$.
- Typically, most of the variance is captured in a relatively small linear subspace.

Digit '3' Dataset





Computational Cost

- Computing full eigenvector decomposition is $O(D^3)$.
- If we only need the first M eigenvectors, the cost is $O(MD^2)$.
- However, this could still be very expensive if D is large

e.g., For an 1800×1600 image and $M = 100$, $O(650 \text{ million})$

Computational Cost

- But the number of training images N is usually much smaller than D, and this leads to a trick:

Let $\mathbf{X}$ be the $N \times D$ centred data matrix whose nth row is given by $\left(\mathbf{x}_n - \bar{\mathbf{x}}\right)^t$.

Then the sample covariance matrix is $\mathbf{S} = \frac{1}{N}\mathbf{X}^t\mathbf{X}$.

and the eigenvector equation is $\frac{1}{N}\overbrace{\mathbf{X}^t\mathbf{X}}^{D \times D}\mathbf{u}_i = \lambda_i \mathbf{u}_i$

Pre-multiplying both sides by $\mathbf{X}$ yields $\frac{1}{N}\mathbf{X}\mathbf{X}^t\left(\mathbf{X}\mathbf{u}_i\right) = \lambda_i\left(\mathbf{X}\mathbf{u}_i\right)$

Now letting $\mathbf{v}_i = \mathbf{X}\mathbf{u}_i$, we have

$$\frac{1}{N}\overbrace{\mathbf{X}\mathbf{X}^t}^{N \times N}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$ ← **Much smaller eigenvector problem!**

Computational Cost

- To find the eigenvectors of **S**, we premultiply by $\mathbf{X}^t$:

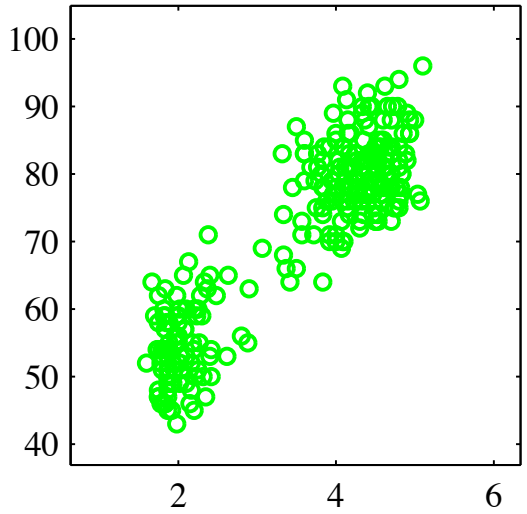
$$\frac{1}{N}\overbrace{\mathbf{X}\mathbf{X}^t}^{N\times N}\mathbf{v}_i = \lambda_i \mathbf{v}_i \rightarrow \overbrace{\left(\frac{1}{N}\mathbf{X}^t\mathbf{X}\right)}^{S}\left(\mathbf{X}^t\mathbf{v}_i\right) = \lambda_i\left(\mathbf{X}^t\mathbf{v}_i\right)$$

and, normalized to unit length, the eigenvectors are $\mathbf{u}_i = \frac{1}{\sqrt{N\lambda_i}}\mathbf{X}^t\mathbf{v}_i$

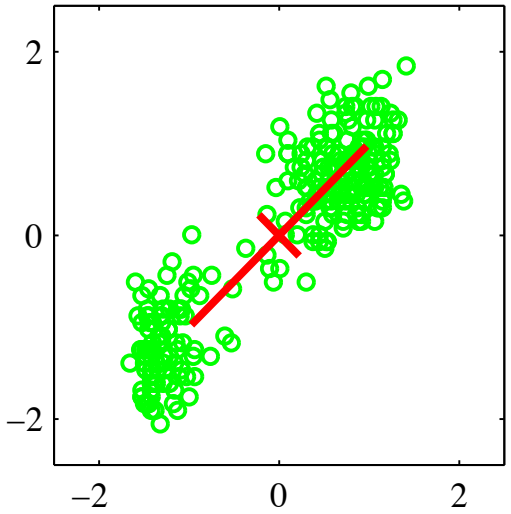
Note that these *N* eigenvectors live in the *N*-dimensional subspace spanned by the training images.

Applications of PCA

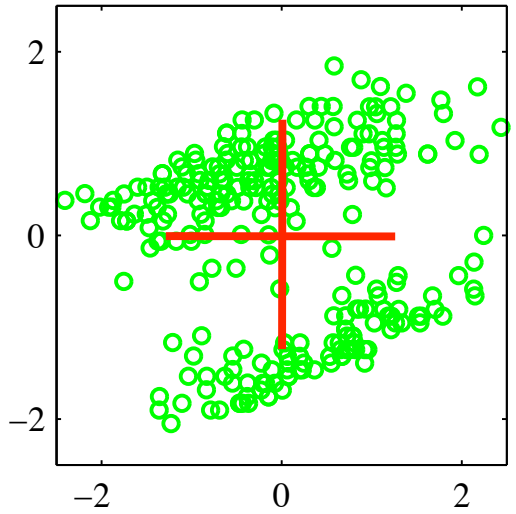
Pre-Whitening



Original Data

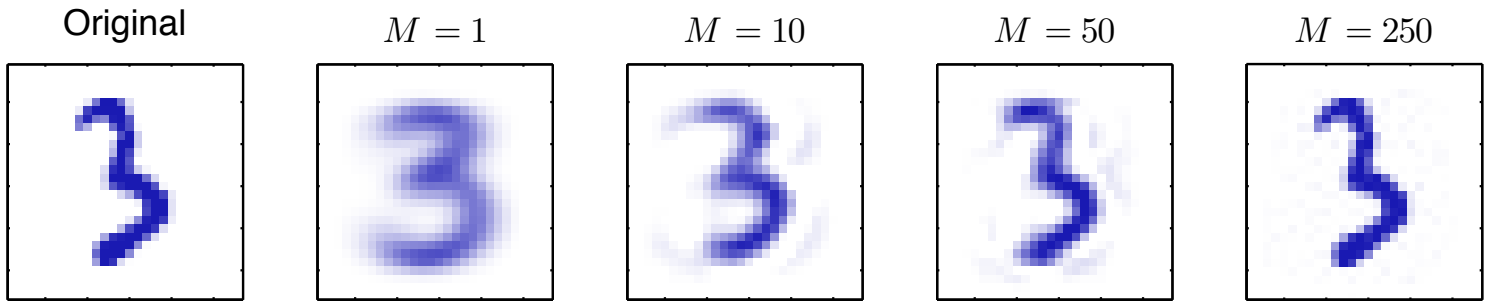


Normalized to 0-mean
and unit variance (z-scores)

