

EECS 545: Machine Learning

Lecture 16. Unsupervised Learning: EM & PCA

Honglak Lee

3/13/2024



Outline

Unsupervised Learning for Clustering:

- Recap: Expectation Maximization
- Gaussian Mixtures with EM (derivation)

Unsupervised Learning for Finding Subspace:

- Principal Component Analysis (PCA)
- Probabilistic PCA
- Kernel PCA

Expectation Maximization

(Recap) Expectation Maximization

- Parameter learning for latent variable models:
when the data is not fully observed.
 - \mathbf{X} : observed variables, \mathbf{Z} : hidden (latent) variables
 - Main idea:
 - (E-step) Run inference about \mathbf{Z} given \mathbf{X} : $q(\mathbf{Z}) = p(\mathbf{Z} \mid \mathbf{X}, \theta)$
 - (M-step) Update parameters by treating q as observation
(i.e., fractional pseudo-counts)!
 - Example:
 - K-means (a special case of Gaussian mixtures)
 - Gaussian mixtures
- $$\operatorname{argmax}_{\theta} \sum_{\mathbf{Z}} q(\mathbf{Z}) \log p(\mathbf{X}, \mathbf{Z} | \theta)$$

The EM Algorithm in a nutshell

- Variational lower bound on the data likelihood:

$$\begin{aligned}\log p(\mathbf{X} \mid \theta) &= \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \frac{p(\mathbf{X}, \mathbf{Z} \mid \theta)}{q(\mathbf{Z})} + \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \frac{q(\mathbf{Z})}{p(\mathbf{Z} \mid \mathbf{X}, \theta)} \\ &= \mathcal{L}(q, \theta) + KL(q(\mathbf{Z}) \parallel p(\mathbf{Z} \mid \mathbf{X}, \theta)) \\ &\geq \mathcal{L}(q, \theta) \quad \text{Evidence Lower bound (ELBO) or variational lower bound}\end{aligned}$$

with equality holding if and only if $q(\mathbf{Z}) = p(\mathbf{Z} \mid \mathbf{X})$

- **EM algorithm:**

* E: expectation
* M: maximization

Repeat alternating optimization until convergence:

- E-step: for fixed θ , find q that maximizes $\mathcal{L}(q, \theta)$
- M-step: for fixed q , find θ that maximizes $\mathcal{L}(q, \theta)$

Note: For M-step, we need to sum the lower bound for all the training samples. See later slides.

(Recap) The EM Algorithm: E-step

- Variational lower bound on the data likelihood:

$$\begin{aligned}\log p(\mathbf{X} \mid \theta) &= \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \frac{p(\mathbf{X}, \mathbf{Z} \mid \theta)}{q(\mathbf{Z})} + \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \frac{q(\mathbf{Z})}{p(\mathbf{Z} \mid \mathbf{X}, \theta)} \\ &= \mathcal{L}(q, \theta) + KL(q(\mathbf{Z}) \parallel p(\mathbf{Z} \mid \mathbf{X}, \theta)) \\ &\geq \mathcal{L}(q, \theta) \quad \text{Evidence Lower bound (ELBO) or variational lower bound}\end{aligned}$$

with equality holding if and only if $q(\mathbf{Z}) = p(\mathbf{Z} \mid \mathbf{X})$

- **(E-step)** For a fixed θ , which q maximizes $\mathcal{L}(q, \theta)$?
⇒ $p(\mathbf{Z} \mid \mathbf{X})$, because all other q would make $\mathcal{L}(q, \theta)$ strictly less than $\log p(\mathbf{X} \mid \theta)$

(Recap) The EM Algorithm: M-step

- We also note that for a fixed q , the $\mathcal{L}(q, \theta)$ term can be decomposed into two terms:

$$\begin{aligned}\mathcal{L}(q, \theta) &= \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \frac{p(\mathbf{X}, \mathbf{Z} \mid \theta)}{q(\mathbf{Z})} \\ &= \sum_{\mathbf{Z}} q(\mathbf{Z}) \log p(\mathbf{X}, \mathbf{Z} \mid \theta) - \sum_{\mathbf{Z}} q(\mathbf{Z}) \log q(\mathbf{Z})\end{aligned}$$

- (1) A weighted sum of $\log p(\mathbf{X}, \mathbf{Z} \mid \theta)$.
This is tractable and can be optimized w.r.t θ
- (2) Entropy of $q(\mathbf{Z})$ which is independent of θ since q is fixed.
- **(M-step)** Thus, when q is fixed, we can find θ that maximizes $\mathcal{L}(q, \theta)$.

The EM Algorithm: summary

- Initialize parameters θ randomly
- Repeat until convergence:
(optimize $\mathcal{L}(q, \theta)$ w.r.t. q and θ alternatingly.)
 - “E-step”: Set $q(\mathbf{Z}) = p(\mathbf{Z} \mid \mathbf{X}, \theta)$ compute posterior → optimal $q(\mathbf{Z})$!
 - “M-step”: Update θ via the following maximization

$$\operatorname{argmax}_{\theta} \mathcal{L}(q, \theta) = \operatorname{argmax}_{\theta} \sum_{\mathbf{Z}} q(\mathbf{Z}) \log p(\mathbf{X}, \mathbf{Z} | \theta)$$

use $q(\mathbf{Z})$ as (factional) pseudo-counts and maximize the “data completion” log-likelihood

- Note we have assumed that $p(\mathbf{Z} \mid \mathbf{X}, \theta)$ is tractable
(i.e., find exact posterior $p(\mathbf{Z} \mid \mathbf{X}, \theta)$). Q. What if it is not?

The EM Algorithm: Multiple data-points

- Variational lower bound for a single example \mathbf{x} :

$$\begin{aligned}\log p(\mathbf{x}|\theta) &= \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(\mathbf{z}, \mathbf{x}|\theta)}{q(\mathbf{z})} + KL(q(\mathbf{z}) \| p(\mathbf{z}|\mathbf{x}, \theta)) \\ &\geq \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(\mathbf{z}, \mathbf{x}|\theta)}{q(\mathbf{z})}\end{aligned}$$

- Lower bound on the log-likelihood of the *entire* training data $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$:

$$\begin{aligned}\log p(\mathcal{D}|\theta) &= \sum_n \log p(\mathbf{x}^{(n)}|\theta) = \sum_n \sum_{\mathbf{z}} q^{(n)}(\mathbf{z}) \log \frac{p(\mathbf{z}, \mathbf{x}^{(n)}|\theta)}{q^{(n)}(\mathbf{z})} + \sum_n KL(q^{(n)}(\mathbf{z}) \| p(\mathbf{z}|\mathbf{x}^{(n)}, \theta)) \\ &\geq \sum_n \sum_{\mathbf{z}} q^{(n)}(\mathbf{z}) \log \frac{p(\mathbf{z}, \mathbf{x}^{(n)}|\theta)}{q^{(n)}(\mathbf{z})}\end{aligned}$$

Note that different $q^{(n)}$ is used for each n

The EM Algorithm: Multiple data-points

$$\begin{aligned}\log p(\mathcal{D}|\theta) &= \sum_n \log p(\mathbf{x}^{(n)}|\theta) = \sum_n \sum_{\mathbf{z}} q^{(n)}(\mathbf{z}) \log \frac{p(\mathbf{z}, \mathbf{x}^{(n)}|\theta)}{q^{(n)}(\mathbf{z})} + \sum_n KL(q^{(n)}(\mathbf{z}) \| p(\mathbf{z}|\mathbf{x}^{(n)}, \theta)) \\ &\geq \sum_n \sum_{\mathbf{z}} q^{(n)}(\mathbf{z}) \log \frac{p(\mathbf{z}, \mathbf{x}^{(n)}|\theta)}{q^{(n)}(\mathbf{z})}\end{aligned}$$

- Initialize random parameters θ
- Repeat until convergence:
 - “**E-step**”: Set $q^{(n)}(\mathbf{z}) = p(\mathbf{z} | \mathbf{x}^{(n)}, \theta)$,
for each training sample n.
 - “**M-step**”: Update θ via the following maximization:

$$\arg \max_{\theta} \sum_n \sum_{\mathbf{z}} q^{(n)}(\mathbf{z}) \log p(\mathbf{z}, \mathbf{x}^{(n)} | \theta)$$

Mixtures of Gaussians (recap)

- Let \mathbf{z} in $\{0,1\}^K$ be a 1-of- K random variable;

