

EECS 545: Machine Learning

Lecture 18. Unsupervised Learning: PCA

Honglak Lee

3/16/2011

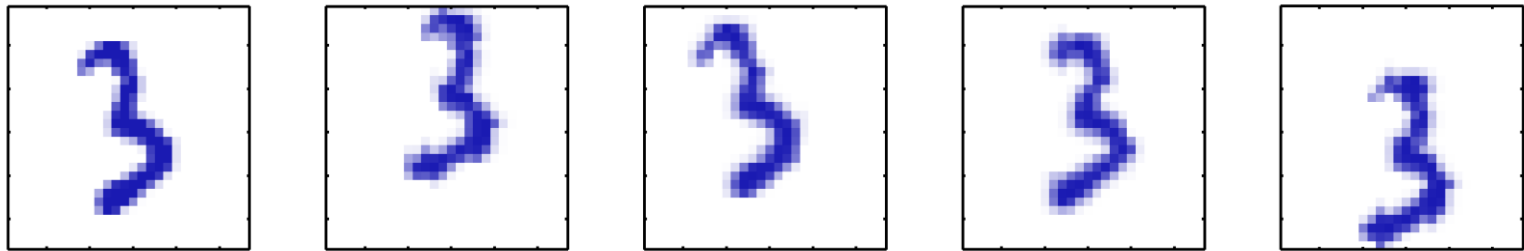


Outline

- Principal Component Analysis

High-Dimensional Data

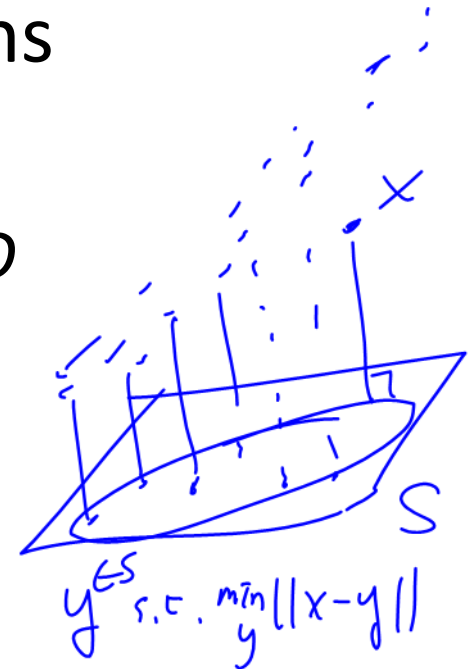
- . . . may have low-dimensional structure.



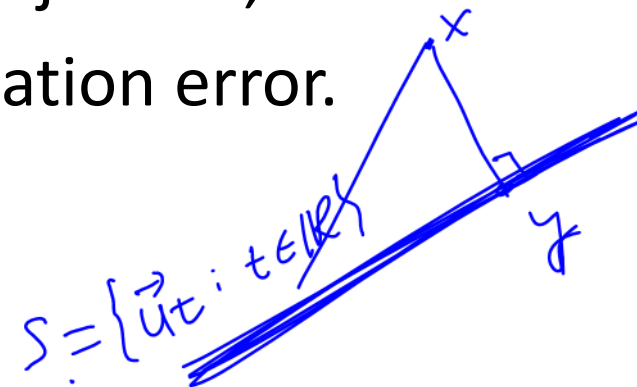
- The data is 100x100-dimensional.
- But there are only three degrees of freedom, so it lies on a 3-dimensional subspace.
 - (on a non-linear manifold, in this case)

Principal Component Analysis

- Given a set $X = \{x_n\}$ of observations
 - in a space of dimension D ,
 - find a subspace of dimension $M < D$
 - that captures most of its variability.