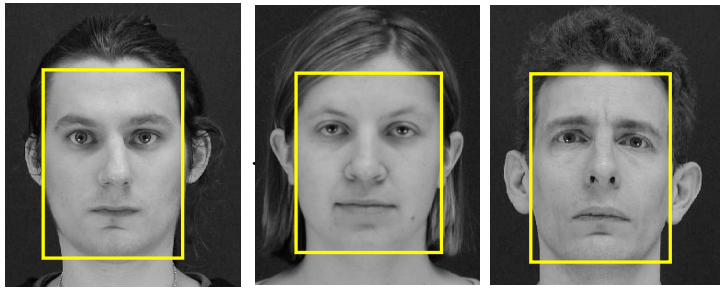


Whitened

Compression



Modeling



Low-dimensional model of variation of registered objects such as faces

$\bar{\mathbf{x}} + \alpha \mathbf{u}_1$

$\bar{\mathbf{x}} + \alpha \mathbf{u}_1$

$\bar{\mathbf{x}} + \alpha \mathbf{u}_1$

$\bar{\mathbf{x}} + \alpha \mathbf{u}_1$

PROBABILISTIC PCA

J. Elder

CSE 6390/PSYC 6225 Computational Modeling of Visual Perception

Probabilistic PCA

- PCA applies to data of any kind.
- But PCA can also be interpreted as the maximum likelihood solution to a probabilistic latent variable model based on a constrained form of the Gaussian distribution:

Let $\mathbf{z}$ be an M-dimensional hidden variable with Gaussian prior

Let $\mathbf{x}$ be the D-dimensional observed variable with Gaussian conditional:

$$p(\mathbf{z}) = N(\mathbf{z} \mid 0, \mathbf{I})$$

$$p(\mathbf{x} \mid \mathbf{z}) = N(\mathbf{x} \mid \mathbf{W}\mathbf{z} + \mu, \sigma^2 \mathbf{I})$$

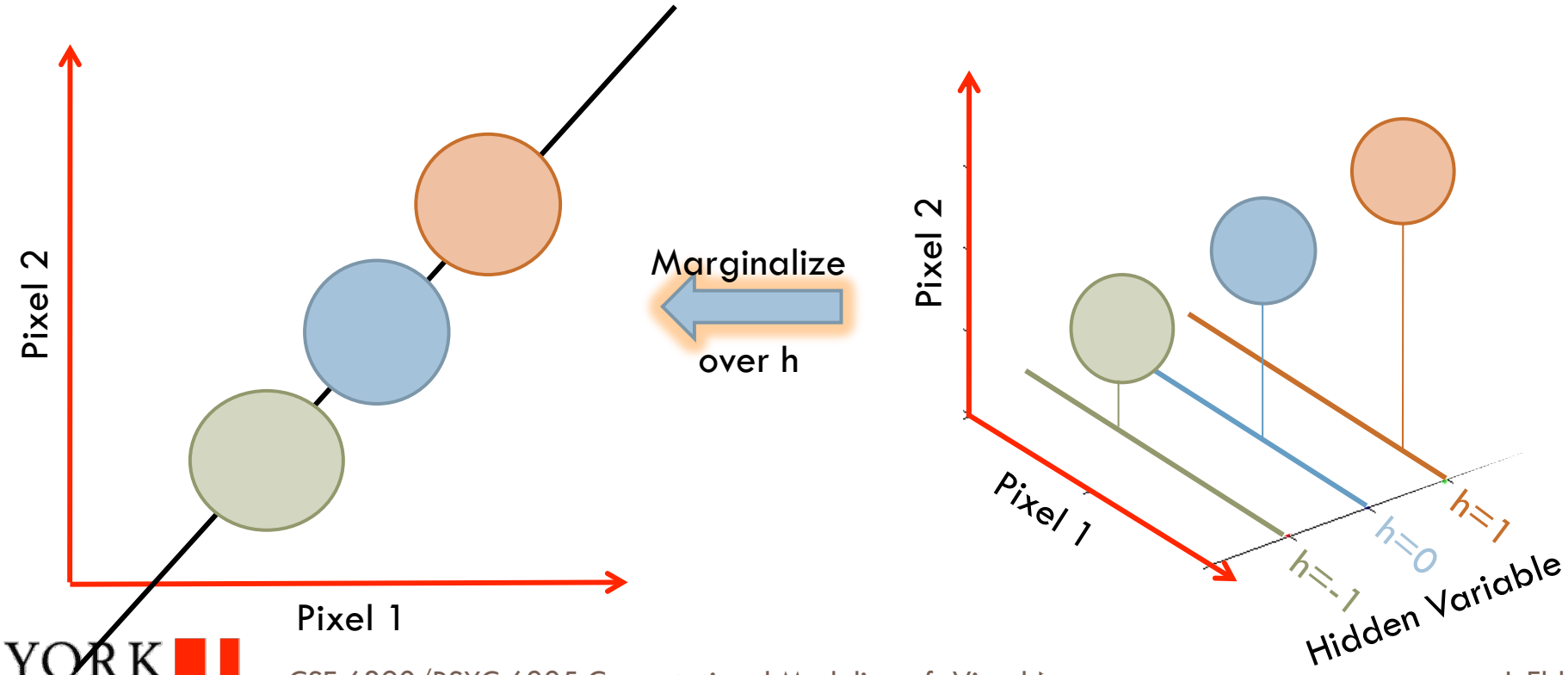
$D \times M$

Probabilistic PCA

- One way to see this is to think of Probabilistic PCA as the limit of a mixture of Gaussians model, as the number of Gaussian components → ∞:

