Dimensionality Reduction

Jia-Bin Huang
Virginia Tech

ECE-5424G / CS-5824

Spring 2019

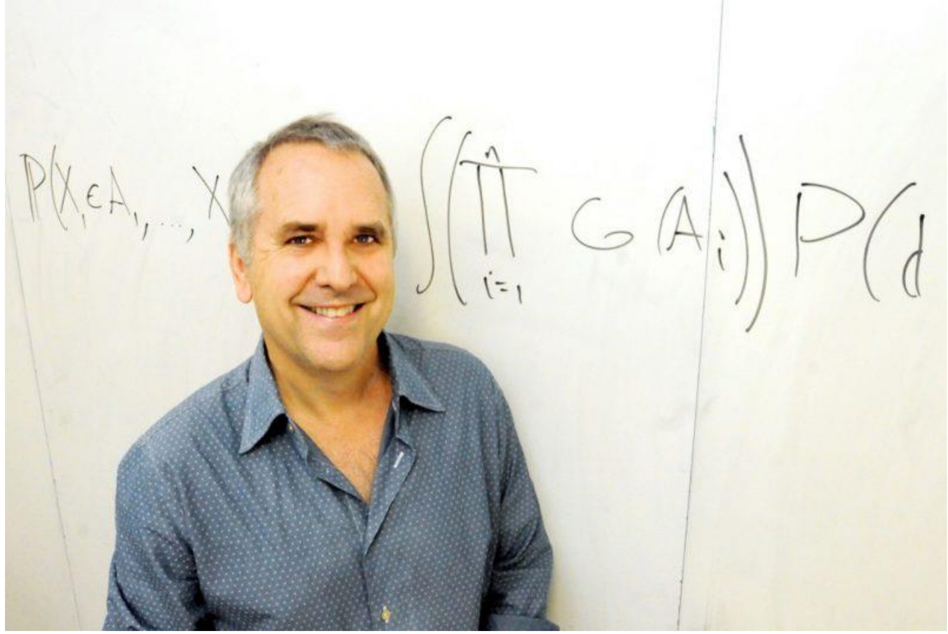
Administrative

- HW 3 due March 27.

- HW 4 out tonight

J. Mark Sowers Distinguished Lecture

- **Michael Jordan**
- Pehong Chen Distinguished Professor
  Department of Statistics and Electrical Engineering and Computer Sciences
- University of California, Berkeley
- **3/28/19**
- 7:30 PM, McBryde 100

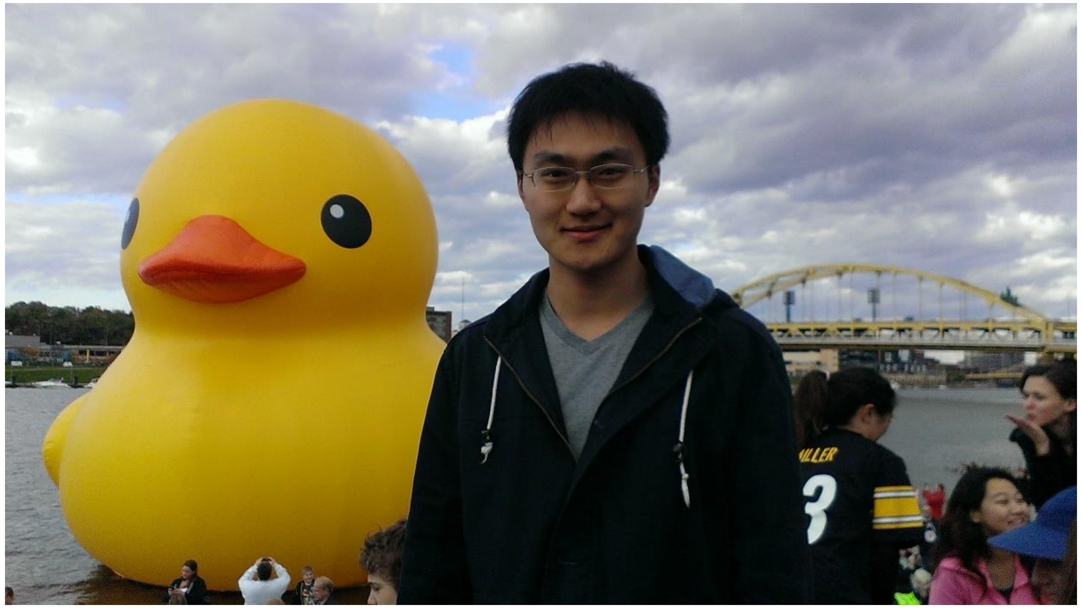


ECE Faculty Candidate Talk

- **Siheng Chen**
- Ph.D. Carnegie Mellon University

- Data science with graphs: From social network analysis to autonomous driving

- Time: 10:00 AM - 11:00 AM March 28
- Location: 457B Whittemore



Expectation Maximization (EM) Algorithm

- Goal: Find θ that maximizes log-likelihood $\sum_i \log p(x^{(i)};\theta)$

$$\sum_i \log p(x^{(i)};\theta) = \sum_i \log \sum_{z^{(i)}} p(x^{(i)}, z^{(i)};\theta)$$

$$= \sum_i \boxed{\log} \boxed{\sum_{z^{(i)}} Q_i(z^{(i)}) \boxed{\frac{p(x^{(i)}, z^{(i)};\theta)}{Q_i(z^{(i)})}}}$$

$$\geq \sum_i \boxed{\sum_{z^{(i)}} Q_i(z^{(i)}) \boxed{\log} \boxed{\frac{p(x^{(i)}, z^{(i)};\theta)}{Q_i(z^{(i)})}}}$$

Jensen's inequality: $f(E[X]) \geq E[f(X)]$

Expectation Maximization (EM) Algorithm

- Goal: Find θ that maximizes log-likelihood $\sum_i \log p(x^{(i)};\theta)$

$$\sum_i \log p(x^{(i)};\theta) \geq \sum_i \sum_{z^{(i)}} Q_i\left(z^{(i)}\right) \log \boxed{\frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})}}$$

- The lower bound works for all possible set of distributions Q_i
- We want **tight** lower-bound: $f(\,E[X]\,) = E[f(\,X\,)]$
- When will that happen? $X = E[X]$ with probability 1 (X is a constant)

$$\frac{p\left(x^{(i)},z^{(i)};\theta\right)}{Q_i(z^{(i)})} = c$$

How should we choose $Q_i(z^{(i)})$?