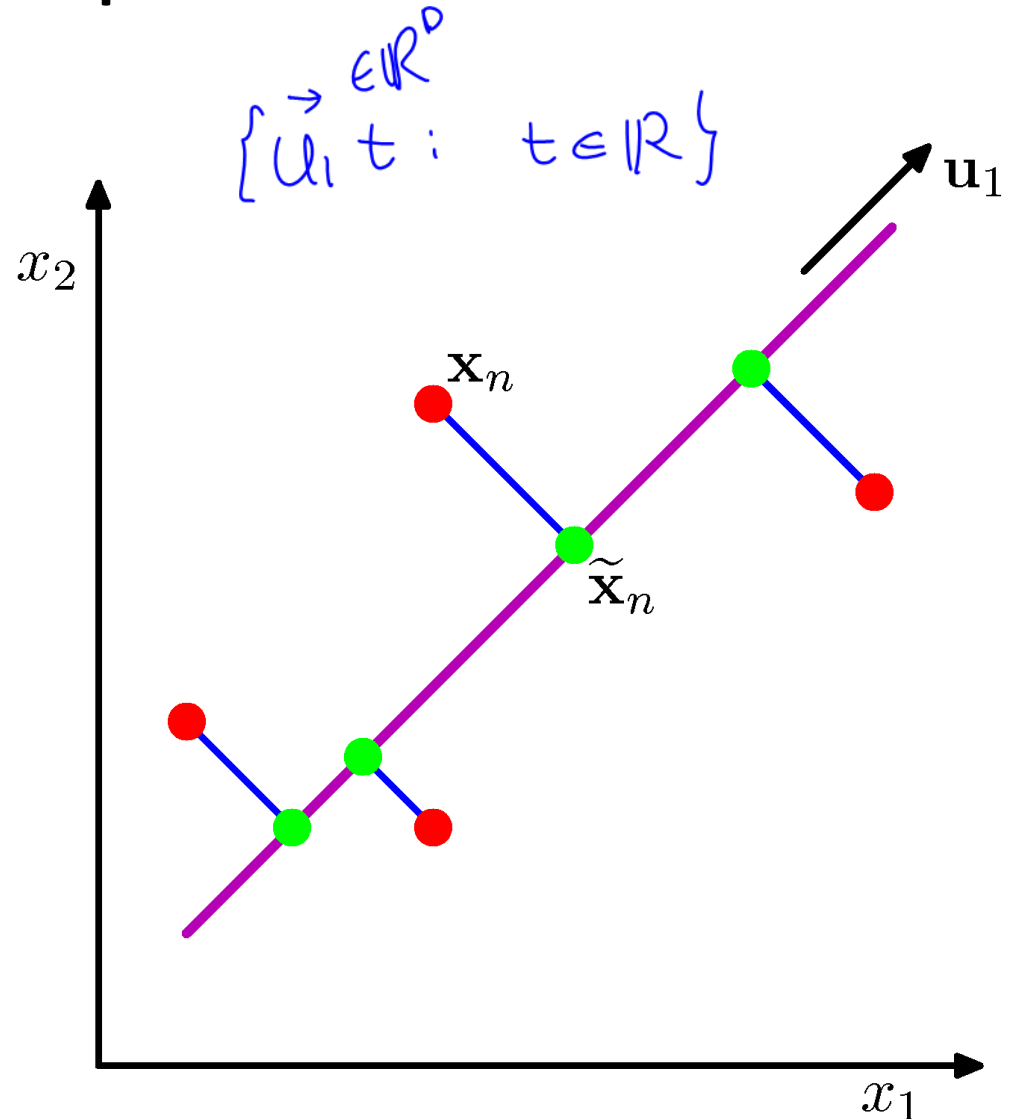
- PCA can be described as either:
 - maximizing the variance of the projection, or
 - minimizing the squared approximation error.



Two Descriptions of PCA

Approximate with the projection:

- Maximize variance, or
- Minimize squared error



Equivalent Descriptions

- With mean at the origin

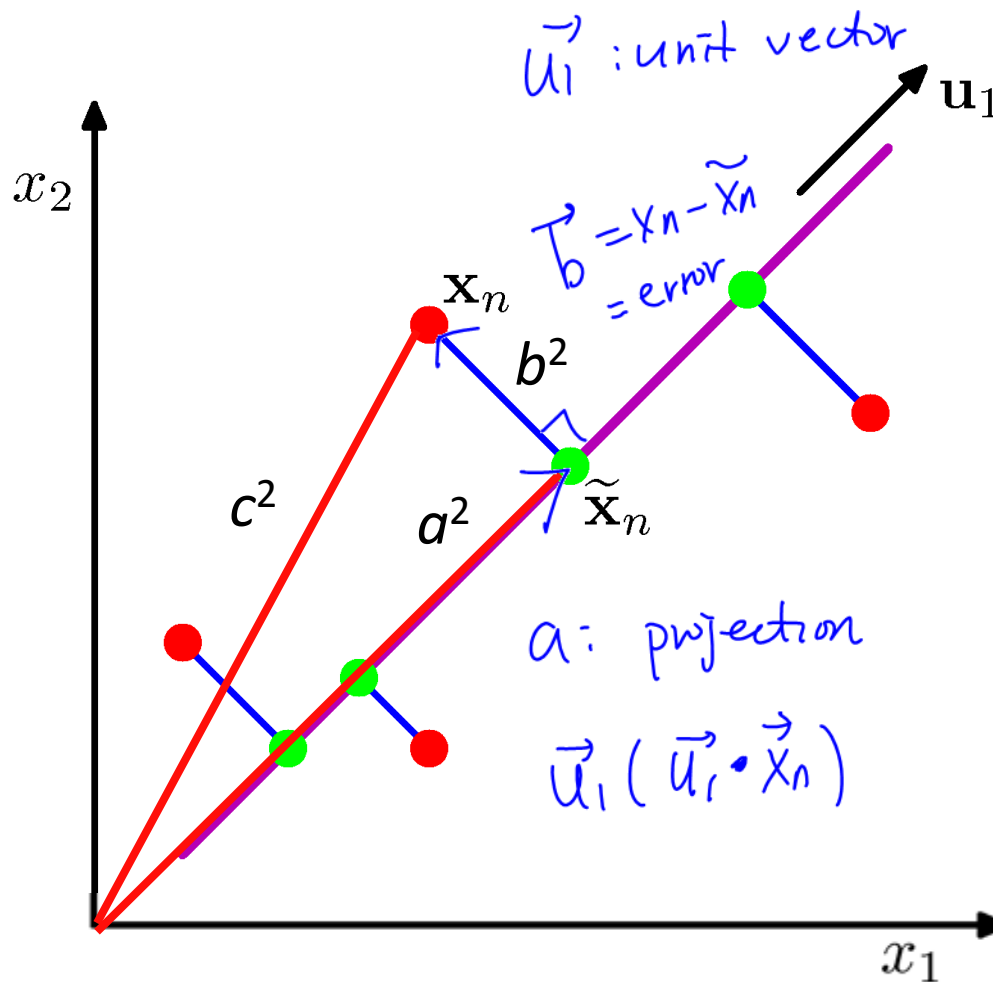
$$\sum_i c_i^2 = \sum_i a_i^2 + \sum_i b_i^2$$

- With constant $\sum_i c_i^2$

- Minimizing $\sum_i b_i^2$

- Maximizes $\sum_i a_i^2$

- And vice versa



First Principal Component

- Given data points $\{\mathbf{x}_n\}$ in D-dim space.