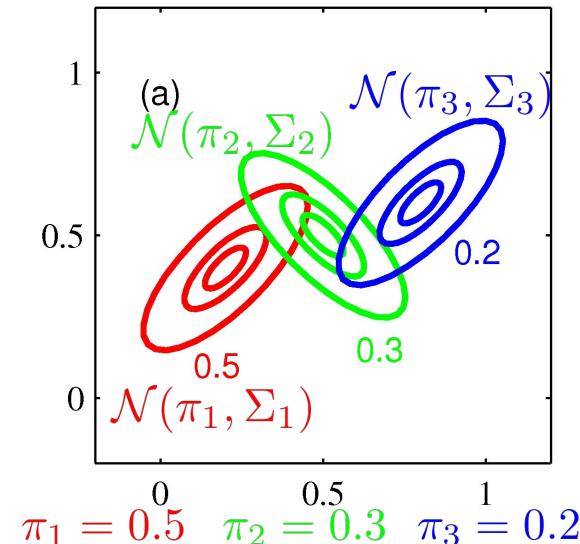
$$p(z_k = 1) = \pi_k \quad \sum_{k=1}^K \pi_k = 1$$

- Generate \mathbf{x} from Gaussian given the selected cluster assignment \mathbf{z}

$$p(\mathbf{x} \mid z_k = 1) = \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)$$

$$p(\mathbf{x}, \mathbf{z}) = \pi_k \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)$$

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z}) p(\mathbf{x} | \mathbf{z}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)$$



Summary: EM for Gaussian Mixtures

- Initialize parameters randomly $\theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$
- Repeat until convergence (alternating optimization)
 - E Step: Given fixed parameters θ , set $q^{(n)}(\mathbf{z}) = p(\mathbf{z} \mid \mathbf{x}^{(n)}, \theta)$

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)} | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}^{(n)} | \mu_j, \Sigma_j)} = P(z_k = 1 | \mathbf{x}^{(n)})$$

- M Step: Given fixed $q(\mathbf{z}^{(n)})$'s for $\mathbf{x}^{(n)}$'s (or $\gamma(z_{nk})$), update θ :

$$\pi_k^{\text{new}} = \frac{N_k}{N} = \frac{\sum_n \gamma(z_{nk})}{N} \quad \arg \max_{\theta} \sum_n \sum_{\mathbf{z}} q^{(n)}(\mathbf{z}) \log p(\mathbf{z}, \mathbf{x}^{(n)} \mid \theta)$$

$$\mu_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}^{(n)}$$

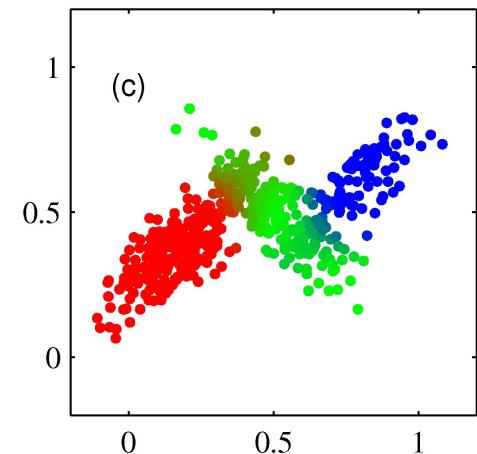
$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}^{(n)} - \mu_k^{\text{new}}) (\mathbf{x}^{(n)} - \mu_k^{\text{new}})^{\top}$$

Mixtures of Gaussians: E-step

- Need to calculate $p(\mathbf{z} \mid \mathbf{X}, \theta)$, i.e., *soft assignments*
- Responsibility is the degree (posterior prob.) to which each Gaussian explains an observation \mathbf{X} .

Q. Verify this! (Hint: Use Bayes Rule)

$$q^{(n)}(\mathbf{z}_k) = p(\mathbf{z}_k | \mathbf{x}^{(n)}) = \frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}^{(n)} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} =: \gamma(\mathbf{z}_{nk})$$



Mixtures of Gaussians: M-step

General formula for M-step:

$$\arg \max_{\theta} \sum_n \sum_{\mathbf{z}} q^{(n)}(\mathbf{z}) \log p(\mathbf{z}, \mathbf{x}^{(n)} \mid \theta)$$

Plug in for GMM: $\theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \mid k \in \{1 \dots K\}\}$

$$\begin{aligned} & \operatorname{argmax}_{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}} \underbrace{\sum_{n=1}^N \sum_{k=1}^K q^{(n)}(\mathbf{z}_k) \log p(\mathbf{z}_k, \mathbf{x}^{(n)} \mid \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}_{=J(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})} \\ & \text{s.t. } \sum_{k=1}^K \pi_k = 1 \end{aligned}$$

Mixtures of Gaussians: M-step

Let's first simplify the expression $J(\pi, \mu, \Sigma)$

$$\begin{aligned} J(\pi, \mu, \Sigma) &= \sum_{n=1}^N \sum_{k=1}^K q^{(n)}(\mathbf{z}_k) \log p(\mathbf{z}_k, \mathbf{x}^{(n)} \mid \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \left(\log \pi_k + \log \frac{1}{(2\pi)^{m/2} (\det \boldsymbol{\Sigma}_k)^{1/2}} - \frac{1}{2} (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) \right) \\ &= \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \pi_k - \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \left((2\pi)^{m/2} (\det \boldsymbol{\Sigma}_k)^{1/2} \right) \\ &\quad - \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) \\ &= \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \pi_k - \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \det \boldsymbol{\Sigma}_k \\ &\quad - \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) + \text{const} \end{aligned}$$

Mixtures of Gaussians: M-step

$$J(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \pi_k - \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \det \boldsymbol{\Sigma}_k \\ - \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) + \text{const}$$

- Maximize J w.r.t. $\boldsymbol{\mu}_k$ by differentiating w.r.t. $\boldsymbol{\mu}_k$ and setting the gradient to 0:

$$\frac{\partial J}{\partial \boldsymbol{\mu}_k} = \sum_{n=1}^N \gamma(\mathbf{z}_{nk}) \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) = 0$$

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^N \gamma(\mathbf{z}_{nk}) \mathbf{x}^{(n)}}{\sum_{n=1}^N \gamma(\mathbf{z}_{nk})}$$

Mixtures of Gaussians: M-step

$$\begin{aligned} J(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = & \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \pi_k - \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \det \boldsymbol{\Sigma}_k \\ & - \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) + \text{const} \end{aligned}$$

- To find $\boldsymbol{\Sigma}_k$, we use change of variables: $\mathbf{M}_k = \boldsymbol{\Sigma}_k^{-1}$

$$\begin{aligned} J(\boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{M}) = & \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \pi_k + \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \det \mathbf{M}_k \\ & - \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top \mathbf{M}_k (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) + \text{const} \end{aligned}$$

Mixtures of Gaussians: M-step

$$J(\boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{M}) = \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \pi_k + \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \det \mathbf{M}_k \\ - \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top \mathbf{M}_k (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) + \text{const}$$

- Maximize J w.r.t. \mathbf{M}_k by differentiating w.r.t. \mathbf{M}_k and setting the gradient to 0:

*Note: $\frac{\partial \log |\det \mathbf{X}|}{\partial \mathbf{X}} = (\mathbf{X}^{-1})^\top \quad \frac{\partial \mathbf{a}^\top \mathbf{X} \mathbf{b}}{\partial \mathbf{X}} = \mathbf{a} \mathbf{b}^\top$

$$\frac{\partial J}{\partial \mathbf{M}_k} = \frac{1}{2} \sum_{n=1}^N \gamma(\mathbf{z}_{nk}) \mathbf{M}_k^{-1} - \frac{1}{2} \sum_{n=1}^N \gamma(\mathbf{z}_{nk}) (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top = 0$$

$$\boldsymbol{\Sigma}_k = \mathbf{M}_k^{-1} = \frac{\sum_{n=1}^N \gamma(\mathbf{z}_{nk}) (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top}{\sum_{n=1}^N \gamma(\mathbf{z}_{nk})}$$

Mixtures of Gaussians: M-step

$$J(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \pi_k - \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \det \boldsymbol{\Sigma}_k \\ - \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) + \text{const}$$

- Finally we need: $\max_{\boldsymbol{\pi}} \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \pi_k \quad \text{s.t.} \quad \sum_{k=1}^K \pi_k = 1$
- Use Lagrange multipliers

$$L(\boldsymbol{\pi}, \alpha) = \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \pi_k - \alpha \left(\sum_{k=1}^K \pi_k - 1 \right)$$

Mixtures of Gaussians: M-step

- Finally we need: $\max_{\boldsymbol{\pi}} \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \pi_k \quad \text{s.t. } \sum_{k=1}^K \pi_k = 1$
- Use Lagrange multipliers

$$L(\boldsymbol{\pi}, \alpha) = \sum_{n=1}^N \sum_{k=1}^K \gamma(\mathbf{z}_{nk}) \log \pi_k - \alpha \left(\sum_{k=1}^K \pi_k - 1 \right)$$

- Setting $\frac{\partial L}{\partial \pi_k} = \sum_{n=1}^N \gamma(\mathbf{z}_{nk}) \frac{1}{\pi_k} - \alpha = 0$ gives $\pi_k = \frac{\sum_{n=1}^N \gamma(\mathbf{z}_{nk})}{\alpha}$
- Using the constraint we get:

$$\pi_k = \frac{\sum_{n=1}^N \gamma(\mathbf{z}_{nk})}{\sum_{k=1}^K \sum_{n=1}^N \gamma(\mathbf{z}_{nk})} = \frac{\sum_{n=1}^N \gamma(\mathbf{z}_{nk})}{N}$$

Mixtures of Gaussians: M-step (putting together)

- The mean of a cluster is the weighted mean, weighted by the responsibilities.