- $\frac{p\left(x^{(i)},z^{(i)};\theta\right)}{Q_i(z^{(i)})} = c$
- $Q_i(z^{(i)}) \propto p\left(x^{(i)}, z^{(i)}; \theta\right)$
- $\sum_z Q_i(z^{(i)}) = 1$ (because it is a distribution)

- $Q_i\left(z^{(i)}\right) = \frac{p\left(x^{(i)},z^{(i)};\theta\right)}{\sum_z p\left(x^{(i)},z^{(i)};\theta\right)} = \frac{p\left(x^{(i)},z^{(i)};\theta\right)}{p\left(x^{(i)};\theta\right)}$

$$= p(z^{(i)}|x^{(i)}; \theta)$$

EM algorithm

Repeat until convergence{

(E-step) For each i, set

$$Q_i\left(z^{(i)}\right) := p(z^{(i)}|x^{(i)};\theta) \qquad \text{(Probabilistic inference)}$$

(M-step) Set

$$\theta := \operatorname{argmax}_\theta \sum_i \sum_{z^{(i)}} Q_i\left(z^{(i)}\right) \log \frac{p\left(x^{(i)}, z^{(i)}; \theta\right)}{Q_i(z^{(i)})}$$

}

Expectation Maximization (EM) Algorithm

Goal: $\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \log\left(\sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z} \mid \theta)\right)$

Log of sums is intractable

Jensen's Inequality $\quad f\left(\mathrm{E}[X]\right) \geq \mathrm{E}\left[f(X)\right]$ for concave functions f(x)

(so we maximize the lower bound!)

Maximum Likelihood from Incomplete Data Via the ***EM* Algorithm**
AP Dempster, NM Laird… - Journal of the Royal …, 1977 - Wiley Online Library
A broadly applicable **algorithm** for computing maximum likelihood estimates from incomplete data is presented at various levels of generality. Theory showing the monotone behaviour of the likelihood and convergence of the **algorithm** is derived. Many examples are sketched …
Cited by 54643 Related articles All 61 versions Web of Science: 23929 Import into BibTeX

See here for proof: www.stanford.edu/class/cs229/notes/cs229-notes8.ps

Expectation Maximization (EM) Algorithm

Goal: $\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \log\left(\sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z} \mid \theta)\right)$

1. E-step: compute

$$\mathrm{E}_{z|x,\theta^{(t)}}\left[\log\left(p(\mathbf{x}, \mathbf{z} \mid \theta)\right)\right] = \sum_{\mathbf{z}} \log\left(p(\mathbf{x}, \mathbf{z} \mid \theta)\right) p\left(\mathbf{z} \mid \mathbf{x}, \theta^{(t)}\right)$$

2. M-step: solve

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} \sum_{\mathbf{z}} \log\left(p(\mathbf{x}, \mathbf{z} \mid \theta)\right) p\left(\mathbf{z} \mid \mathbf{x}, \theta^{(t)}\right)$$

log of expectation of P(x|z)

Goal: $\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \log\left(\sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z} \mid \theta)\right)$ $\qquad f(\mathrm{E}[X]) \geq \mathrm{E}[f(X)]$

1. E-step: compute

expectation of log of P(x|z)

$$\mathrm{E}_{z|x,\theta^{(t)}}\left[\log(p(\mathbf{x}, \mathbf{z} \mid \theta))\right] = \sum_{\mathbf{z}} \log(p(\mathbf{x}, \mathbf{z} \mid \theta))\, p\left(\mathbf{z} \mid \mathbf{x}, \theta^{(t)}\right)$$

2. M-step: solve

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} \sum_{\mathbf{z}} \log(p(\mathbf{x}, \mathbf{z} \mid \theta))\, p\left(\mathbf{z} \mid \mathbf{x}, \theta^{(t)}\right)$$

EM for Mixture of Gaussians - derivation

$$p\left(x_n \mid \boldsymbol{\mu},\boldsymbol{\sigma}^2,\boldsymbol{\pi}\right)=\sum_m p\left(x_n, z_n = m \mid \mu_m, {\sigma_m}^2, \pi_m\right) = \sum_m \frac{1}{\sqrt{2\pi{\sigma_m}^2}}\exp\left(-\frac{\left(x_n-\mu_m\right)^2}{{\sigma_m}^2}\right)\cdot\pi_m$$

1. E-step: $\mathrm{E}_{z|x,\theta^{(t)}}\left[\log\left(p\left(\mathbf{x},\mathbf{z}\mid\theta\right)\right)\right]=\sum_{\mathbf{z}}\log\left(p\left(\mathbf{x},\mathbf{z}\mid\theta\right)\right)p\left(\mathbf{z}\mid\mathbf{x},\theta^{(t)}\right)$
2. M-step: $\theta^{(t+1)}=\underset{\theta}{\operatorname{argmax}}\sum_{\mathbf{z}}\log\left(p\left(\mathbf{x},\mathbf{z}\mid\theta\right)\right)p\left(\mathbf{z}\mid\mathbf{x},\theta^{(t)}\right)$

EM for Mixture of Gaussians

$$p\left(x_n \mid \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\pi}\right) = \sum_m p\left(x_n, z_n = m \mid \mu_m, \sigma_m{}^2, \pi_m\right) = \sum_m \frac{1}{\sqrt{2\pi\sigma_m{}^2}} \exp\left(-\frac{\left(x_n - \mu_m\right)^2}{\sigma_m{}^2}\right) \cdot \pi_m$$

1. E-step: $\mathrm{E}_{z|x,\theta^{(t)}}\left[\log\left(p\left(\mathbf{x}, \mathbf{z} \mid \theta\right)\right)\right] = \sum_{\mathbf{z}} \log\left(p\left(\mathbf{x}, \mathbf{z} \mid \theta\right)\right) p\left(\mathbf{z} \mid \mathbf{x}, \theta^{(t)}\right)$
2. M-step: $\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} \sum_{\mathbf{z}} \log\left(p\left(\mathbf{x}, \mathbf{z} \mid \theta\right)\right) p\left(\mathbf{z} \mid \mathbf{x}, \theta^{(t)}\right)$

$$\alpha_{nm} = p(z_n = m \mid x_n, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2^{(t)}}, \boldsymbol{\pi}^{(t)})$$

$$\hat{\mu}_m{}^{(t+1)} = \frac{1}{\sum_n \alpha_{nm}} \sum_n \alpha_{nm} x_n \qquad \hat{\sigma}_m{}^{2(t+1)} = \frac{1}{\sum_n \alpha_{nm}} \sum_n \alpha_{nm} \left(x_n - \hat{\mu}_m\right)^2 \qquad \hat{\pi}_m{}^{(t+1)} = \frac{\sum_n \alpha_{nm}}{N}$$