- Mean $\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n$ $\mathbb{E}_{\text{emp}}[(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T]$
- Data covariance $\mathbf{S} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T$
 - D x D matrix

- Let \mathbf{u}_1 be the principal component we want.

- Length 1: $\mathbf{u}_1^T \mathbf{u}_1 = 1$

- Projection of \mathbf{x}_n : $\mathbf{u}_1^T \mathbf{x}_n$
 $\vec{u}_1 (\vec{u}_1^T \mathbf{x}_n)$

First Principal Component

$$\mathbb{E}_{\text{emp}}[(s - \bar{s})^2] \quad s_n = \mathbf{u}_1^T \mathbf{x}_n$$

- Maximize the projection variance:

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

- Use a Lagrange multiplier to enforce $\mathbf{u}_1^T \mathbf{u}_1 = 1$
- Maximize: $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1)$
- Derivative is zero when $\mathbf{S} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1$
 - That is $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 = \lambda_1$
- So \mathbf{u}_1 is eigenvector with largest eigenvalue.

$$\frac{1}{N} \sum_{n=1}^N (u_1^T x_n - u_1^T \bar{x})^2$$

$$= \frac{1}{N} \sum_n \left[\overset{||}{\underbrace{u_1^T (x_n - \bar{x})}_{\text{Scalar}}} \right]^2$$

$$= \frac{1}{N} \sum_n (u_1^T (x_n - \bar{x})) (u_1^T (x_n - \bar{x}))^T$$

$$= \frac{1}{N} \sum_n u_1^T (x_n - \bar{x}) (x_n - \bar{x})^T u_1$$

$$= u_1^T \underbrace{\left[\frac{1}{N} \sum_n (x_n - \bar{x}) (x_n - \bar{x})^T \right]}_S u_1$$

$$u_1 \in \mathbb{R}^D$$

$$\max_{u_1} u_1^T S u_1$$

$$\text{s.t.} \quad u_1^T u_1 - 1 = 0$$

Lagrange multiplier.