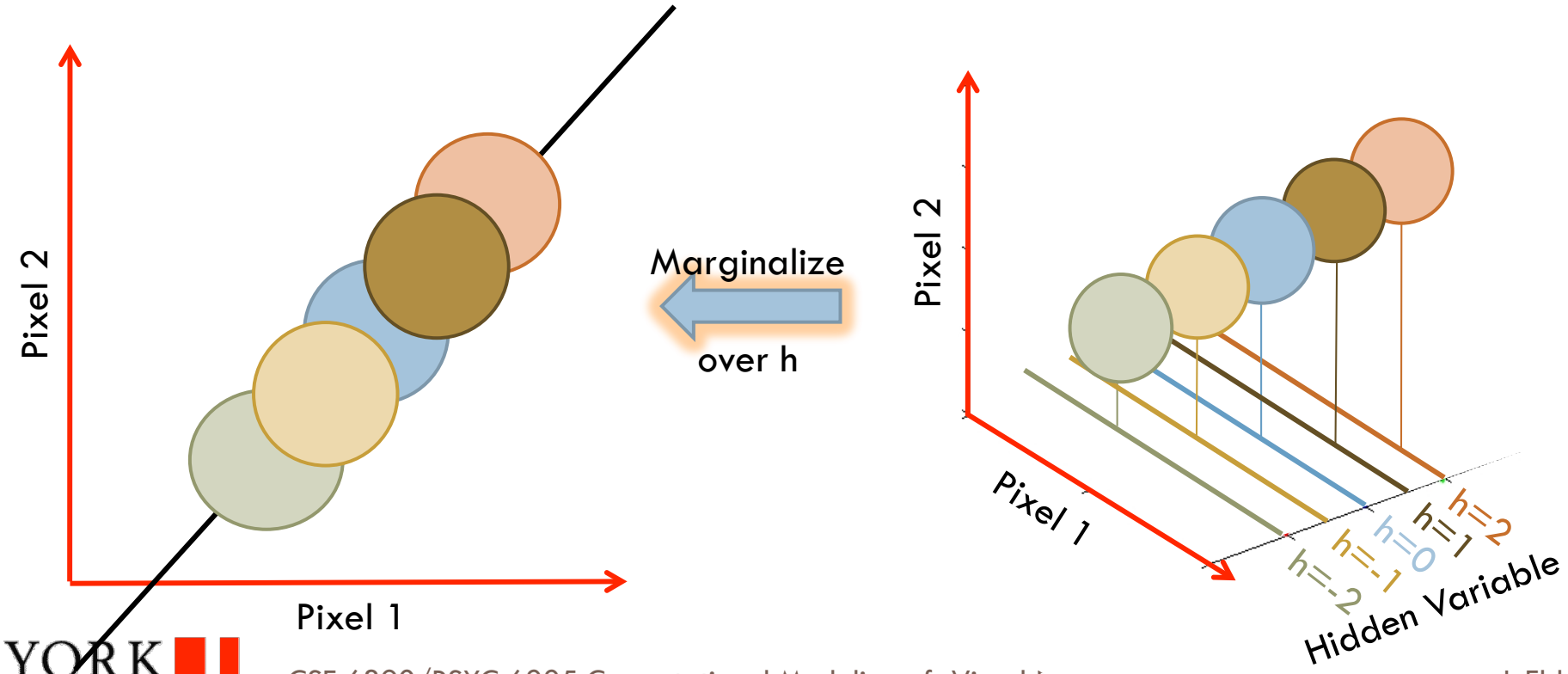
Probabilistic PCA

Consider putting the means of the Gaussians mixture components all on a line and forcing their diagonal covariances to be identical.
What happens if we keep adding more and more Gaussians along this line?