EM algorithm - derivation

$$p(\mathbf{x}|\Theta) = \sum_{i=1}^{M} \alpha_i p_i(\mathbf{x}|\theta_i)$$

$$\log(\mathcal{L}(\Theta|\mathcal{X})) = \log \prod_{i=1}^{N} p(x_i|\Theta) = \sum_{i=1}^{N} \log \left(\sum_{j=1}^{M} \alpha_j p_j(x_i|\theta_j) \right)$$

$$\log(\mathcal{L}(\Theta|\mathcal{X}, \mathcal{Y})) = \log(P(\mathcal{X}, \mathcal{Y}|\Theta)) = \sum_{i=1}^{N} \log\left(P(x_i|y_i)P(y)\right) = \sum_{i=1}^{N} \log\left(\alpha_{y_i} p_{y_i}(x_i|\theta_{y_i})\right)$$

$$p(y_i|x_i, \Theta^g) = \frac{\alpha_{y_i}^g p_{y_i}(x_i|\theta_{y_i}^g)}{p(x_i|\Theta^g)} = \frac{\alpha_{y_i}^g p_{y_i}(x_i|\theta_{y_i}^g)}{\sum_{k=1}^{M} \alpha_k^g p_k(x_i|\theta_k^g)}$$

$$p(\mathbf{y}|\mathcal{X}, \Theta^g) = \prod_{i=1}^{N} p(y_i|x_i, \Theta^g)$$

http://lasa.epfl.ch/teaching/lectures/ML_Phd/Notes/GP-GMM.pdf

EM algorithm – E-Step

$$
\begin{aligned}
Q(\Theta, \Theta^g) &= \sum_{\mathbf{y} \in \Upsilon} \log\left(\mathcal{L}(\Theta|\mathcal{X}, \mathbf{y})\right) p(\mathbf{y}|\mathcal{X}, \Theta^g) \\
&= \sum_{\mathbf{y} \in \Upsilon} \sum_{i=1}^{N} \log\left(\alpha_{y_i} p_{y_i}(x_i|\theta_{y_i})\right) \prod_{j=1}^{N} p(y_j|x_j, \Theta^g) \\
&= \sum_{y_1=1}^{M} \sum_{y_2=1}^{M} \cdots \sum_{y_N=1}^{M} \sum_{i=1}^{N} \log\left(\alpha_{y_i} p_{y_i}(x_i|\theta_{y_i})\right) \prod_{j=1}^{N} p(y_j|x_j, \Theta^g) \\
&= \sum_{y_1=1}^{M} \sum_{y_2=1}^{M} \cdots \sum_{y_N=1}^{M} \sum_{i=1}^{N} \sum_{\ell=1}^{M} \delta_{\ell, y_i} \log\left(\alpha_\ell p_\ell(x_i|\theta_\ell)\right) \prod_{j=1}^{N} p(y_j|x_j, \Theta^g) \\
&= \sum_{\ell=1}^{M} \sum_{i=1}^{N} \log\left(\alpha_\ell p_\ell(x_i|\theta_\ell)\right) \underbrace{\sum_{y_1=1}^{M} \sum_{y_2=1}^{M} \cdots \sum_{y_N=1}^{M} \delta_{\ell, y_i} \prod_{j=1}^{N} p(y_j|x_j, \Theta^g)}_{p(\ell|x_i, \Theta^g)}
\end{aligned}
$$

EM algorithm – E-Step

$$
\begin{aligned}
Q(\Theta, \Theta^g) &= \sum_{\ell=1}^{M} \sum_{i=1}^{N} \log\left(\alpha_\ell p_\ell(x_i|\theta_\ell)\right) p(\ell|x_i, \Theta^g) \\
&= \sum_{\ell=1}^{M} \sum_{i=1}^{N} \log(\alpha_\ell) p(\ell|x_i, \Theta^g) + \sum_{\ell=1}^{M} \sum_{i=1}^{N} \log(p_\ell(x_i|\theta_\ell)) p(\ell|x_i, \Theta^g)
\end{aligned}
$$

EM algorithm – M-Step

$$\frac{\partial}{\partial \alpha_\ell}\left[\sum_{\ell=1}^{M}\sum_{i=1}^{N}\log(\alpha_\ell)p(\ell|x_i,\Theta^g)+\lambda\left(\sum_\ell \alpha_\ell - 1\right)\right]=0$$

$$\sum_{i=1}^{N}\frac{1}{\alpha_\ell}p(\ell|x_i,\Theta^g)+\lambda=0$$

$$\alpha_\ell=\frac{1}{N}\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)$$

EM algorithm – M-Step

$$\sum_{\ell=1}^{M} \sum_{i=1}^{N} \log \left(p_\ell(x_i | \mu_\ell, \Sigma_\ell) \right) p(\ell | x_i, \Theta^g)$$

$$= \sum_{\ell=1}^{M} \sum_{i=1}^{N} \left(-\frac{1}{2} \log(|\Sigma_\ell|) - \frac{1}{2} (x_i - \mu_\ell)^T \Sigma_\ell^{-1} (x_i - \mu_\ell) \right) p(\ell | x_i, \Theta^g)$$

Take derivative with respect to μ_l

$$\sum_{i=1}^{N} \Sigma_\ell^{-1} (x_i - \mu_\ell) p(\ell | x_i, \Theta^g) = 0$$

$$\mu_\ell = \frac{\sum_{i=1}^{N} x_i p(\ell | x_i, \Theta^g)}{\sum_{i=1}^{N} p(\ell | x_i, \Theta^g)}$$

EM algorithm – M-Step

Take derivative with respect to $\sum_{l}^{-1}$

$$\Sigma_\ell = \frac{\sum_{i=1}^{N} p(\ell|x_i,\Theta^g) N_{\ell,i}}{\sum_{i=1}^{N} p(\ell|x_i,\Theta^g)} = \frac{\sum_{i=1}^{N} p(\ell|x_i,\Theta^g)(x_i-\mu_\ell)(x_i-\mu_\ell)^T}{\sum_{i=1}^{N} p(\ell|x_i,\Theta^g)}$$

EM Algorithm for GMM

$$\alpha_\ell^{new} = \frac{1}{N}\sum_{i=1}^{N} p(\ell|x_i, \Theta^g)$$

$$\mu_\ell^{new} = \frac{\sum_{i=1}^{N} x_i p(\ell|x_i, \Theta^g)}{\sum_{i=1}^{N} p(\ell|x_i, \Theta^g)}$$

$$\Sigma_\ell^{new} = \frac{\sum_{i=1}^{N} p(\ell|x_i, \Theta^g)(x_i - \mu_\ell^{new})(x_i - \mu_\ell^{new})^T}{\sum_{i=1}^{N} p(\ell|x_i, \Theta^g)}$$