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^N \gamma(\mathbf{z}_{nk}) \mathbf{x}^{(n)}}{N_k}$$

[N_k = sum of pseudo-counts γ_{nk} over n.]

- N_k is the effective number of points in cluster k

$$N_k = \sum_{n=1}^N \gamma(\mathbf{z}_{nk}) \quad \pi_k = \frac{N_k}{N}$$

- Likewise for covariance:

$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^N \gamma(\mathbf{z}_{nk})(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top}{N_k}$$

EM for Gaussian Mixtures: Summary

- Initialize means, covariances, and mixing coefficients for the K Gaussians.
- E Step: Given the parameters, evaluate the responsibilities or the posterior

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)} | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}^{(n)} | \mu_j, \Sigma_j)} = P(z_k = 1 | \mathbf{x}^{(n)})$$

EM for Gaussian Mixtures: Summary

- M Step: Given the responsibilities, re-evaluate the coefficients.

$$\pi_k^{\text{new}} = \frac{N_k}{N} = \frac{\sum_n \gamma(z_{nk})}{N}$$

$$\mu_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}^{(n)}$$

$$\sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}^{(n)} - \mu_k^{\text{new}}) (\mathbf{x}^{(n)} - \mu_k^{\text{new}})^{\top}$$

- Stop when either coefficients or log likelihood converges.

Summary: EM for Gaussian Mixtures

- Initialize parameters randomly $\theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$
- Repeat until convergence (alternating optimization)
 - E Step: Given fixed parameters θ , set $q^{(n)}(\mathbf{z}) = p(\mathbf{z} \mid \mathbf{x}^{(n)}, \theta)$

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)} | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}^{(n)} | \mu_j, \Sigma_j)} = P(z_k = 1 | \mathbf{x}^{(n)})$$

- M Step: Given fixed $q(\mathbf{z}^{(n)})$'s for $\mathbf{x}^{(n)}$'s (or $\gamma(z_{nk})$), update θ :

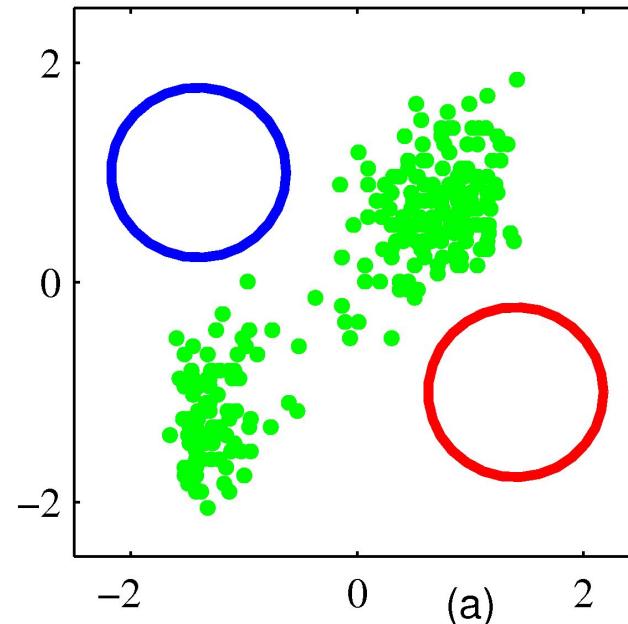
$$\pi_k^{\text{new}} = \frac{N_k}{N} = \frac{\sum_n \gamma(z_{nk})}{N} \quad \arg \max_{\theta} \sum_n \sum_{\mathbf{z}} q^{(n)}(\mathbf{z}) \log p(\mathbf{z}, \mathbf{x}^{(n)} \mid \theta)$$

$$\mu_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}^{(n)}$$

$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}^{(n)} - \mu_k^{\text{new}}) (\mathbf{x}^{(n)} - \mu_k^{\text{new}})^{\top}$$

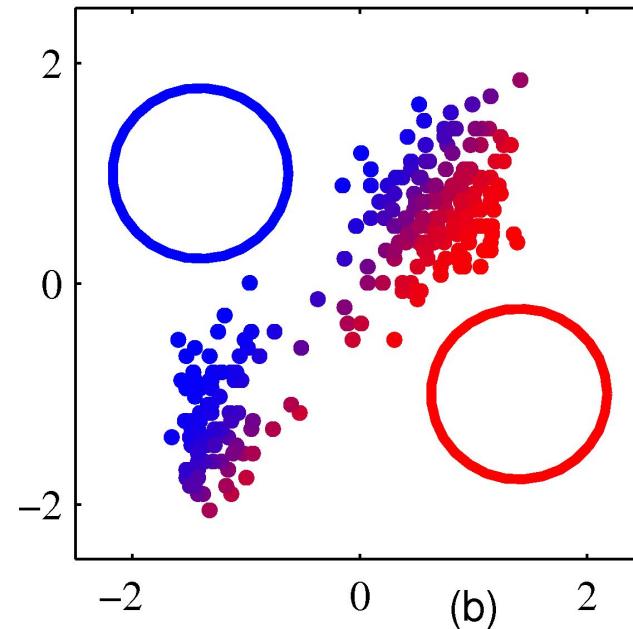
EM Example

- Initialize parameters: means, covariances, and mixing coefficients.