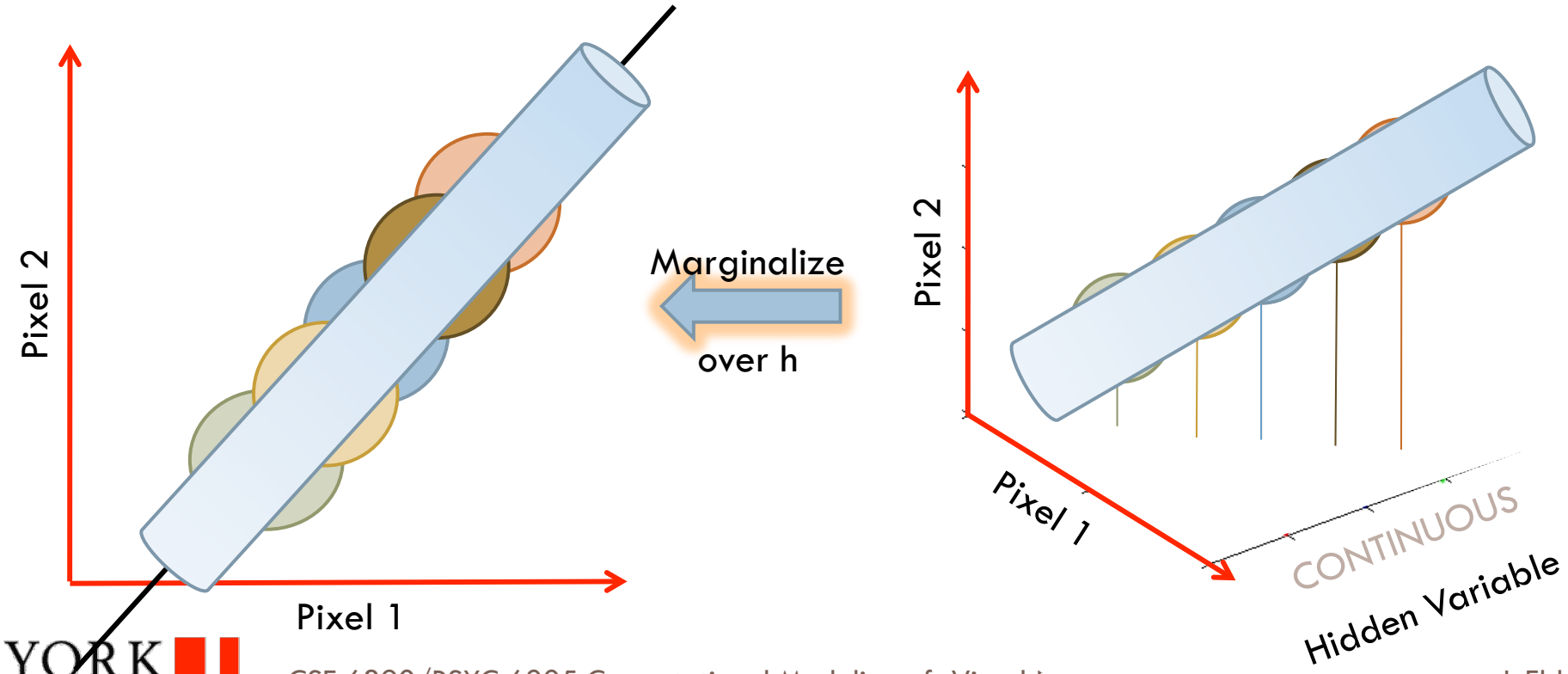
Probalistic PCA

Consider putting the means of the Gaussians mixture components all on a line and forcing their diagonal covariances to be identical.
What happens if we keep adding more and more Gaussians along this line? In the limit the hidden variable become continuous



Probabilistic PCA

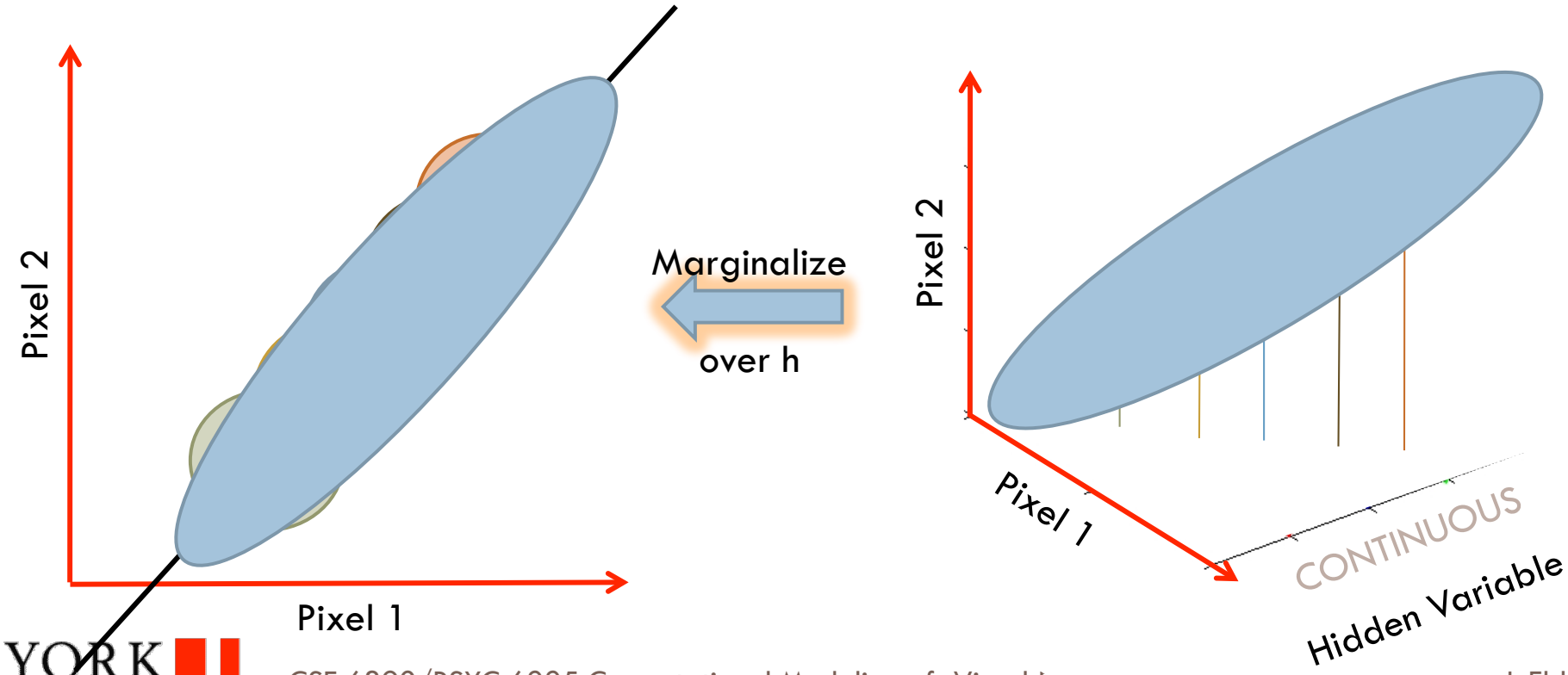
Consider putting the means of the Gaussians mixture components all on a line and forcing their diagonal covariances to be identical.
What happens if we keep adding more and more Gaussians along this line? In the limit the hidden variable become continuous



Now consider weighting the constituent Gaussians...

Probabilistic PCA

Consider putting the means of the Gaussians mixture components all on a line and forcing their diagonal covariances to be identical.
What happens if we keep adding more and more Gaussians along this line? In the limit the hidden variable become continuous



Probabilistic PCA

- PCA as the maximum likelihood solution to a probabilistic latent variable model based on a constrained form of the Gaussian distribution:

Let $\mathbf{z}$ be an M-dimensional hidden variable with Gaussian prior

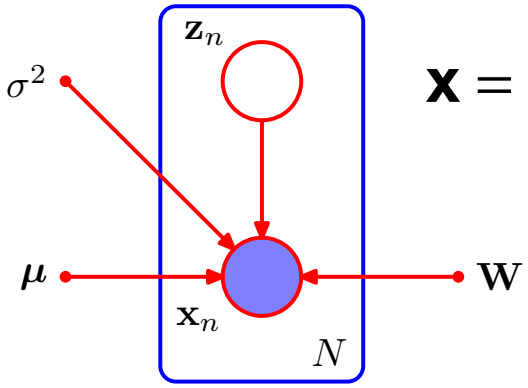
Let $\mathbf{x}$ be the D-dimensional observed variable with Gaussian conditional:

$$p(\mathbf{z}) = N(\mathbf{z} \mid 0, \mathbf{I})$$

$$p(\mathbf{x} \mid \mathbf{z}) = N(\mathbf{x} \mid \mathbf{W}\mathbf{z} + \mu, \sigma^2\mathbf{I})$$