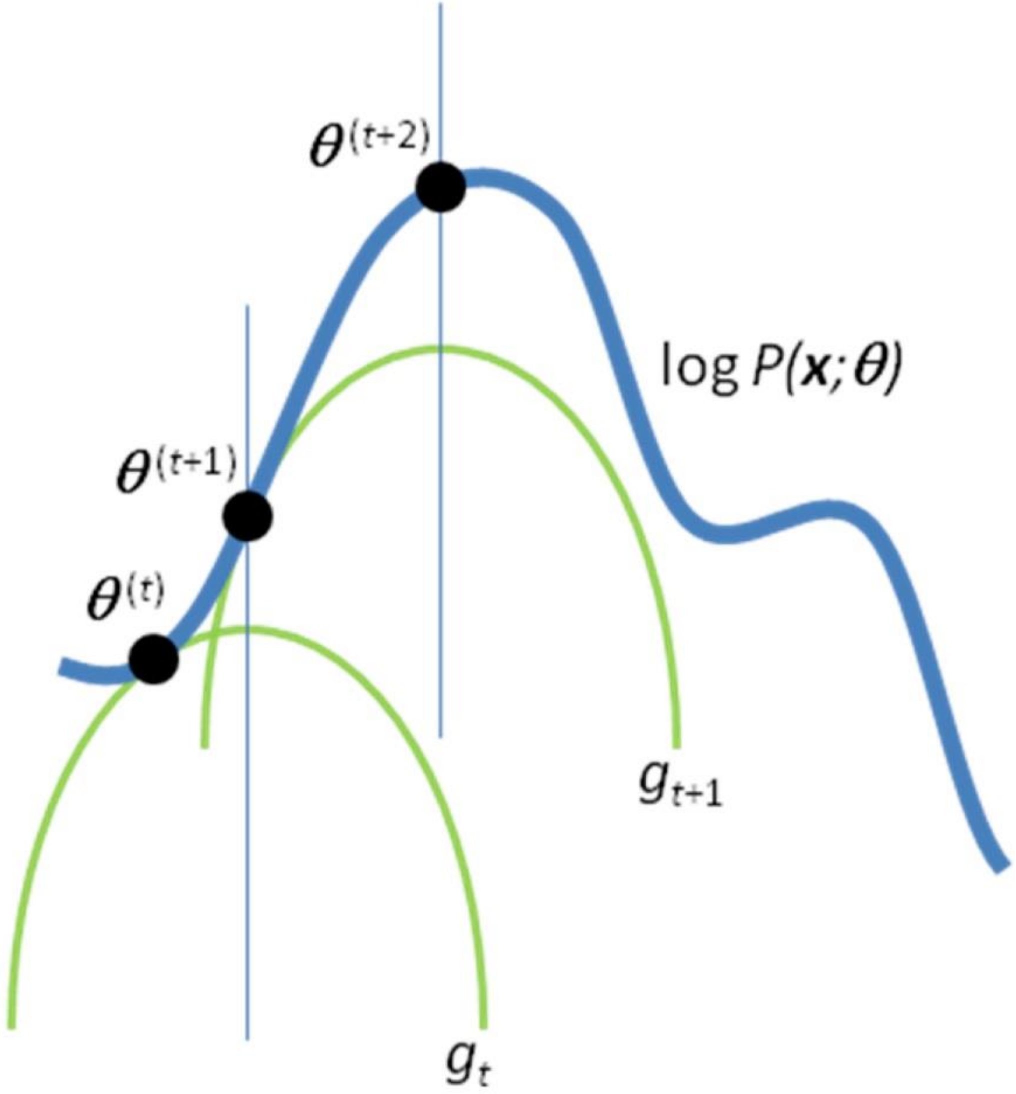
EM Algorithm

- Maximizes a lower bound on the data likelihood at each iteration

- Each step increases the data likelihood
  - Converges to *local maximum*

- Common tricks to derivation
  - Find terms that sum or integrate to 1
  - Lagrange multiplier to deal with constraints

Convergence of EM Algorithm



"Hard EM"

- Same as EM except compute z* as most likely values for hidden variables
- K-means is an example
- Advantages
  - Simpler: can be applied when cannot derive EM
  - Sometimes works better if you want to make hard predictions at the end
- But
  - Generally, pdf parameters are not as accurate as EM

Dimensionality Reduction

- Motivation
  - Data compression
  - Data visualization
- Principal component analysis
  - Formulation
  - Algorithm
  - Reconstruction
- Choosing the number of principal components
- Applying PCA

Dimensionality Reduction

- **Motivation**

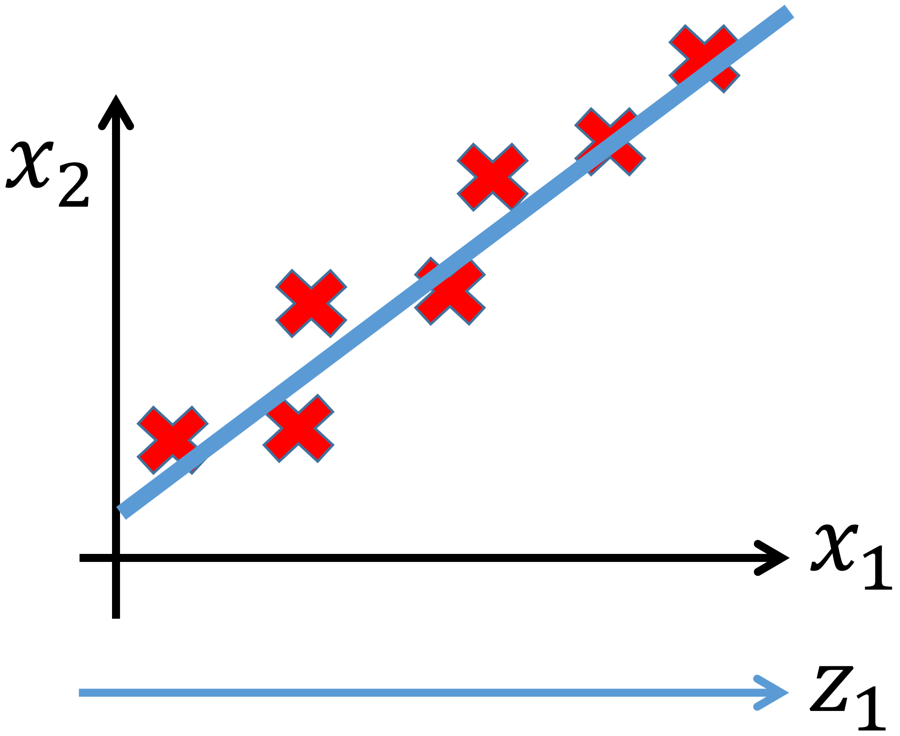
- Principal component analysis
  - Formulation
  - Algorithm
  - Reconstruction