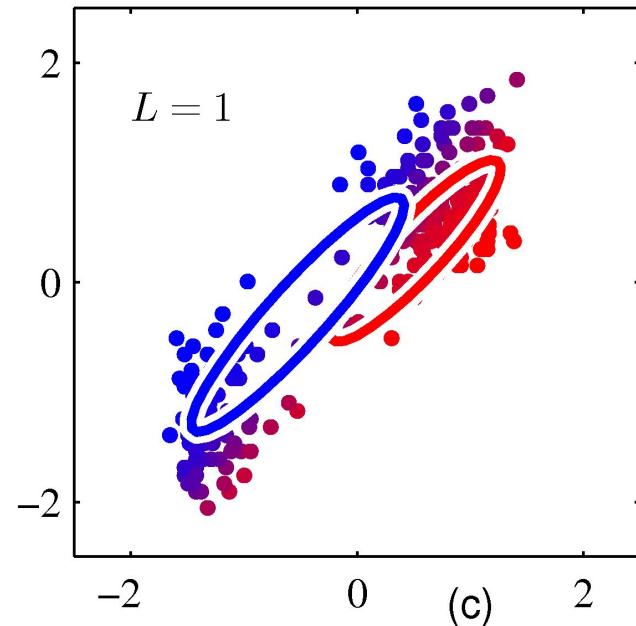
EM Example

- First E Step



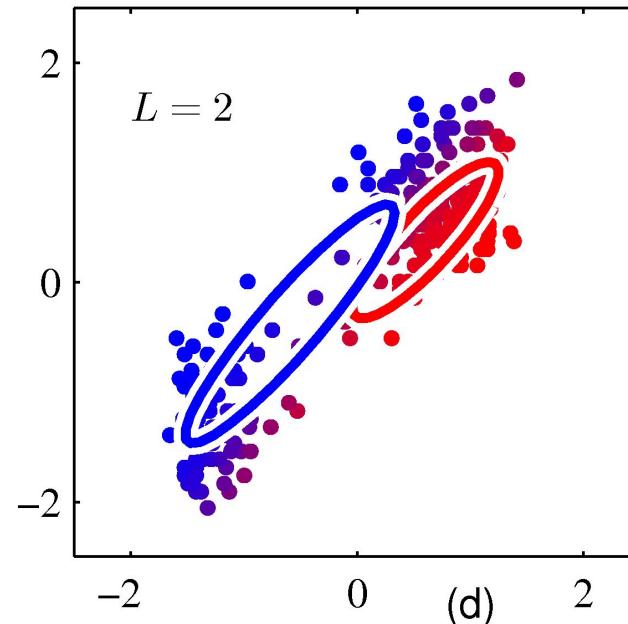
EM Example

- First M Step



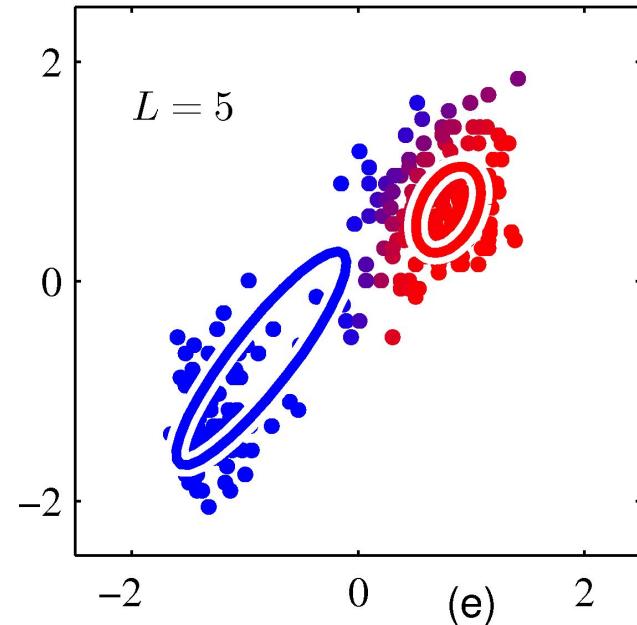
EM Example

- Second E and M Steps



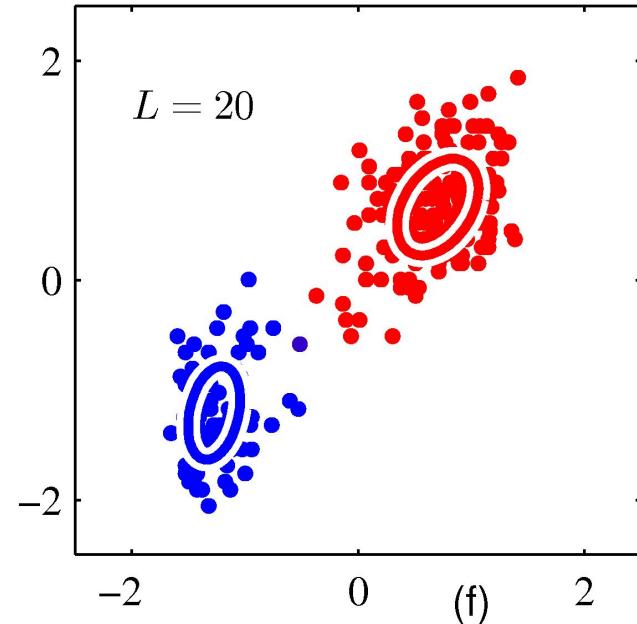
EM Example

- Three more E-M cycles



EM Example

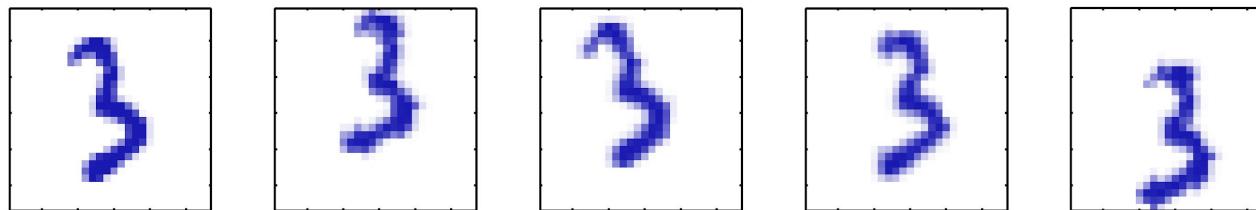
- Fifteen E-M cycles later



Principal Component Analysis (PCA)

High-Dimensional Data

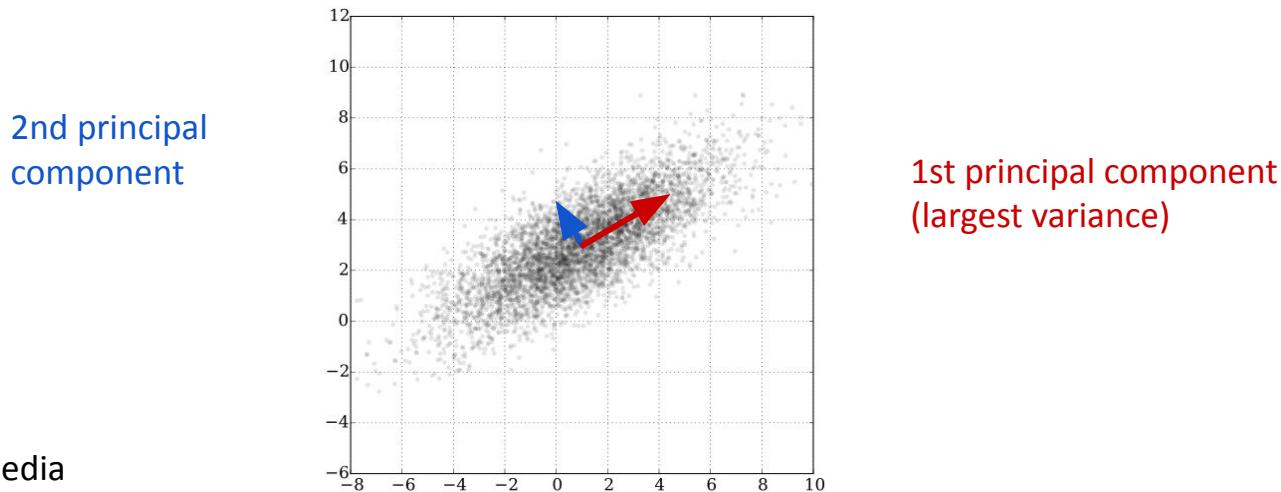
- ... may have low-dimensional structure.



- 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 of $\{\mathbf{x}^{(n)}\}_{n=1,\dots,N}$ of observations
 - in a space of dimension D , i.e., $\mathbf{x}^{(n)} \in \mathbb{R}^D$
 - find a subspace of dimension $M < D$
 - that captures most of its *variability*. (i.e., approximate $\mathbf{x}^{(n)}$'s using principal components as basis vectors)



Principal Component Analysis

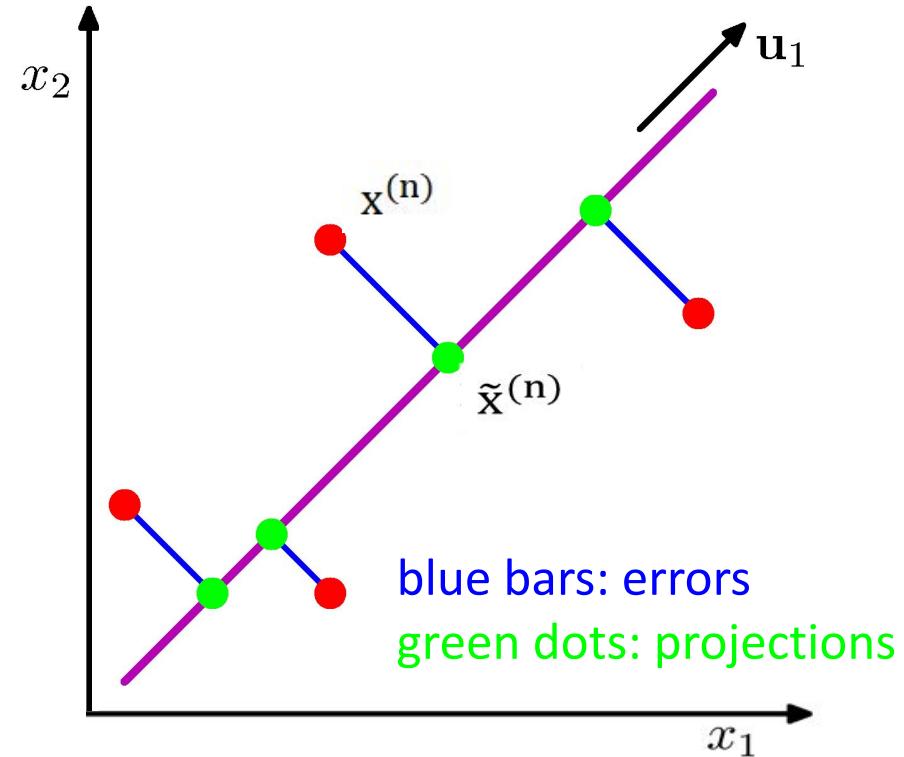
- Given a set of $\{\mathbf{x}^{(n)}\}_{n=1,\dots,N}$ of observations
 - in a space of dimension D ,
 - find a subspace of dimension $M < D$
 - that captures most of its *variability*. (i.e., approximate $\mathbf{x}^{(n)}$'s using principal components as basis vectors)
- PCA can be described as either:
 - maximizing the variance of the projection, or
 - minimizing the squared approximation error.
 - (both are equivalent; see the next slide)

Two Descriptions of PCA

Approximate the data
with *projection*

(i.e., for each $x^{(n)}$, find closest point on
on the subspace spanned by principal
components):