$D \times M$

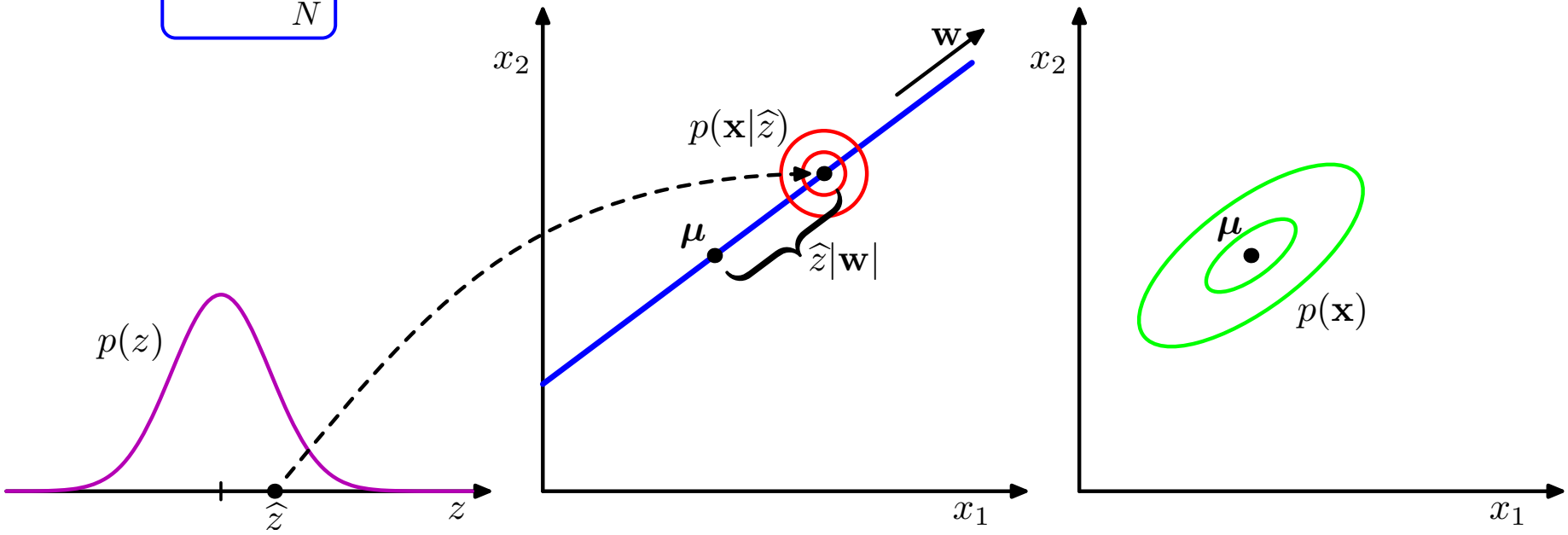
Generative Model



$$\mathbf{x} = \mathbf{W}\mathbf{z} + \mu + \varepsilon \longrightarrow$$

$$p(\mathbf{z}) = N(\mathbf{z} \mid 0, \mathbf{I})$$

$$p(\mathbf{x} \mid \mathbf{z}) = N(\mathbf{x} \mid \mathbf{W}\mathbf{z} + \mu, \sigma^2\mathbf{I})$$



Prior over latent variable | Subspace | Marginal density

Marginal Distribution for Probabilistic PCA

- The marginal distribution of the observed variable is

$$p(x) = N(\mathbf{x} \mid \mu, \mathbf{C})$$

where

$$\mathbf{C} = \mathbf{W}\mathbf{W}^t + \sigma^2\mathbf{I}$$

Maximum Likelihood PCA

$$\mu = \bar{\mathbf{x}}$$

$$\sigma_{ML}^2 = \frac{1}{D-M}\sum_{i=M+1}^{D}\lambda_i$$

$$\mathbf{W}_{ML} = \mathbf{U}_M\left(\mathbf{L}_M - \sigma^2\mathbf{I}\right)^{1/2}\mathbf{R}$$

where

$\mathbf{U}_M$ is a $D \times M$ matrix whose columns are given by any subset of size M of the eigenvectors of $\mathbf{S}$

$\mathbf{L}_M$ is an $M \times M$ diagonal matrix containing the M corresponding eigenvalues λ_i

$\mathbf{R}$ is an arbitrary rotation matrix

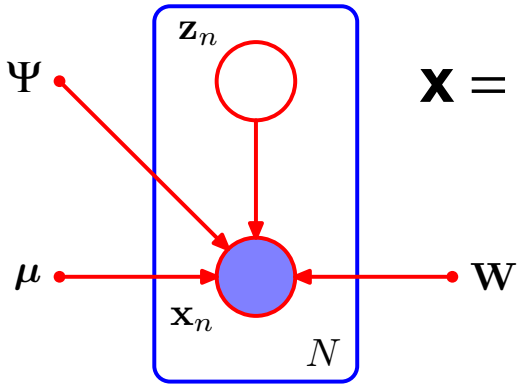
- ML parameters can be found either by
  - determining M eigenvectors and eigenvalues directly
  - EM $\quad \mathbf{x} = \mathbf{W}\mathbf{z} + \mu + \varepsilon$