- Choosing the number of principal components

- Applying PCA

Data Compression

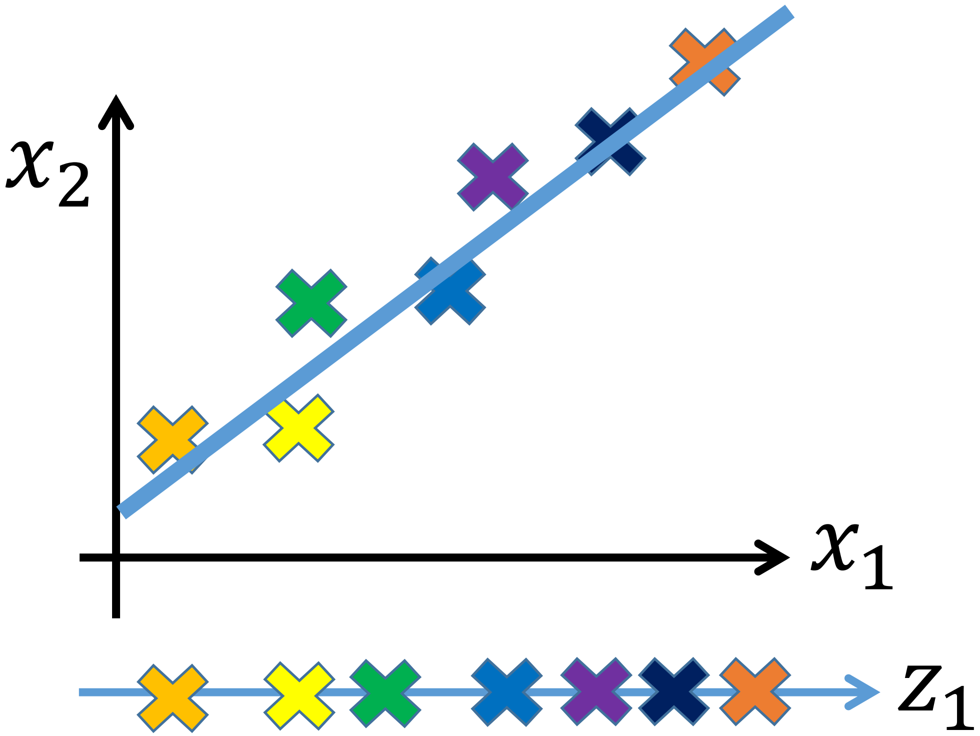
- Reduces the required time and storage space
- Removing multi-collinearity improves the interpretation of the parameters of the machine learning model.



$$x^{(1)} \in R^2 \rightarrow z^{(1)} \in R$$
$$x^{(2)} \in R^2 \rightarrow z^{(1)} \in R$$
$$\vdots$$
$$x^{(m)} \in R^2 \rightarrow z^{(m)} \in R$$

Data Compression

- Reduces the required time and storage space
- Removing multi-collinearity improves the interpretation of the parameters of the machine learning model.



$$x^{(1)} \in R^2 \rightarrow z^{(1)} \in R$$

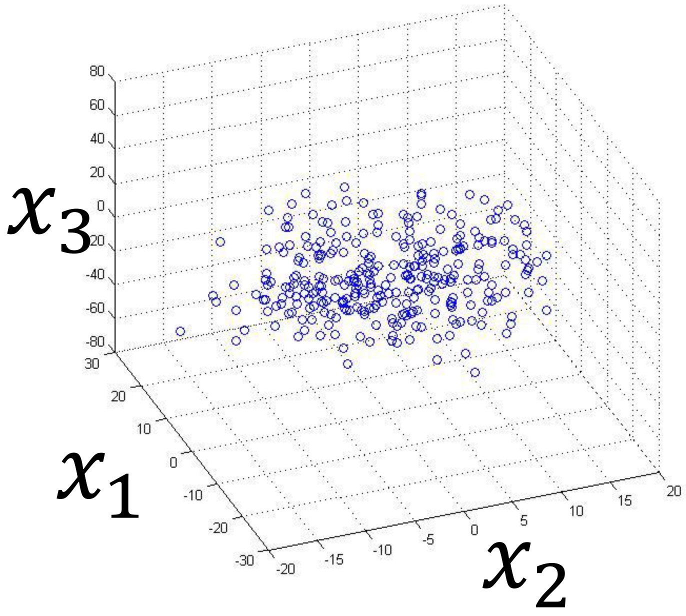
$$x^{(2)} \in R^2 \rightarrow z^{(1)} \in R$$