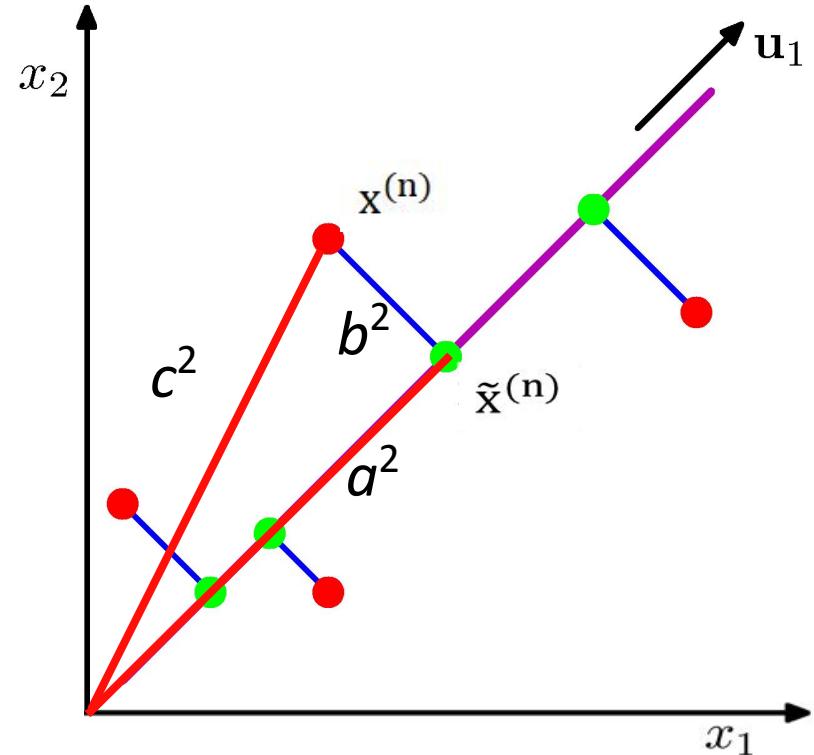
- Maximize variance, or
- Minimize squared error



Main idea: We want to find a basis vector (e.g. u_1) (= principal component)
that does the best approximation or best preserves the variance when projected

Equivalent Descriptions

- With mean at the origin $c_i^2 = a_i^2 + b_i^2$
- With constant $\sum_i c_i^2$
 - Minimizing $\sum_i b_i^2$
 - Maximizes $\sum_i a_i^2$
 - ... and vice versa



Note: without loss of generality, here we assume that the input data x has zero mean.

First Principal Component

- Given data points $\{\mathbf{x}^{(n)}\}$ in a D-dim space,
 - Mean $\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}^{(n)}$
 - Data covariance $\mathbf{S} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}^{(n)} - \bar{\mathbf{x}})(\mathbf{x}^{(n)} - \bar{\mathbf{x}})^\top$
D x D matrix
- Let \mathbf{u}_1 be the PC maximizing variance of projection:
 - It should have length 1: $\mathbf{u}_1^\top \mathbf{u}_1 = 1$
 - Projection of $\mathbf{x}^{(n)}$ to \mathbf{u}_1 subspace: $(\mathbf{u}_1^\top \mathbf{x}^{(n)}) \mathbf{u}_1$
 - Remark: More generally, projection of $\mathbf{x}^{(n)}$ to subspace spanned by $\mathbf{u}_1, \dots, \mathbf{u}_M$:
$$\sum_{j=1}^M (\mathbf{u}_j^\top \mathbf{x}^{(n)}) \mathbf{u}_j$$

First Principal Component

- Maximize the projection variance:

$$\frac{1}{N} \sum_{n=1}^N (\mathbf{u}_1^\top \mathbf{x}^{(n)} - \mathbf{u}_1^\top \bar{\mathbf{x}})^2 = \mathbf{u}_1^\top \mathbf{S} \mathbf{u}_1$$

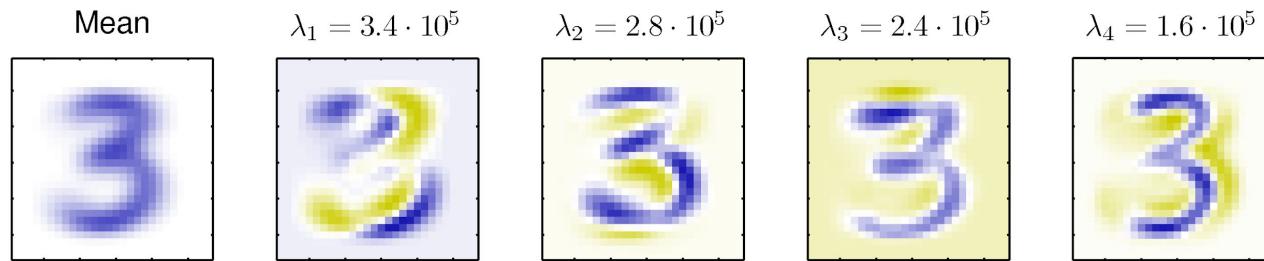
- Use a Lagrange multiplier to enforce $\mathbf{u}_1^\top \mathbf{u}_1 = 1$
- Maximize: $\mathbf{u}_1^\top \mathbf{S} \mathbf{u}_1 + \lambda_1(1 - \mathbf{u}_1^\top \mathbf{u}_1)$
- Derivative is zero when $\mathbf{S} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1$
 - That is, $\mathbf{u}_1^\top \mathbf{S} \mathbf{u}_1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

PCA by Maximizing Variance

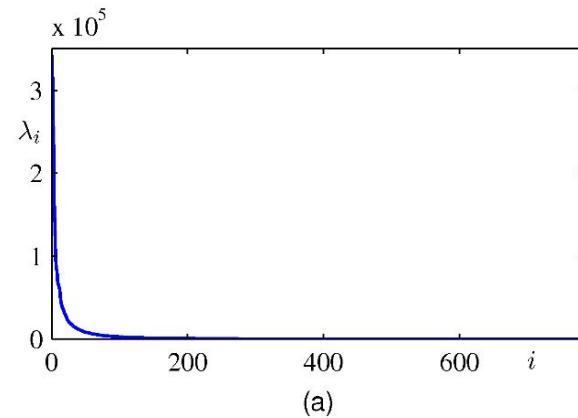
- Repeat to find the M eigenvectors of the data covariance matrix \mathbf{S} corresponding to the M largest eigenvalues.
 - The *total variance* is the sum of variances of all individual principal components
 - Principal components are orthogonal to each other
- We can also do the same thing from a “minimizing (projection) squared error” viewpoint.

Digit Image Example

- The mean and first four PCA eigenvectors.

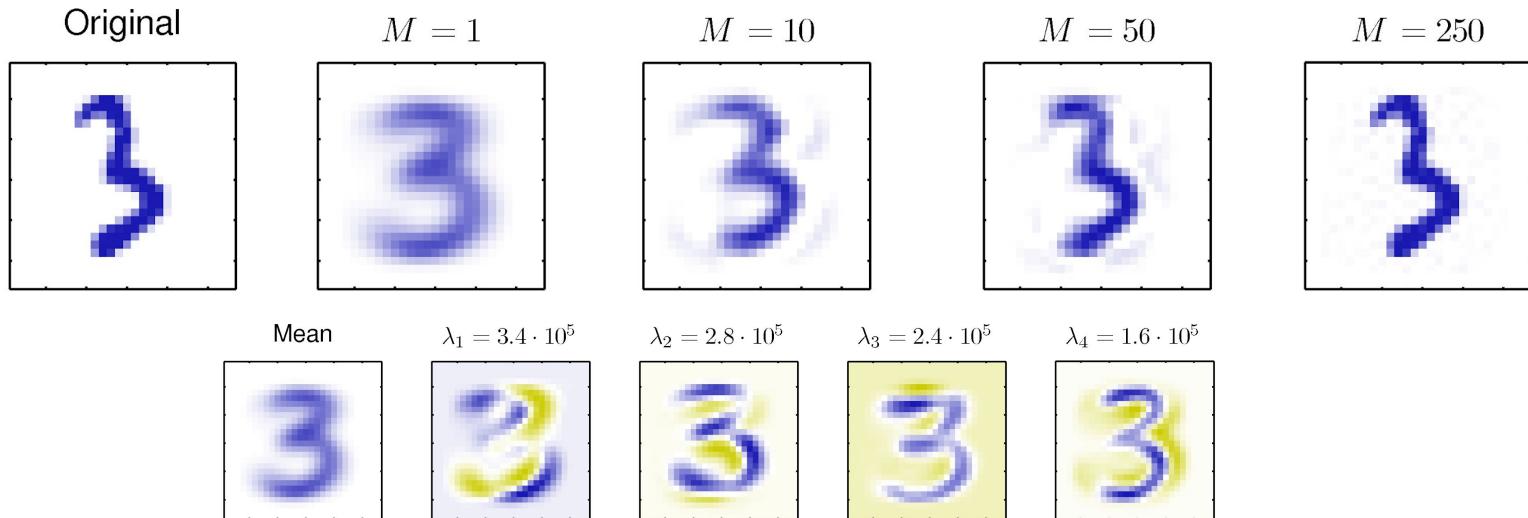


- The eigenvalue spectrum:



Reconstructing the Image

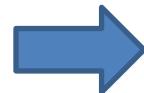
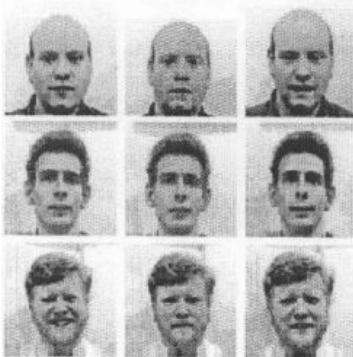
- Compress the image representation by using only first M eigenvectors, and discarding the less important information.