$$\max_{u_1} \frac{1}{2} (u_1^T S u_1 - \lambda (u_1^T u_1 - 1))$$

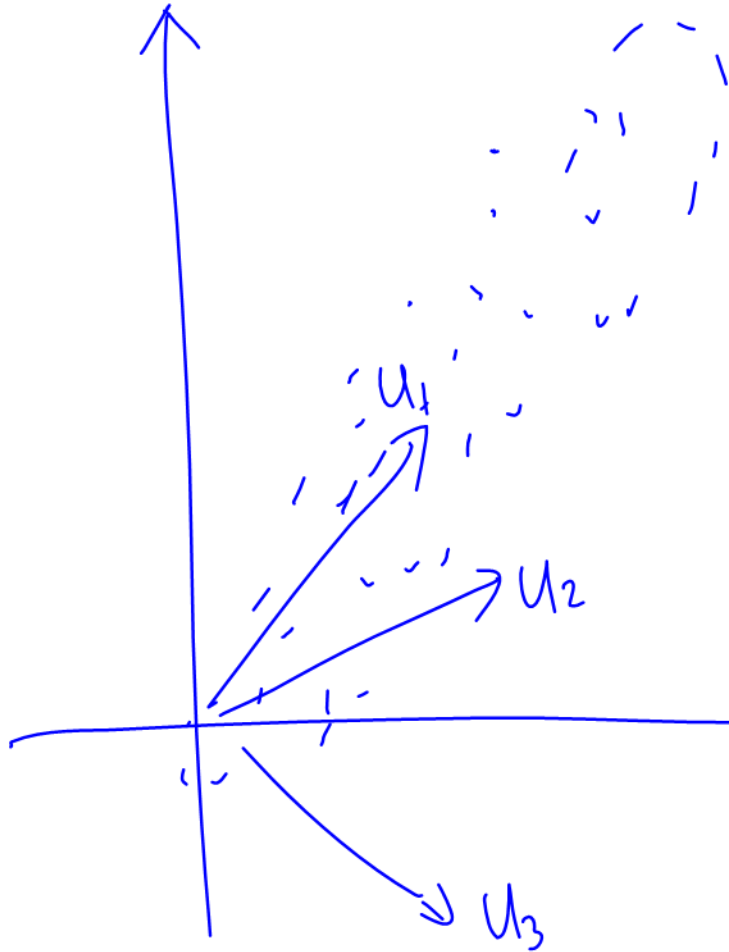
$\mathcal{L}(u_1, \lambda)$

$$\nabla_{u_1} \mathcal{L}$$

$$= S u_1 - \lambda u_1 = 0$$

$$\Rightarrow S u_1 = \lambda u_1$$

$$\text{s.t.} \quad u_1^T u_1 = 1$$

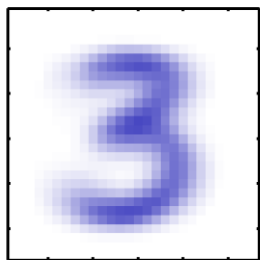
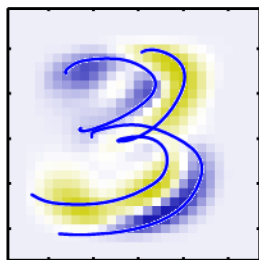
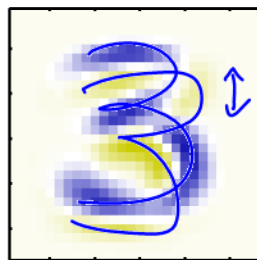
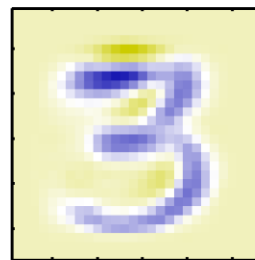
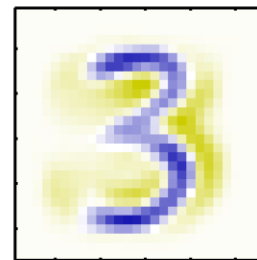


PCA by Maximizing Variance

- Repeat to find the M eigenvectors of the data covariance matrix S corresponding to the M largest eigenvalues.
- We can do the same thing by minimizing the squared error of the projection.
 - Homework

3 3 3 \longrightarrow 10000

- Mean


$$\lambda_1 = 3.4 \cdot 10^5$$

$$\lambda_2 = 2.8 \cdot 10^5$$

$$\lambda_3 = 2.4 \cdot 10^5$$

$$\lambda_4 = 1.6 \cdot 10^5$$


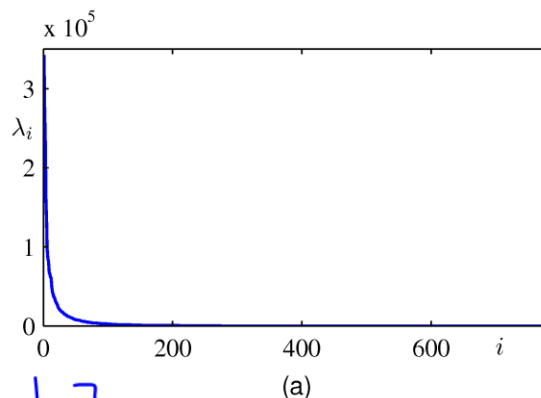
- $$X = \begin{bmatrix} \overleftarrow{\overrightarrow{x_1}} \\ \overleftarrow{\overrightarrow{x_2}} \\ \vdots \\ \overleftarrow{\overrightarrow{x_p}} \end{bmatrix} \begin{matrix} \updownarrow \\ N \end{matrix}$$