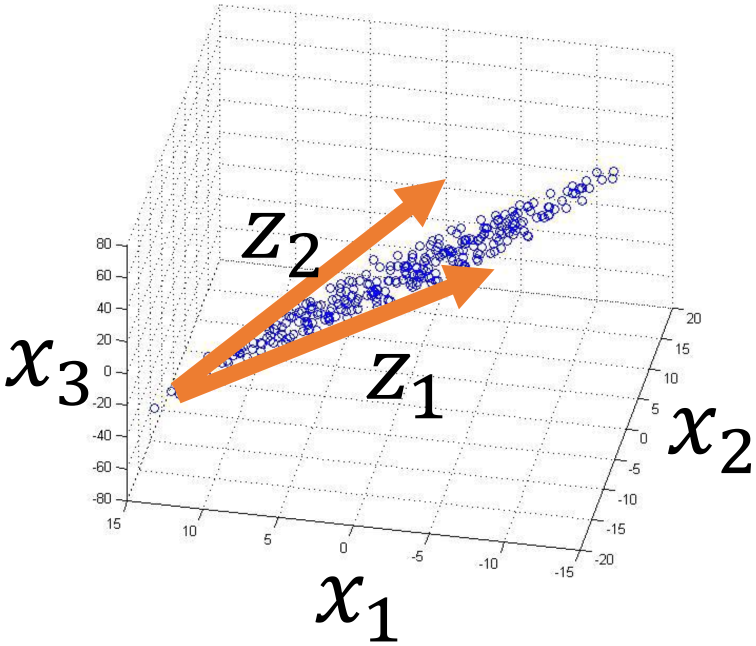
$$\vdots$$

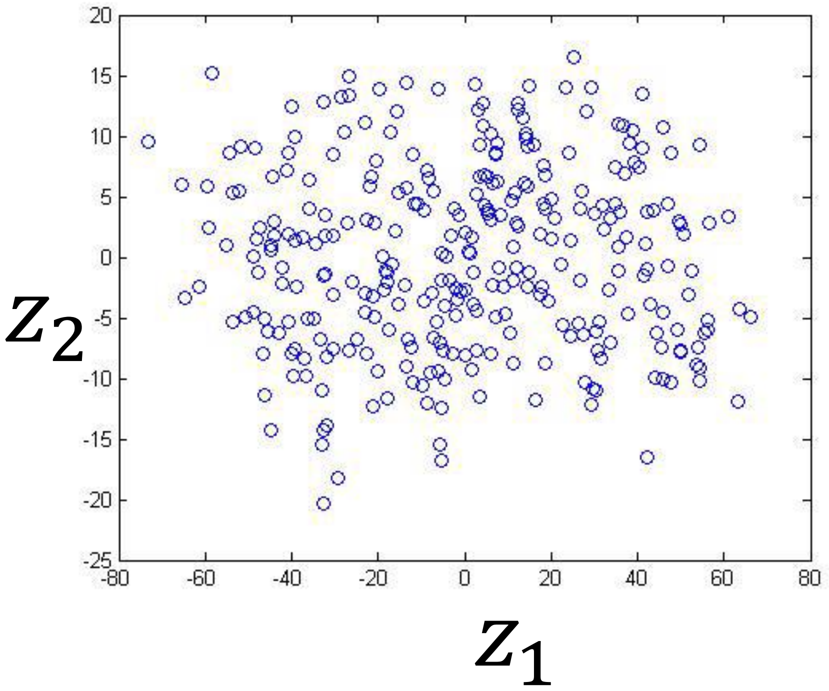
$$x^{(m)} \in R^2 \rightarrow z^{(m)} \in R$$

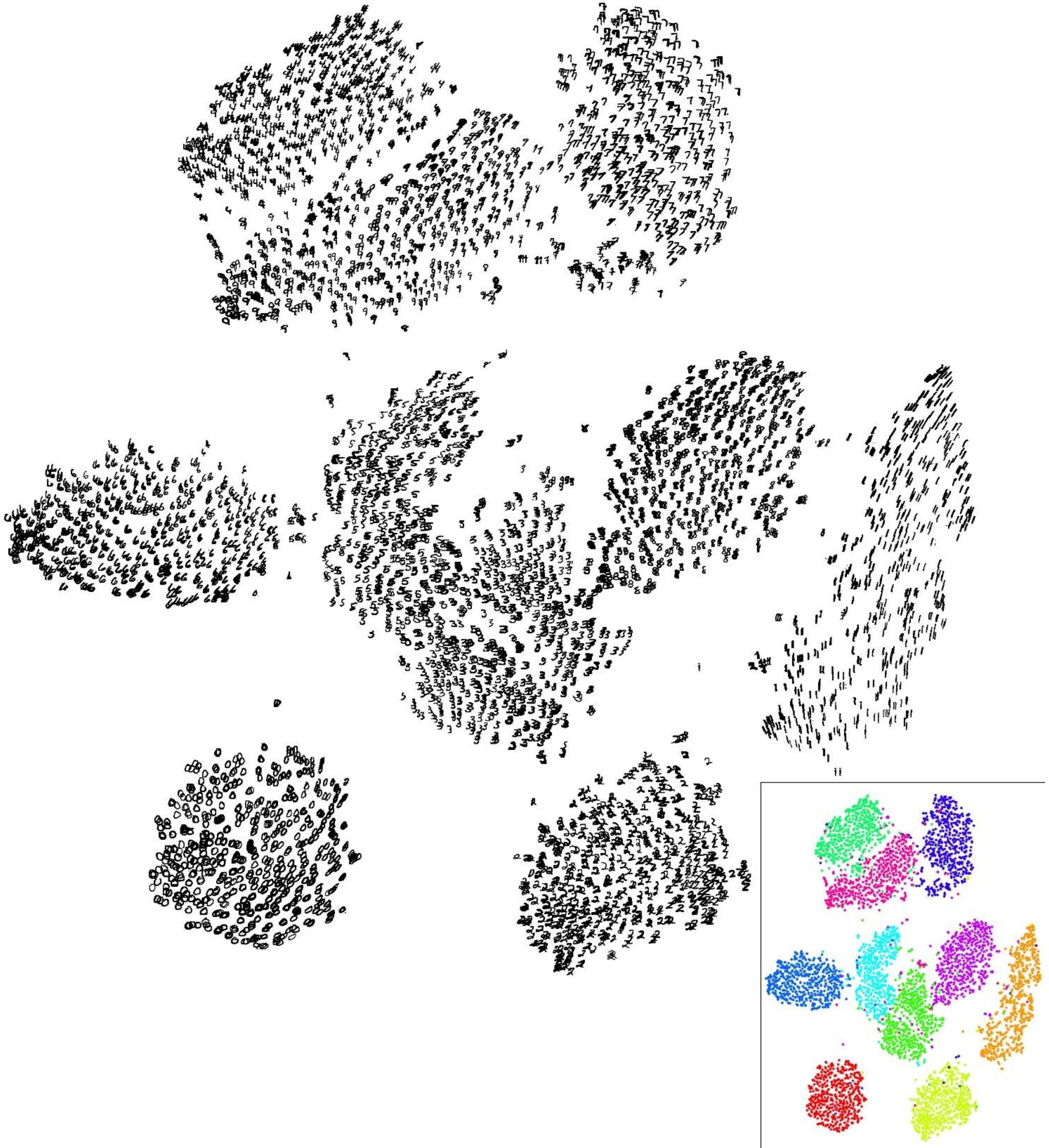
Data Compression

- Reduce data from 3D to 2D (in general 1000D -> 100D)