Learning features via PCA

- Example: Eigenfaces

Training face images



Learned PCA bases



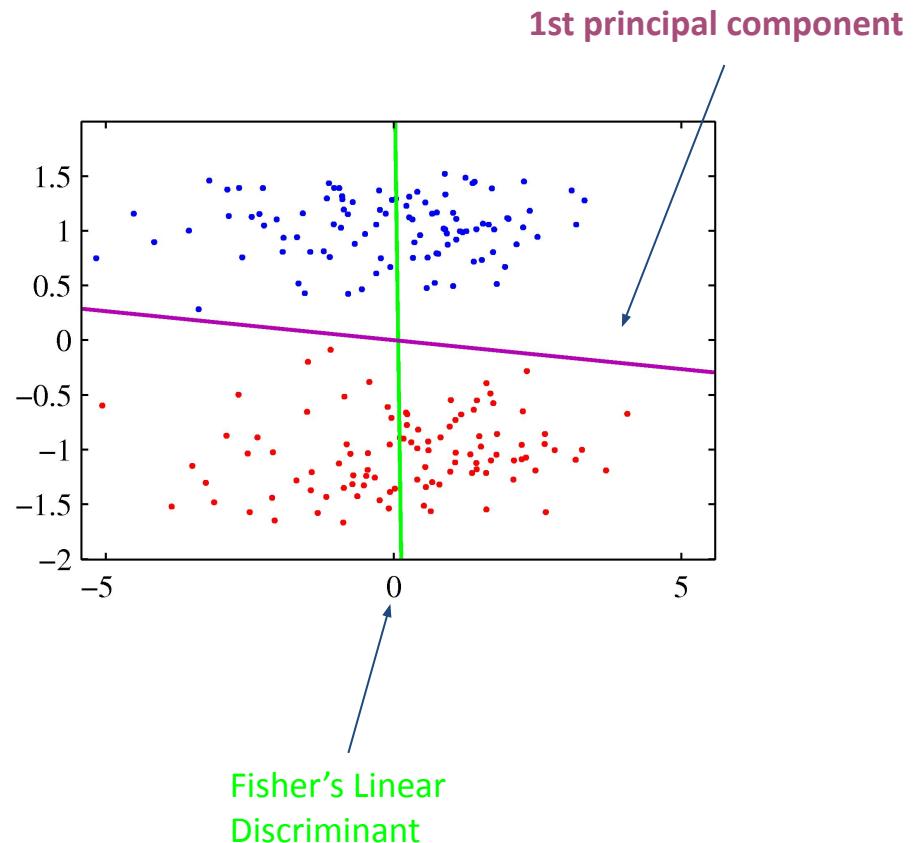
Test example

$$\text{Test Face} = 0.9571 * \text{Face 1} - 0.1945 * \text{Face 2} + 0.0461 * \text{Face 3} + 0.0586 * \text{Face 4}$$

The equation shows a test face being reconstructed as a weighted sum of the learned PCA bases. The weights are 0.9571, -0.1945, 0.0461, and 0.0586 respectively. Each term is multiplied by one of the four learned PCA basis images.

Limits to PCA

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



Probabilistic PCA

- 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.

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon} \quad p(\mathbf{z}) = \mathcal{N}(\mathbf{z} \mid \mathbf{0}, \mathbf{I}) \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$$

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

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

Probabilistic PCA

- Given the generative model

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

- we can infer

$$\mathbb{E}[\mathbf{x}] = \mathbb{E}[\mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}] = \boldsymbol{\mu}$$

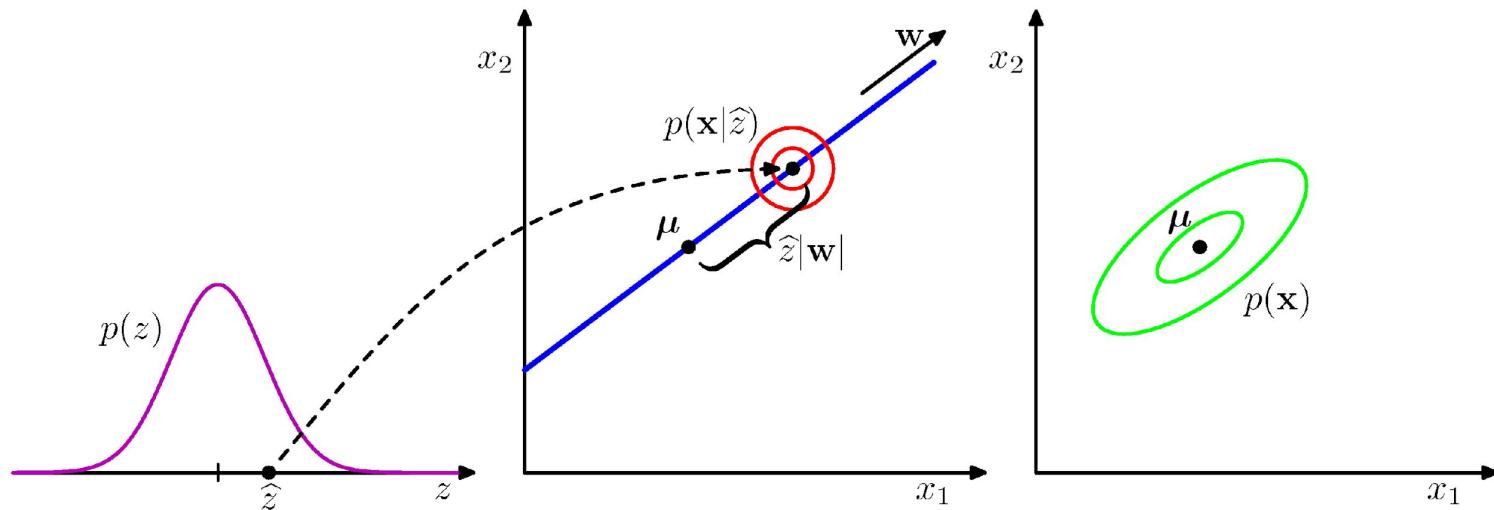
$$\begin{aligned}\text{cov}[\mathbf{x}] &= \mathbb{E}[(\mathbf{W}\mathbf{z} + \boldsymbol{\epsilon})(\mathbf{W}\mathbf{z} + \boldsymbol{\epsilon})^\top] \\ &= \mathbb{E}[\mathbf{W}\mathbf{z}\mathbf{z}^\top\mathbf{W}^\top] + \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top] = \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}\end{aligned}$$

Probabilistic PCA

- The generative model

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

can be illustrated as:



Likelihood of Probabilistic PCA

- (Marginal) likelihood:

$$\begin{aligned} & \log p(\mathbf{x} \mid \mathbf{W}, \boldsymbol{\mu}, \sigma^2) \\ &= \sum_i p(\mathbf{x}^{(i)} \mid \mathbf{W}, \boldsymbol{\mu}, \sigma^2) \\ &= -\frac{ND}{2} \log 2\pi - \frac{N}{2} \log |C| - \frac{1}{2} \sum_i (\mathbf{x}^{(i)} - \boldsymbol{\mu})^\top C^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}) \end{aligned}$$

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

- We can simply maximize this likelihood function with respect to $\mathbf{W}, \boldsymbol{\mu}, \sigma^2$.