$$X = X - \text{mean}(X)$$

$$C = \frac{1}{N} X^T X$$

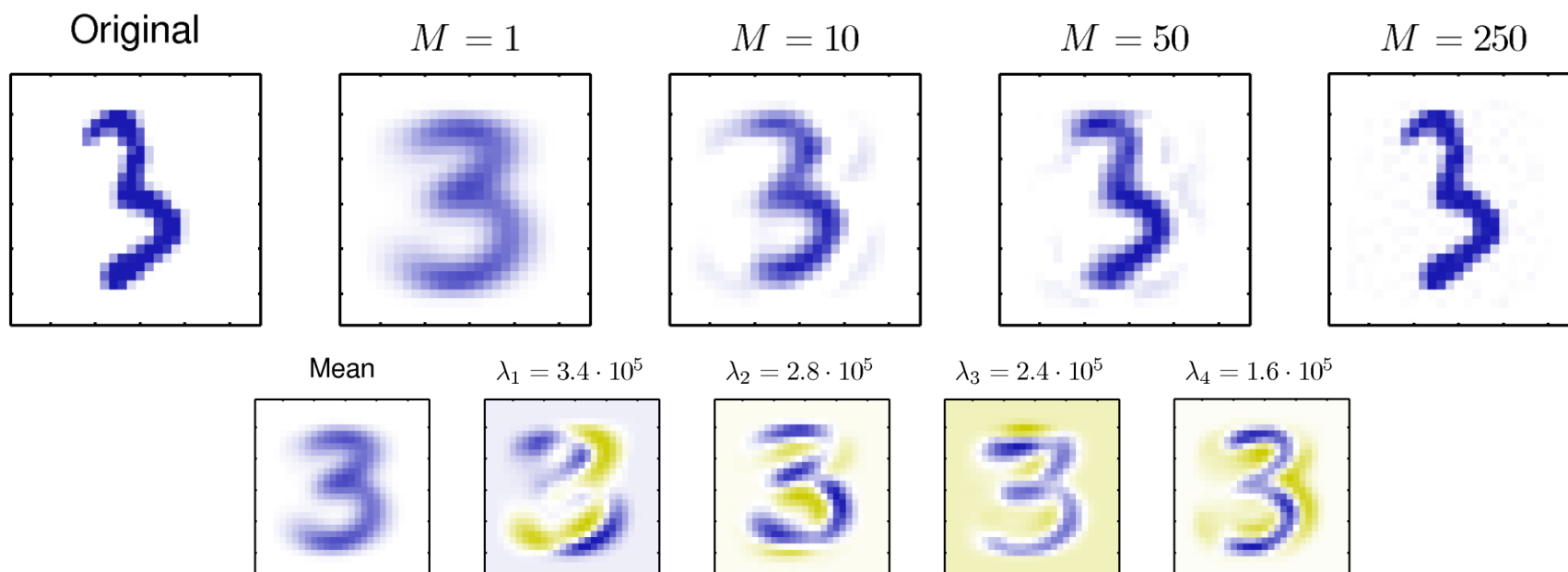
$$[VD] = \text{eig}(C)$$

$$\Rightarrow V = \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}$$



Reconstructing the Image

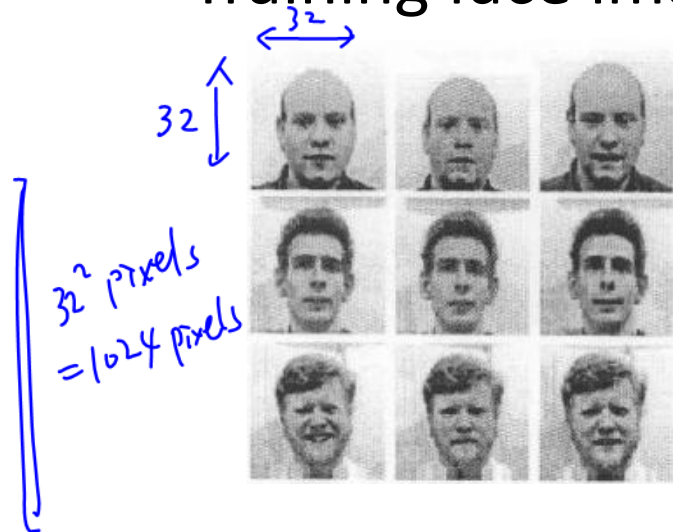
- Compress the image representation by using only first M eigenvectors, and discarding the less important information.



Learning features via PCA

- Example: Eigenfaces

Training face images



Learned PCA bases


$$X \approx \vec{a}_1^* + \vec{a}_2^* + \vec{a}_3^* + \vec{a}_4^*$$

Test example


$$= 0.9571 * \vec{a}_1^* - 0.1945 * \vec{a}_2^* + 0.0461 * \vec{a}_3^* + 0.0586 * \vec{a}_4^*$$

PCA

$$PC_i = U_i^T X$$

PCA whitening

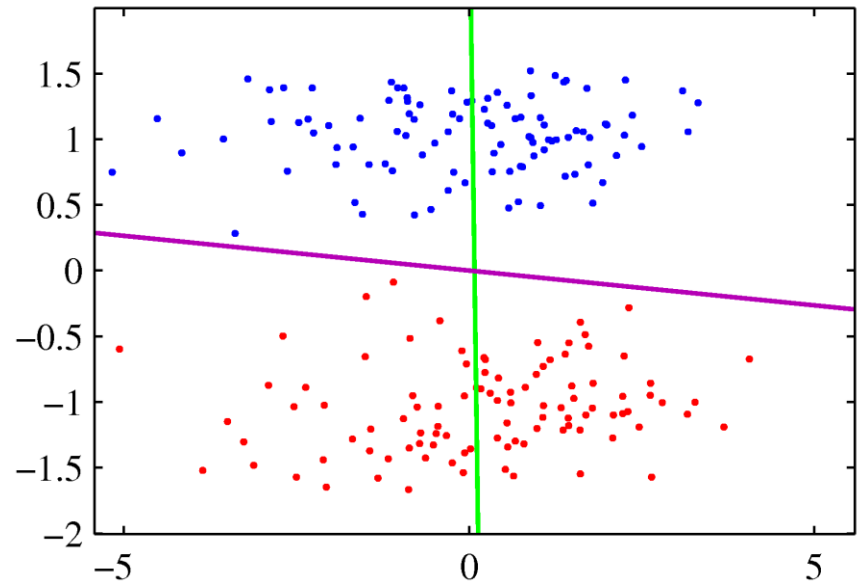
$$PC_whiten \bar{i} = \frac{1}{\sqrt{\lambda_i}} U_i^T X$$
$$\frac{1}{\sqrt{\lambda_i + \epsilon}}$$

λ_i = ^{ith} eigenvalue of C .

$$\epsilon = 0.01$$

Limits to PCA

- Maximizing variance is not always the best way to make the structure visible.
- PCA vs Fisher's linear discriminant



Probabilistic PCA

$$\text{PCA: } \mathbf{x} \approx \mathbf{W}\mathbf{z} \quad \mathbf{W}_j \text{ are orthogonal} \quad \mathbf{z} = \mathbf{W}^T \mathbf{x}$$

- We can view PCA as solving a probabilistic latent variable problem.
- Describe a distribution $p(\mathbf{x})$ in D -dimensional space, in terms of a latent variable \mathbf{z} in M -dimensional space.