FACTOR ANALYSIS

J. Elder

CSE 6390/PSYC 6225 Computational Modeling of Visual Perception

Generative Model

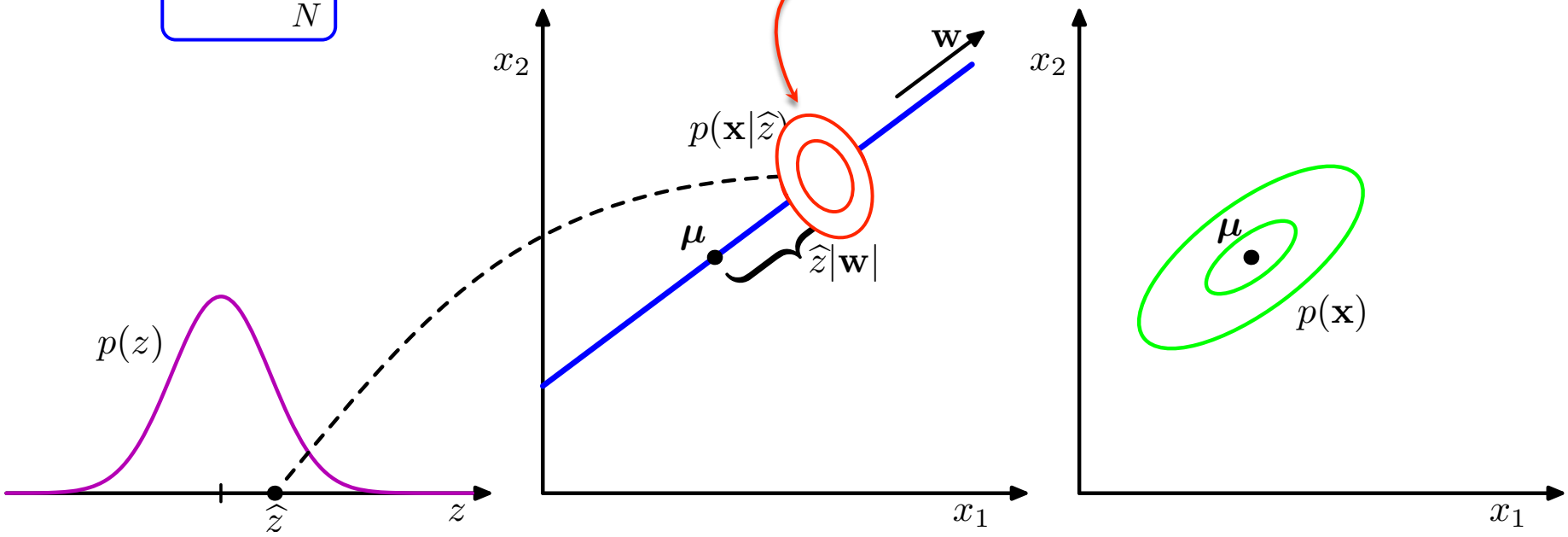


$$\mathbf{x} = \mathbf{W}\mathbf{z} + \mu + \varepsilon \longrightarrow$$

$$p(\mathbf{z}) = N(\mathbf{z} \mid 0, \mathbf{I})$$

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

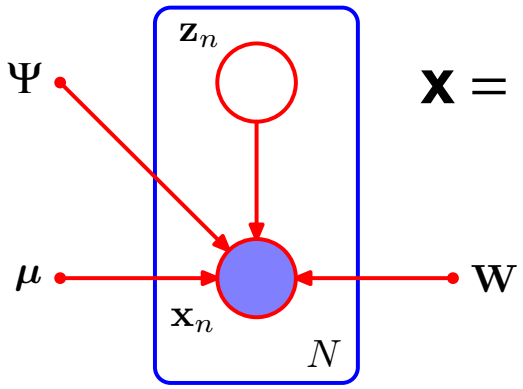
Ψ = Covariance of ε : diagonal but not isotropic



Prior over latent variable | Subspace | Marginal density

Factor Analysis Terminology

- Columns of W are called **factor loadings**
- Diagonal elements of Ψ are called **uniquenesses**



$$\mathbf{x} = \mathbf{W}\mathbf{z} + \mu + \varepsilon$$

Marginal Distribution for Factor Analysis

- The marginal distribution of the observed variable is

$$p(x) = N(\mathbf{x} \mid \mu, \mathbf{C})$$

where

$$\mathbf{C} = \mathbf{W}\mathbf{W}^t + \Psi$$

Maximum Likelihood Parameter Estimation

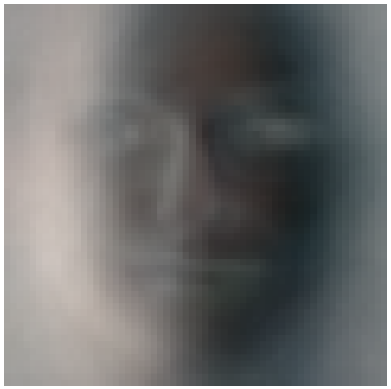
$\mu_{ML} = \bar{x}$

However, no closed-form solution for $\mathbf{W}_{ML}$. Instead, can use EM to find $\mathbf{W}_{ML}$ and Ψ_{ML}.

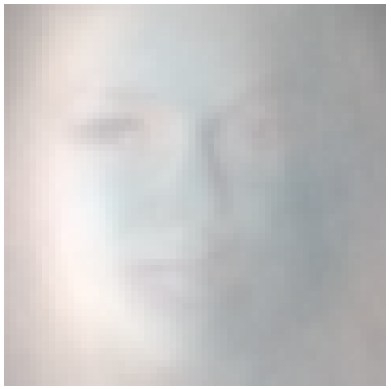
Learning Results: Two Factor Model

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \mu + \varepsilon \quad \text{where } \mathbf{W} = \left[\mathbf{w}_1\ \mathbf{w}_2 \ldots \mathbf{w}_D\right]$$

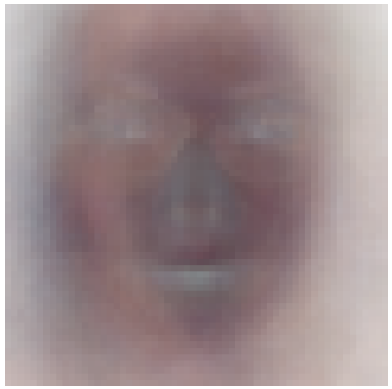
μ



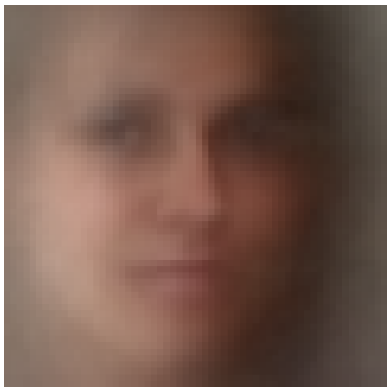
$\mathbf{w}_1$



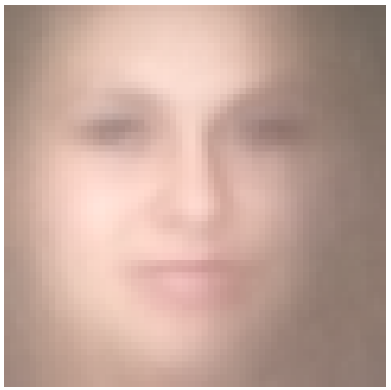
$\mathbf{w}_2$



Ψ



$\mu + 2\mathbf{w}_1$



$\mu + 2\mathbf{w}_2$

Learning results: Two Factor Model

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \mu + \varepsilon \quad \text{where } \mathbf{W} = \left[\mathbf{w}_1\ \mathbf{w}_2 \ldots \mathbf{w}_D\right]$$