Maximum Likelihood Parameters

- Mean: $\mu = \bar{\mathbf{x}}$
- Noise: $\sigma_{\text{ML}}^2 = \frac{1}{D - M} \sum_{i=M+1}^D \lambda_i$
- \mathbf{W} : $\mathbf{W}_{\text{ML}} = \mathbf{U}_M (\mathbf{L}_M - \sigma^2 \mathbf{I})^{1/2} \mathbf{R}$

where

- \mathbf{L}_M is diag with the M largest eigenvalues
- \mathbf{U}_M is the M corresponding eigenvectors
- \mathbf{R} is an arbitrary M by M orthogonal matrix (rotation matrix) (i.e., \mathbf{z} can be defined by rotating “back”)

Maximum likelihood by EM

- Latent variable model

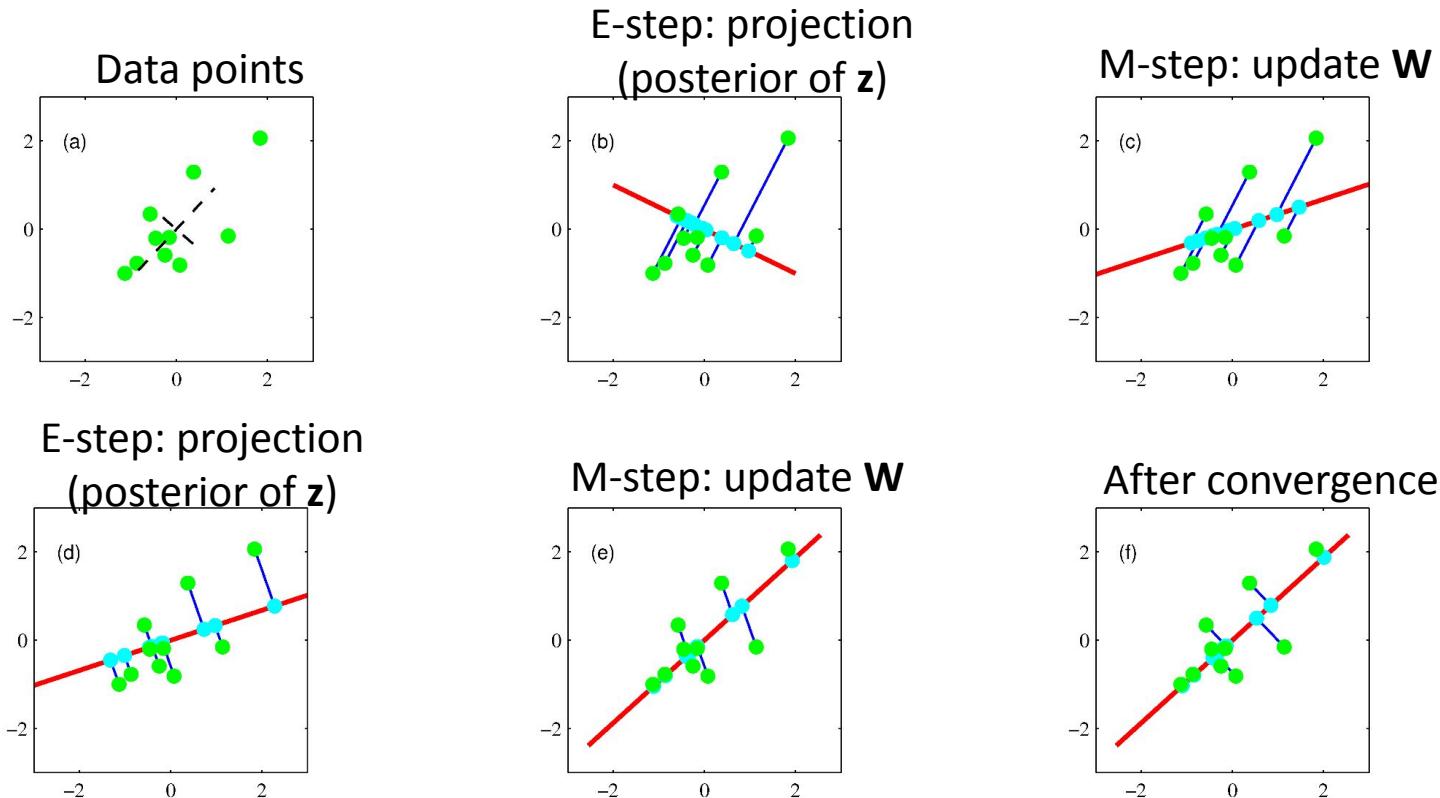
$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z} \mid \mathbf{0}, \mathbf{I})$$

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

- E-step: Estimate the posterior $q(\mathbf{z}) = p(\mathbf{z} \mid \mathbf{x})$
 - Use linear Gaussian
- M-step: Maximize the data-completion likelihood given $q(\mathbf{z})$

$$\text{maximize}_{\theta=\{\mathbf{W}, \boldsymbol{\mu}, \sigma\}} \sum_i \sum_{\mathbf{z}^{(i)}} q(\mathbf{z}^{(i)}) \log p_\theta(\mathbf{x}^{(i)}, \mathbf{z}^{(i)})$$

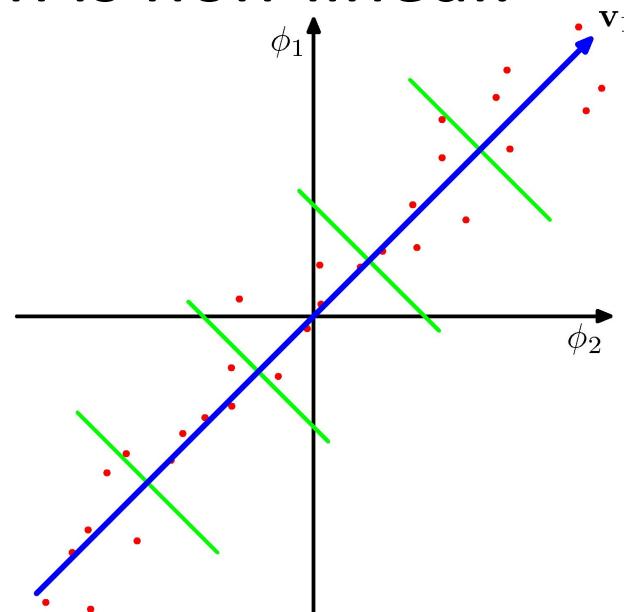
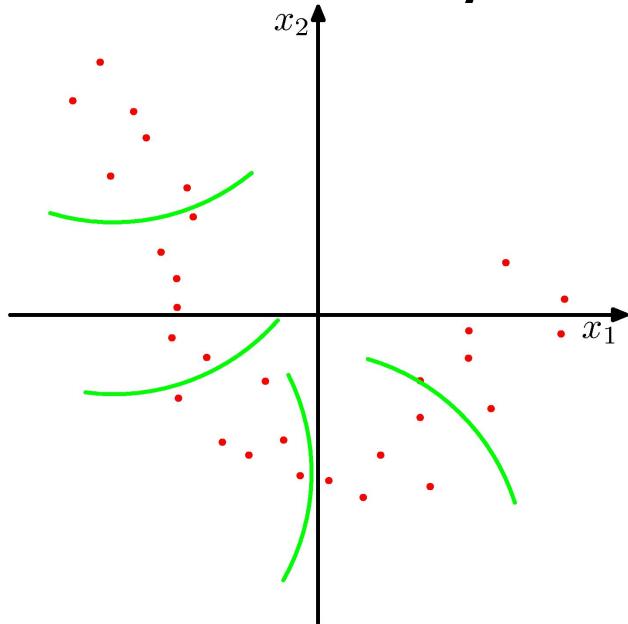
Finding PCA params by EM



- Illustrating EM on simulated data

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

$$\{ \mathbf{x}^{(n)} \} \rightarrow \{ \phi(\mathbf{x}^{(n)}) \}$$

- Linear PCA (on the nonlinear feature space) 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.

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}')$$

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

$$K_{nm} = \phi(\mathbf{x}^{(n)})^\top \phi(\mathbf{x}^{(m)}) = k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)})$$

Kernel PCA

- Assume that $\{\phi(\mathbf{x}^{(n)})\}$ have zero mean.
- Sample covariance matrix: $S = \frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}^{(n)})\phi(\mathbf{x}^{(n)})^\top = \frac{1}{N} \Phi^\top \Phi$
- Let \mathbf{v} be an eigenvector for S

$$S\mathbf{v} = \lambda\mathbf{v} \implies \lambda\mathbf{v} = \Phi^\top \left(\frac{1}{N} \Phi \mathbf{v} \right)$$

$$\therefore \mathbf{v} = \Phi^\top \boldsymbol{\alpha} \text{ for some } \boldsymbol{\alpha} \in \mathbb{R}^N$$

- Thus, $S\mathbf{v} = \lambda\mathbf{v} \implies \lambda\Phi^\top \boldsymbol{\alpha} = \frac{1}{N} \Phi^\top \Phi \Phi^\top \boldsymbol{\alpha} = \frac{1}{N} \Phi^\top K \boldsymbol{\alpha}$
- Multiply Φ on both sides and cancel out $K = \Phi \Phi^\top$

$$\lambda N \boldsymbol{\alpha} = K \boldsymbol{\alpha} \implies \boldsymbol{\alpha} \text{ is an eigenvector of } K$$