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \mu + \epsilon \quad p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$$

- \mathbf{W} is a $D \times M$ linear transformation from \mathbf{z} to \mathbf{x}

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

$p(\mathbf{z}|\mathbf{x})$

Probabilistic PCA

$$x \approx \mathbf{W}z$$

- Given the generative model

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

$$\left. \begin{array}{l} p(z) \\ p(x|z) \end{array} \right\} \text{gaussian}$$

- we can infer

$$\downarrow$$

$$\left. \frac{p(x)}{p(z|x)} \right\} \text{gaussian}$$

$$E[\mathbf{x}] = E[(\mathbf{W}\mathbf{z} + \mu + \epsilon)] = \mu$$

$$= E[(\mathbf{x} - E[\mathbf{x}]) (\mathbf{x} - E[\mathbf{x}])^T]$$

$$\text{cov}[\mathbf{x}] = E[(\mathbf{W}\mathbf{z} + \epsilon)(\mathbf{W}\mathbf{z} + \epsilon)^T]$$

$$= E[\mathbf{W}\mathbf{z}\mathbf{z}^T\mathbf{W}^T] + E[\epsilon\epsilon^T] = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}$$

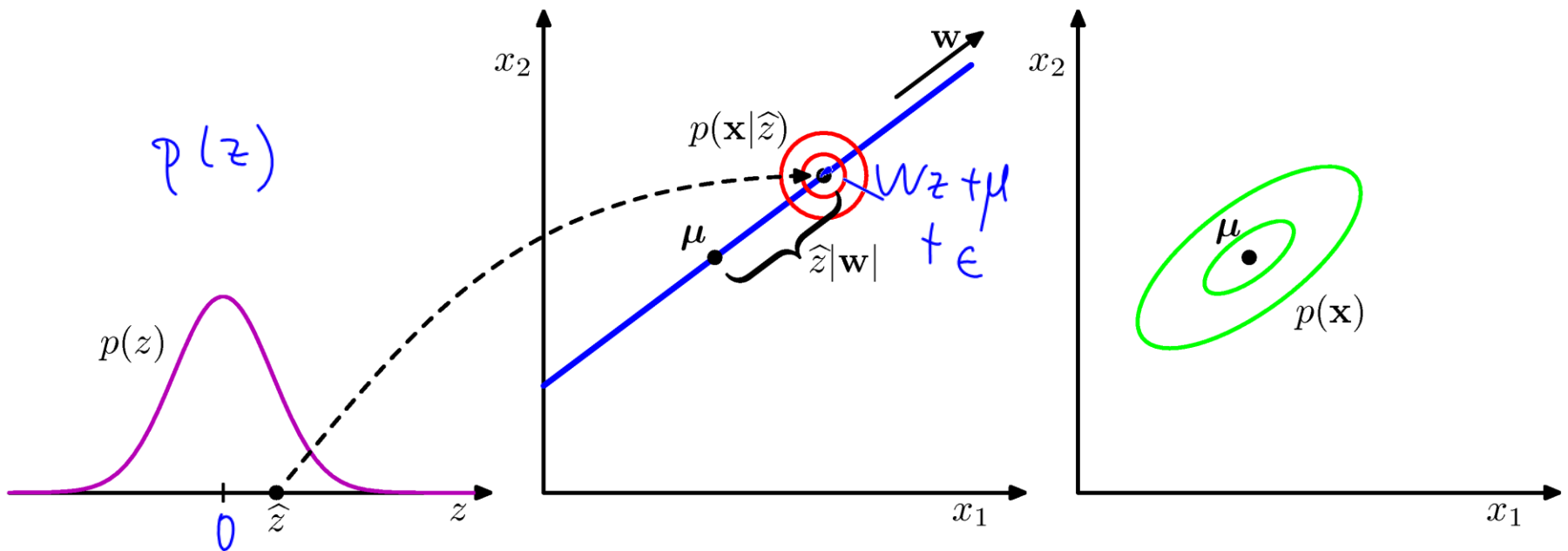
Q. Verify this

Probabilistic PCA

- The generative model

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

- can be illustrated



Likelihood of Probabilistic PCA

- (Marginal) likelihood $\sum_z p(x, z | \mu, W, \sigma^2)$

$$\begin{aligned} \mathcal{L} & \stackrel{\text{def}}{=} \ln p(X | \mu, W, \sigma^2) = \sum_n \ln p(x_n | W, \mu, \sigma^2) \\ &= -\frac{ND}{2} \ln 2\pi - \frac{N}{2} \ln |C| - \frac{1}{2} \sum_n (x_n - \mu)^T C^{-1} (x_n - \mu) \end{aligned}$$

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

- We can simply maximize this likelihood function with respect to μ, W, σ .

$$\textcircled{1} \quad \text{grad. desc.} \quad \mu_i = \mu + \lambda \nabla_{\mu} \mathcal{L}$$

$$W_i = W + \lambda \nabla_W \mathcal{L}$$

$$\sigma_i = \sigma + \lambda \nabla_{\sigma} \mathcal{L}$$

$\textcircled{2}$

$$\nabla_{\mu} \mathcal{L} = 0$$

$$\nabla_W \mathcal{L} = 0$$

$$\nabla_{\sigma} \mathcal{L} = 0$$

Maximum Likelihood Parameters

- Mean: $\mu = \bar{\mathbf{x}}$
- Noise: $\sigma_{ML}^2 = \frac{1}{D-M} \sum_{i=M+1}^D \lambda_i$
- W: $\mathbf{W}_{ML} = \mathbf{U}_M (\mathbf{L}_M - \sigma^2 \mathbf{I})^{1/2} \mathbf{R}$
 - where \mathbf{L}_M is diag with the M largest eigenvalues
 - and \mathbf{U}_M is the M corresponding eigenvectors
 - And \mathbf{R} is an arbitrary $M \times M$ rotation