μ

$\mathbf{w}_1$

$\mathbf{w}_2$

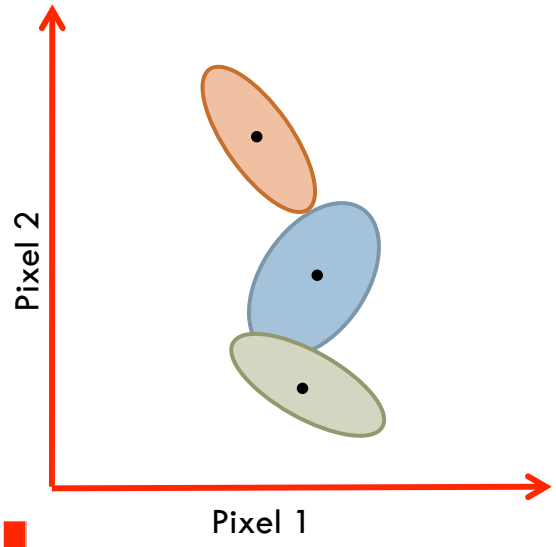
Ψ

$\mu + 2\mathbf{w}_1$

$\mu + 2\mathbf{w}_2$

Non-Linear Extensions

- Mixture of factor analyzers (MOFA)
  - Two levels of the EM algorithm
  - One to learn each factor analyzer
  - One to learn the mixture model
  - Can describe quite complex manifold structures in high dimensions with only a limited number of parameters



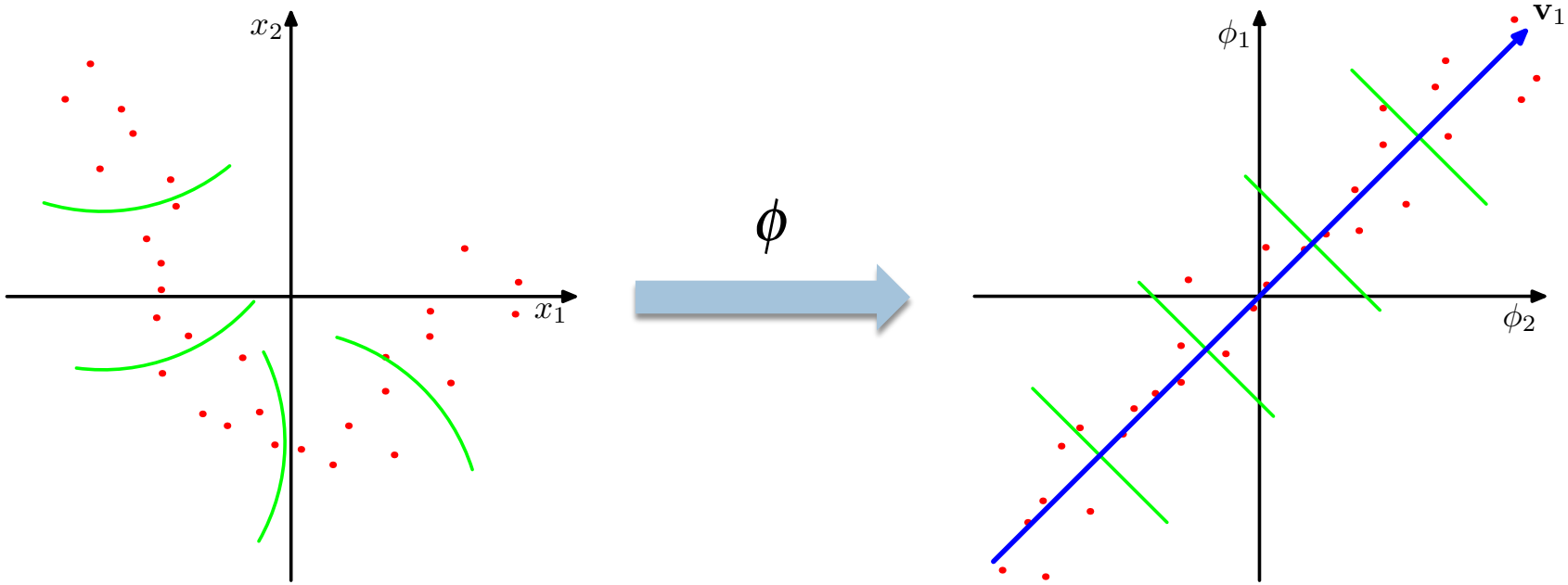
$$p(x) = \sum_{k=1}^{K} N(\mathbf{x} \mid \mu_k, \mathbf{C}_k)$$

where

$$\mathbf{C}_k = \mathbf{W}_k \mathbf{W}_k^t + \Psi_k$$

Non-Linear Extensions

- Kernel PCA
  - Idea:
    - Use a non-linear mapping ϕ to an M-dimensional 'feature space'
    - Now perform PCA in this new space



Kernel PCA

Assume 0-mean data vectors: $\sum_{n=1}^{N} \mathbf{x}_n = 0.$

Covariance in feature space $C = \frac{1}{N}\sum_{n=1}^{N} \phi(x_n)\phi(x_n)^t$

Eigenvector expansion $Cv_i = \lambda v_i$

$$\rightarrow \frac{1}{N}\sum_{n=1}^{N} \phi(x_n)\left\{\phi(x_n)^t v_i\right\} = \lambda_i v_i$$

M-dimensional eigenvector

Thus the eigenvector v_i is a linear combination of the transformed data vectors $\phi(x_n)$:

$$v_i = \sum_{n=1}^{N} a_{in}\phi(x_n)$$

Substituting, we have $\frac{1}{N}\sum_{n=1}^{N} \phi(x_n)\phi(x_n)^t \sum_{m=1}^{N} a_{im}\phi(x_m) = \lambda_i \sum_{n=1}^{N} a_{in}\phi(x_n)$