Kernel PCA

- We thus have $\mathbf{v} = \Phi^\top \boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is eigenvector of the kernel matrix K .
- Now, $\|\mathbf{v}\| = 1 \implies \boldsymbol{\alpha}^\top K \boldsymbol{\alpha} = \boldsymbol{\alpha}^\top \lambda_K \boldsymbol{\alpha} = 1 \implies \|\boldsymbol{\alpha}\| = \lambda_K^{-1/2}$
- It is often infeasible to obtain \mathbf{v} (depends on dim of Φ), but we can compute projections:

$$\mathbf{v}^\top \phi(\mathbf{x}) = \boldsymbol{\alpha}^\top \Phi \phi(\mathbf{x}) = \boldsymbol{\alpha}^\top k(\mathbf{x}) \quad \text{where} \quad k(\mathbf{x}) = [k(\mathbf{x}^{(1)}, \mathbf{x}), \dots, k(\mathbf{x}^{(N)}, \mathbf{x})]$$

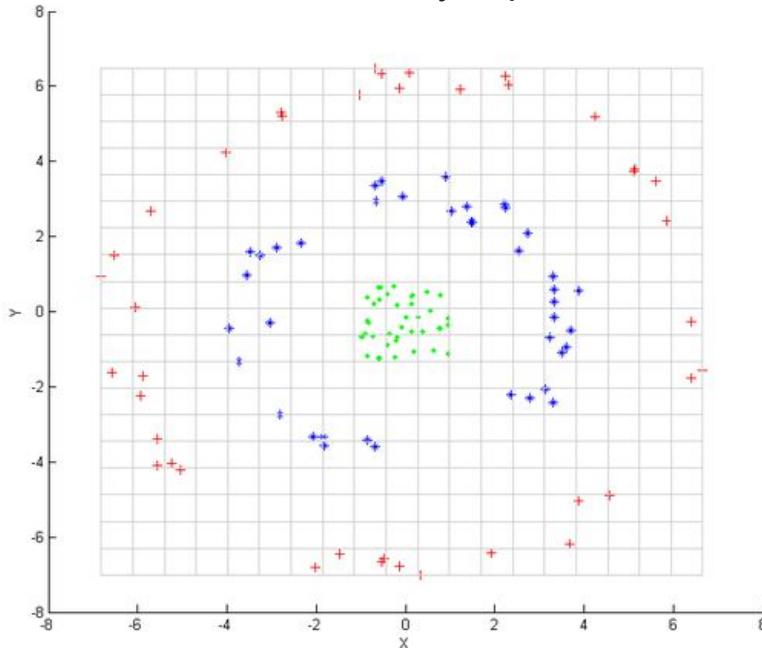
- Finally, some care is required to centralize data (to ensure that features have zero mean):

$$K' = K - \mathbf{1}_N K - K \mathbf{1}_N + \mathbf{1}_N K \mathbf{1}_N$$

where $\mathbf{1}_N \in \mathbb{R}^{N \times N}$ is a matrix of ones.

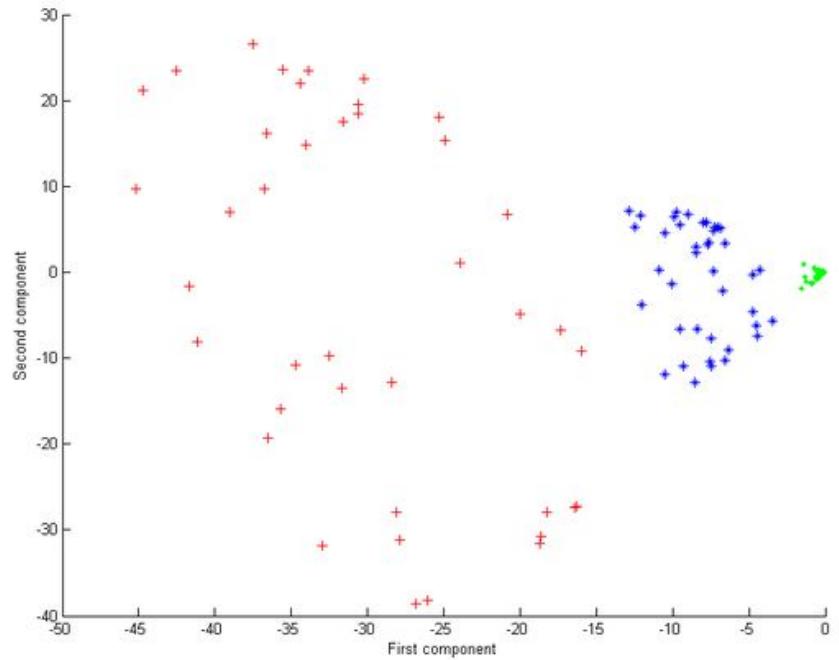
Kernel PCA

Linear PCA operates only in the given (in this case two-dimensional) space, in which these concentric point clouds are not linearly separable.



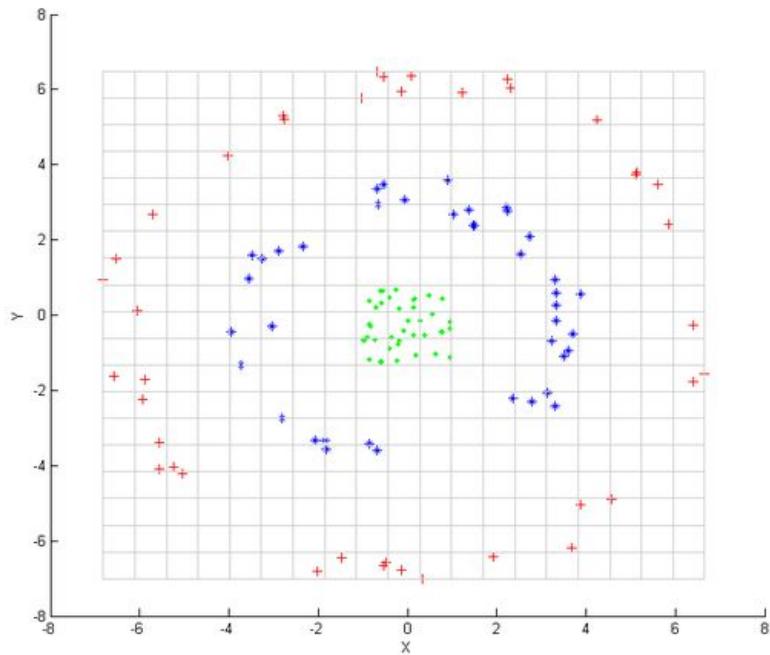
Data

The first principal component is enough to distinguish the three different groups

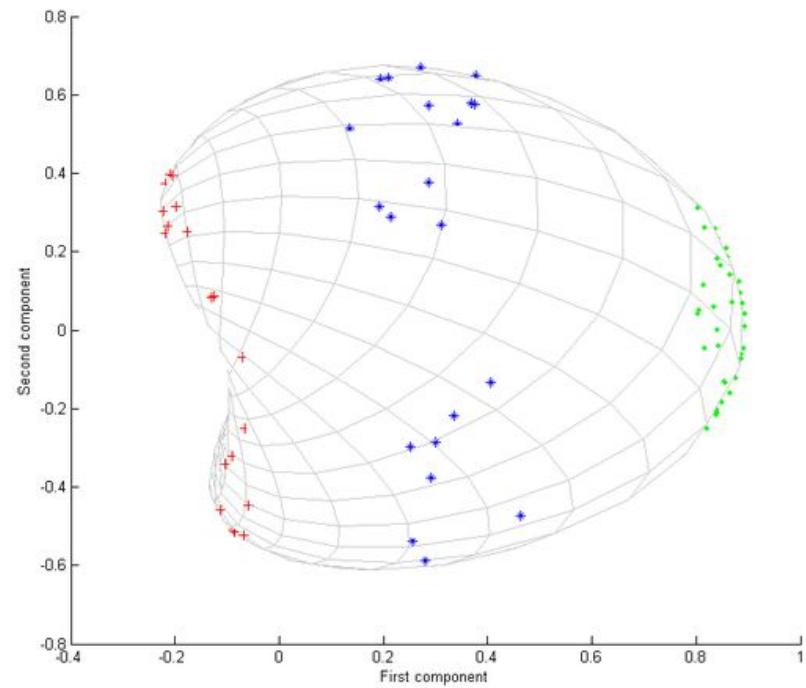


Kernel PCA with
 $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + 1)^2$

Kernel PCA



Data



Kernel PCA with
Gaussian kernel

Any feedback (about lecture, slide, homework, project, etc.)?

(via anonymous google form: <https://forms.gle/99jeftYTaozJvCEF8>)



Change Log of lecture slides:

https://docs.google.com/document/d/e/2PACX-1vRKx40eOJKACqrKWraio0AmIFS1_xBMINuWcc-jzpfo-ySj_gBuqTVdfHy8v4HDmqDJ3b3TvAW1FVuH/pub