Maximum likelihood by EM

- Latent variable model

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

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

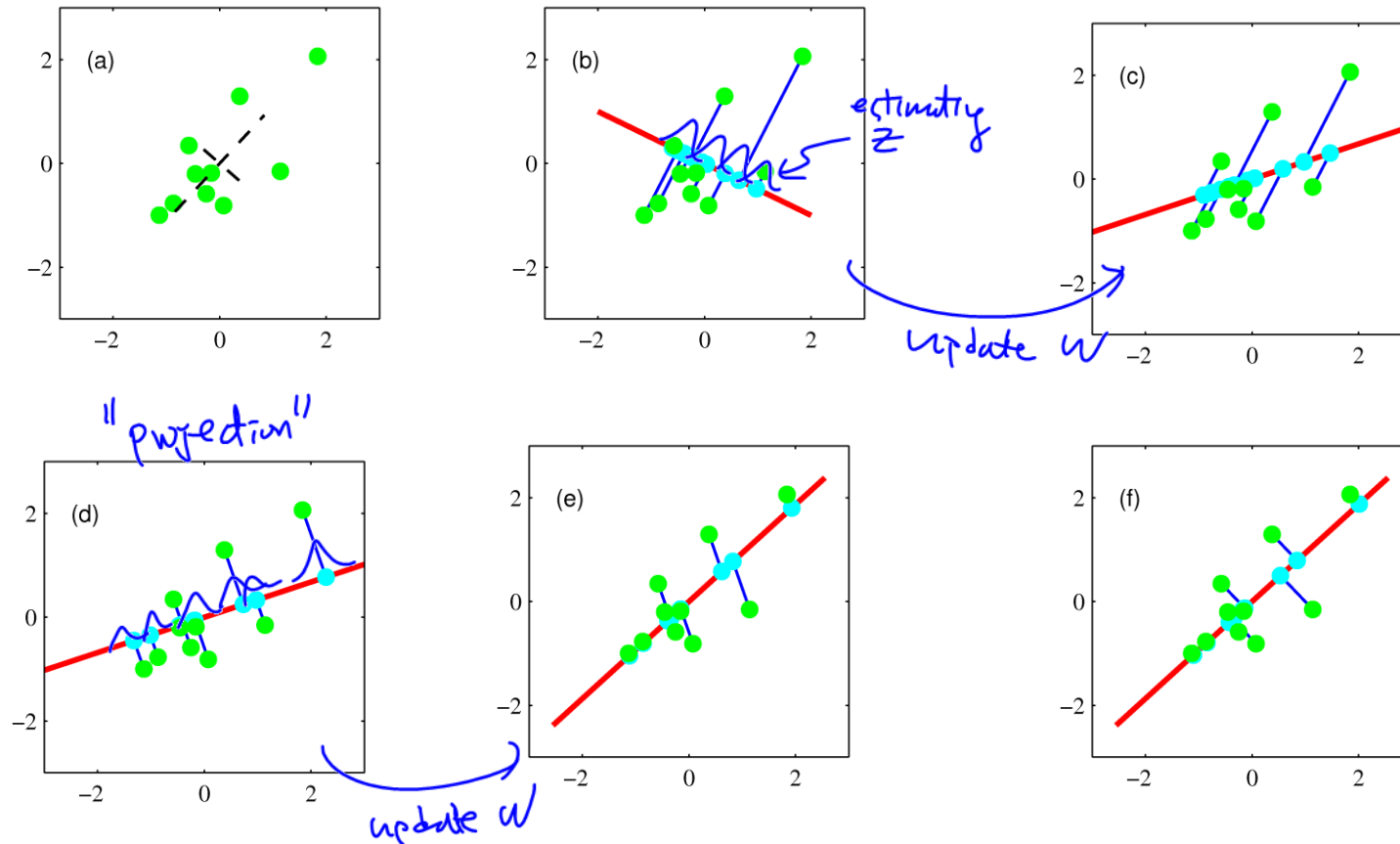
• $P_{\theta}(x, z)$ z : latent var.
 Goal: $\max_{\theta} \log P_{\theta}(x)$
 E step: $Q(z) = P_{\theta}(z | x)$
 M step: $\max_{\theta} \sum_{n=1}^N \sum_z Q(z^{(n)}) \log P(x^{(n)}, z^{(n)})$

- E-step: Estimate the posterior $Q(z) = P(z | x)$
 - Use linear Gaussian
- M-step: Maximize the data-completion likelihood given $Q(z)$:

$$\text{maximize}_{\theta=\{\mu, W, \sigma\}} \sum_i \sum_{z^{(i)}} Q(z^{(i)}) \log P_{\theta}(x^{(i)}, z^{(i)})$$

$\begin{matrix} \text{val} & \text{val pa} \\ P(x_i | P_a x_i) = \end{matrix}$
 $\frac{\text{counts}[\text{val}, \text{val pa}]}{\text{counts}[\text{val pa}]}$

Finding PCA params by EM



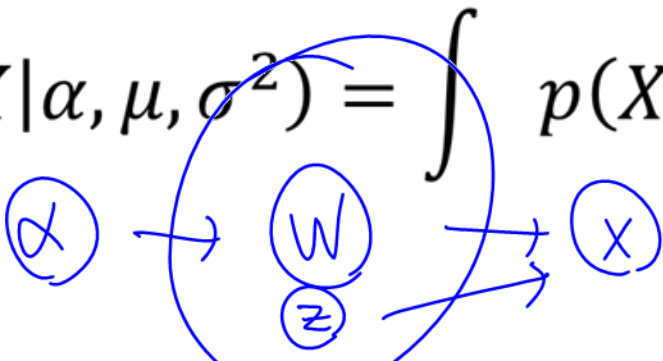
- Illustrating EM on simulated data

Bayesian PCA (sketch)