Now multiplying both sides by $\phi(x_l)^t$, we obtain

$$\frac{1}{N}\sum_{n=1}^{N} \phi(x_l)^t \phi(x_n) \sum_{m=1}^{N} a_{im}\phi(x_n)^t \phi(x_m) = \lambda_i \sum_{n=1}^{N} a_{in}\phi(x_l)^t \phi(x_n)$$

Finally, defining the kernel function $k(x_n, x_m) = \phi(x_n)^t \phi(x_m)$, we can write

$$\frac{1}{N}\sum_{n=1}^{N} k(x_l, x_n) \sum_{m=1}^{N} a_{im}k(x_n, x_m) = \lambda_i \sum_{n=1}^{N} a_{in}k(x_l, x_n)$$

Kernel PCA

$$\frac{1}{N}\sum_{n=1}^{N} k(x_l, x_n)\sum_{m=1}^{N} a_{im} k(x_n, x_m) = \lambda_i \sum_{n=1}^{N} a_{in} k(x_l, x_n)$$

N-dimensional eigenvector

or, in matrix notation, $\mathbf{K}^2\mathbf{a}_i = \lambda_i N\mathbf{K}\mathbf{a}_i \rightarrow \mathbf{K}\mathbf{a}_i = \lambda_i N\mathbf{a}_i$

Requiring that the eigenvectors $\mathbf{v}_i$ in feature space be unit vectors leads to the constraint $|\mathbf{a}_i|^2 = \frac{1}{N\lambda_i}$.

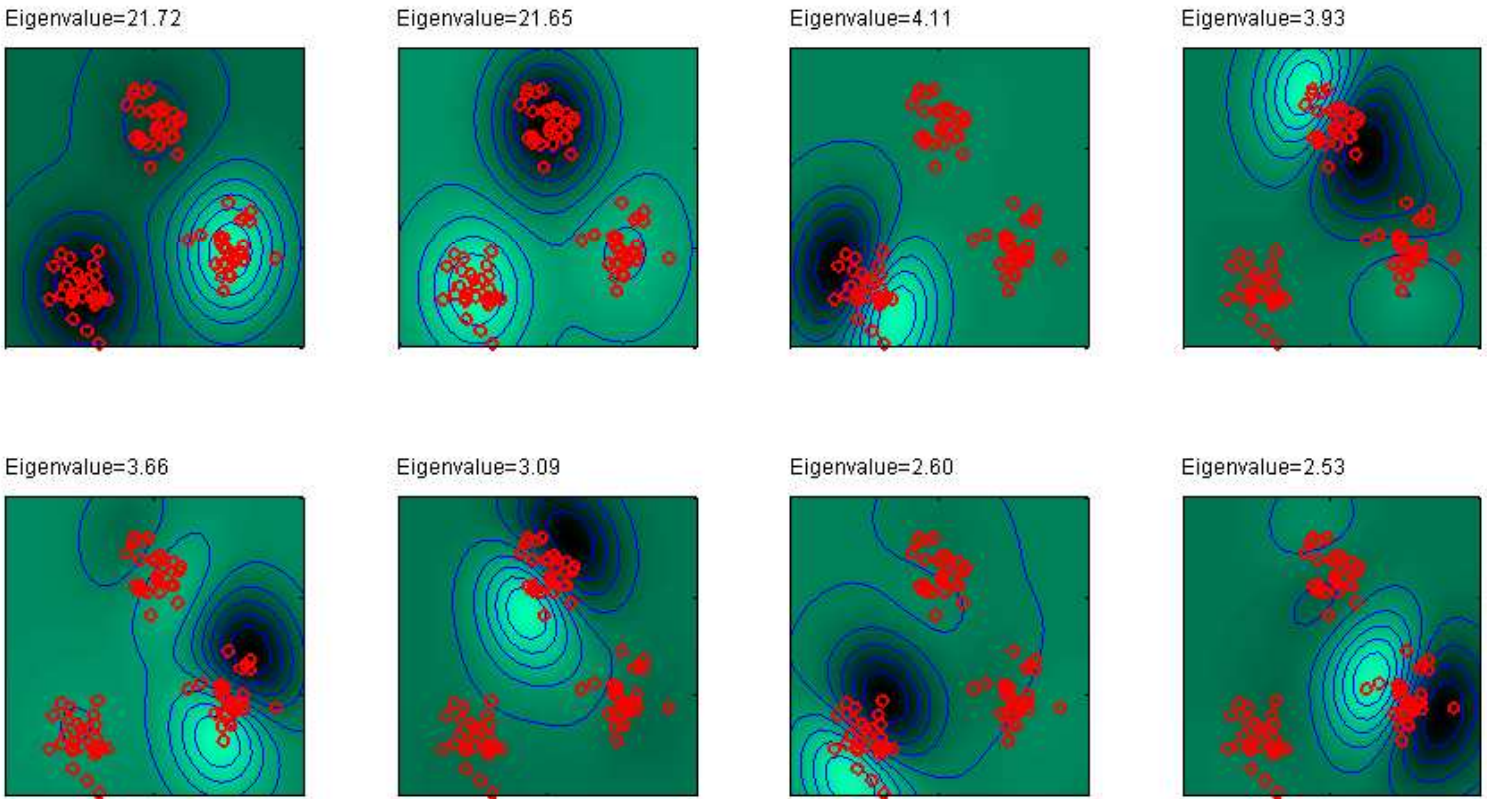
In practice, for the projected data to have 0 mean, use $\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_N\mathbf{K} - \mathbf{K}\mathbf{1}_N + \mathbf{1}_N\mathbf{K}\mathbf{1}_N$

where

$$\mathbf{1}_N = \begin{pmatrix} \frac{1}{N} & \cdots & \frac{1}{N} \\ \vdots & \ddots & \vdots \\ \frac{1}{N} & \cdots & \frac{1}{N} \end{pmatrix}$$

N (vertical), N (horizontal)

Kernel PCA


Eigenvalue=21.72
Eigenvalue=21.65
Eigenvalue=4.11
Eigenvalue=3.93
Eigenvalue=3.66
Eigenvalue=3.09
Eigenvalue=2.60
Eigenvalue=2.53