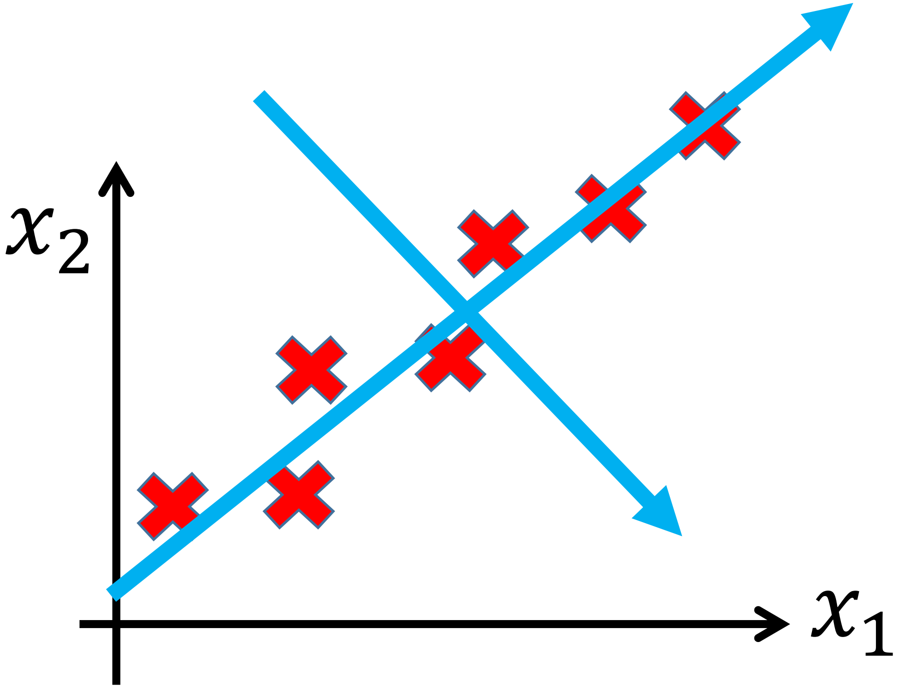




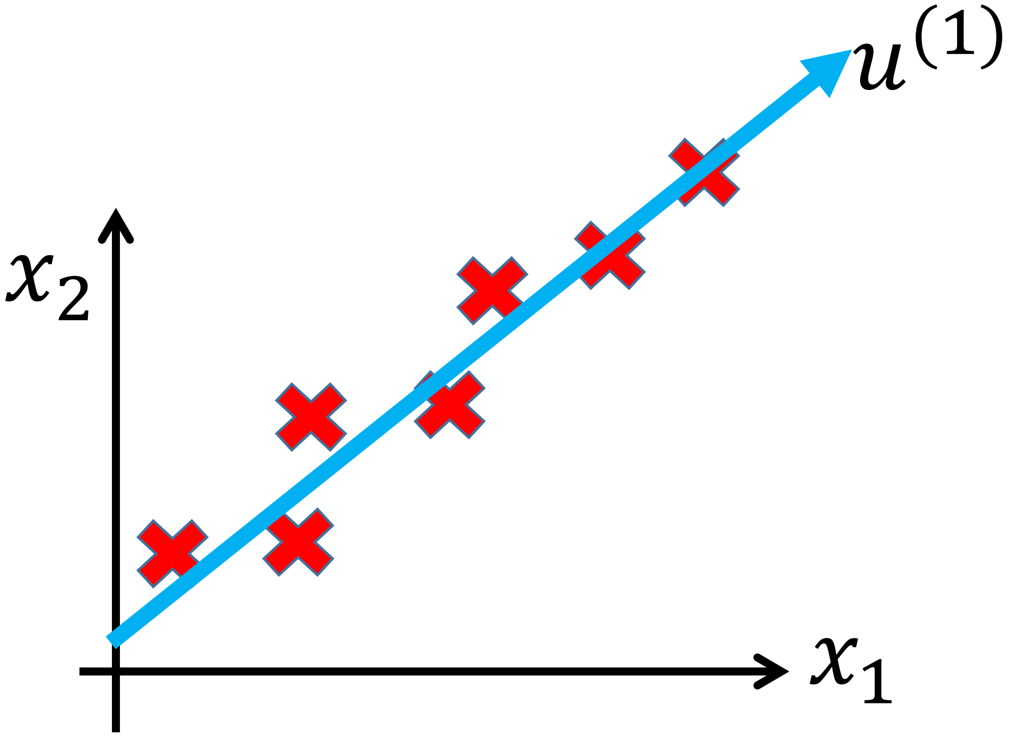
Dimensionality Reduction

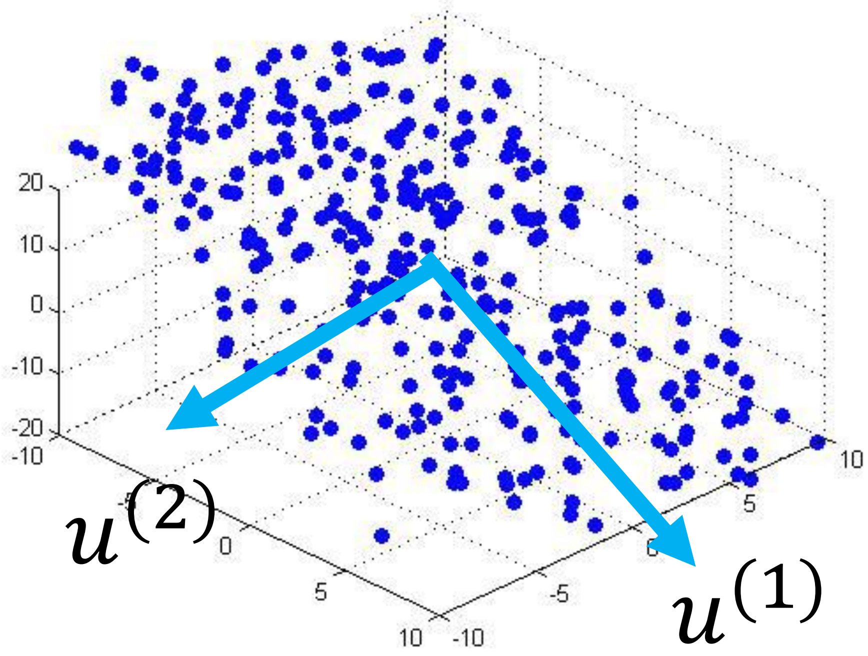
- Motivation
- **Principal component analysis**
  - **Formulation**
  - **Algorithm**
  - **Reconstruction**
- Choosing the number of principal components
- Applying PCA

Principal Component Analysis Formulation



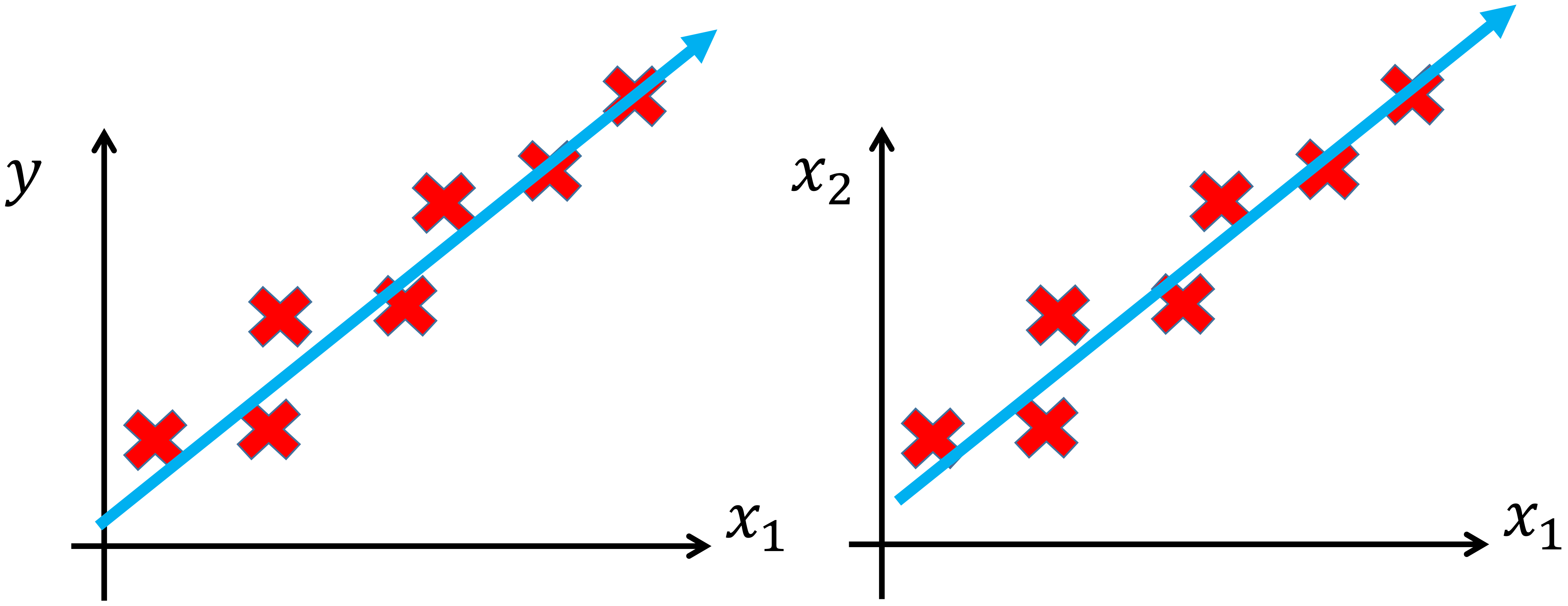
Principal Component Analysis Formulation