- Note that the maximum likelihood for probabilistic PCA is still a point estimate on W .
- Main idea of Bayesian PCA: Put a prior on W

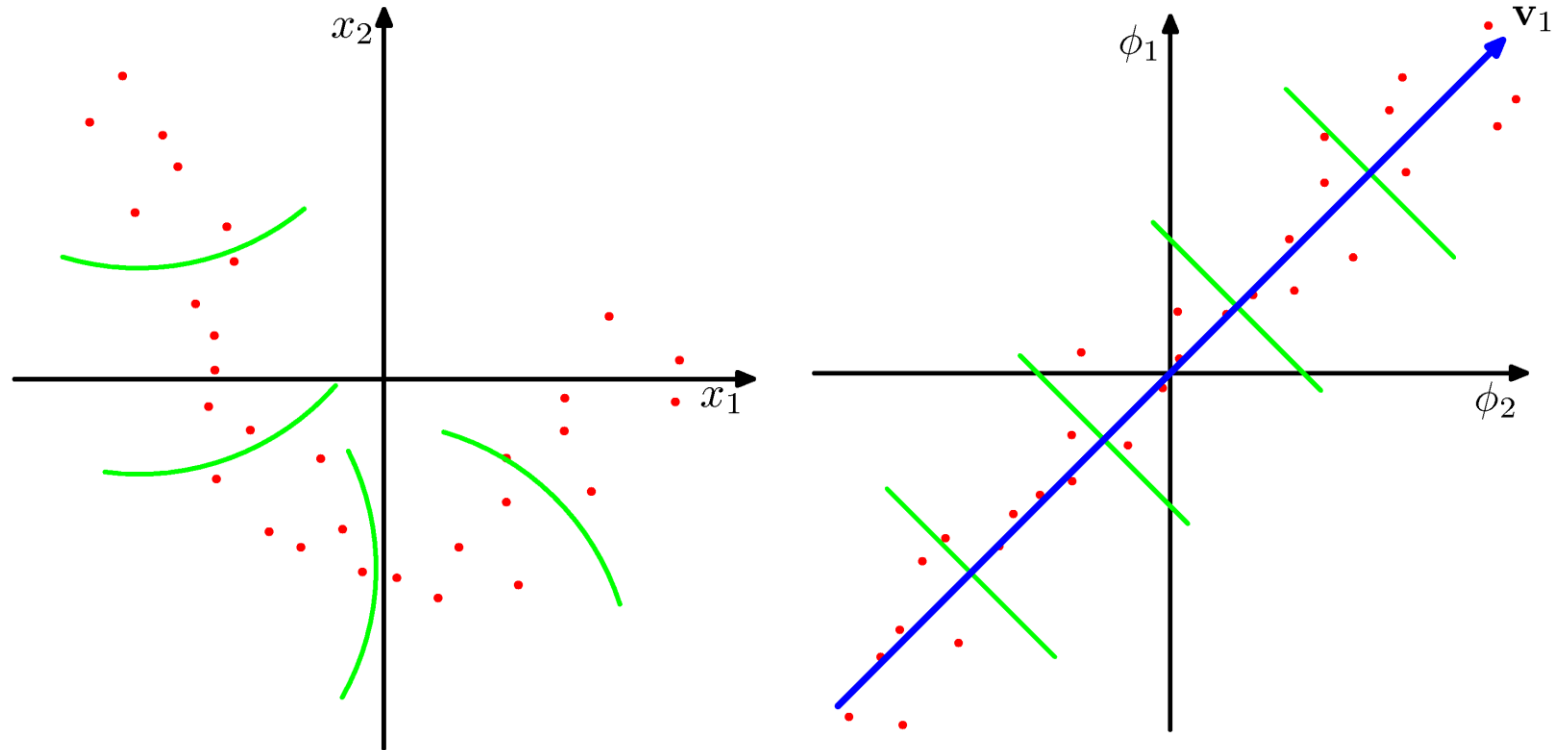
$$p(W|\alpha) = \prod_i \left(\frac{\alpha_i}{2\pi}\right)^{\frac{D}{2}} \exp\left(-\frac{1}{2} \alpha_i w_i^T w_i\right)$$

- Maximize the marginal likelihood (i.e., marginalize W)

$$p(X|\alpha, \mu, \sigma^2) = \int p(X|W, \mu, \sigma^2) p(W|\alpha) d\alpha$$


Kernel PCA

- Suppose the regularity that allows dimensionality reduction is non-linear.



Kernel PCA

- As with regression and classification, we can transform the raw input data $\{\mathbf{x}_n\}$ to a set of feature values

$$\{\mathbf{x}_n\} \longrightarrow \{\phi(\mathbf{x}_n)\}$$

- Linear PCA gives us a linear subspace in the feature value space, corresponding to nonlinear structure in the data space.

Kernel PCA

- Define a kernel, to avoid having to evaluate the feature vectors explicitly.

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$$

- Define the Gram matrix K of pairwise similarities among the data points:

$$K_{nm} = \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m)$$

- Express PCA in terms of the kernel,
 - Some care is required to centralize the data.

Next

- Next, from Bishop:
 - Non-linear dimensionality reduction
 - Hidden Markov Models, Dynamical Systems
- Then, Reinforcement Learning
 - From the Sutton & Barto book