- Reduce n-D to k-D: find $u^{(1)}, u^{(2)}, \cdots, u^{(k)} \in R^n$ onto which to project the data, so as to minimize the projection error

PCA vs. Linear regression



Data pre-processing

- Training set: $x^{(1)}, x^{(2)}, \cdots, x^{(m)}$
- Preprocessing (feature scaling/mean normalization)

$$\mu_j = \frac{1}{m} \sum_i x_j^{(i)}$$

Replace each $x_j^{(i)}$ with $x_j - \mu_j$

If different features on different scales, scale features to have comparable range of values

$$x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{s_j}$$

Principal Component Analysis Algorithm

- Goal: Reduce data from n-dimensions to k-dimensions
- Step 1: Compute "covariance matrix"

$$\Sigma = \frac{1}{m}\sum_{i=1}^{n}\left(x^{(i)}\right)\left(x^{(i)}\right)^{\top}$$

- Step 2: Compute "eigenvectors" of the covariance matrix

```
[U, S, V] = svd(Sigma);
```

$$\mathrm{U} = \left[u^{(1)}, u^{(2)}, \cdots, u^{(n)}\right] \in R^{n\times n}$$

Principal components: $u^{(1)}, u^{(2)}, \cdots, u^{(k)} \in R^{n}$

Principal Component Analysis Algorithm

- Goal: Reduce data from n-dimensions to k-dimensions
- Principal components: $u^{(1)}, u^{(2)}, \cdots, u^{(k)} \in R^n$

$$z^{(i)} = \left[u^{(1)}, u^{(2)}, \cdots, u^{(k)}\right]^{\top} x^{(i)} \in R^k$$

PCA algorithm summary

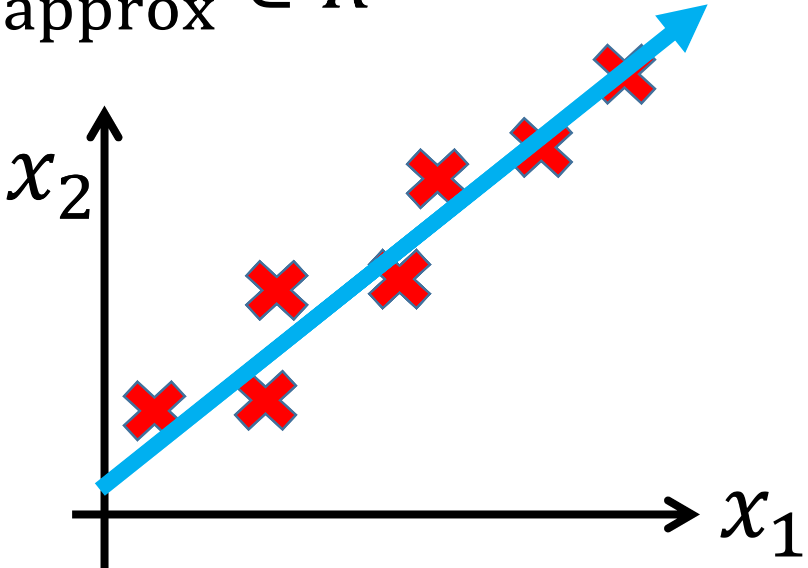
- After mean normalization (ensure every feature has zero mean) and optionally feature scaling

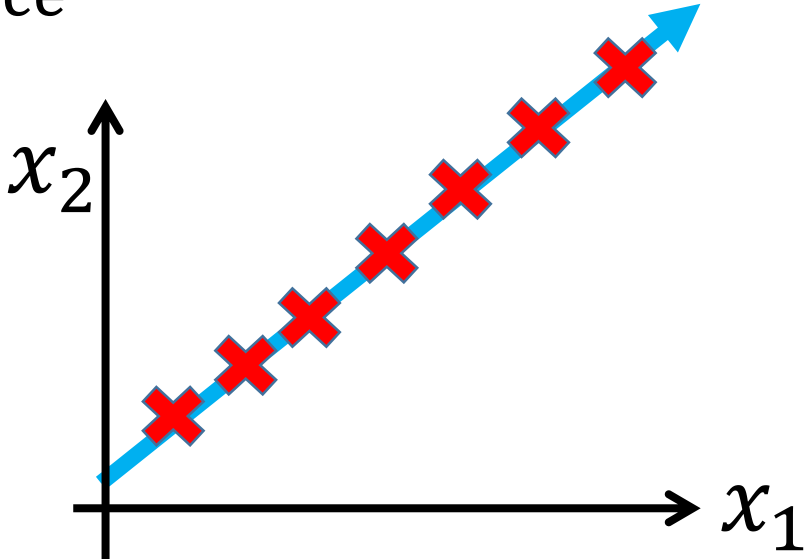
- `Simga =` $\frac{1}{m}\sum_{i=1}^{n}\left(x^{(i)}\right)\left(x^{(i)}\right)^{\mathsf{T}}$
- `[U, S, V] = svd(Sigma);`
- `Ureduce = U(:, 1:k);`
- `z = Ureduce' * x;`

Reconstruction from compressed representation

- Compression: $z^{(i)} = U_{\text{reduce}}^{\top} x^{(i)}$
- Reconstruction: $x_{\text{approx}}^{(i)} = U_{\text{reduce}} z^{(i)}$

- $x_{\text{approx}}^{(i)} \in R^{n}$ $\qquad U_{\text{reduce}} \in R^{n \times k}$ $\qquad z^{(i)} \in R^{k \times 1}$





3D face modeling

A morphable model for the synthesis of 3D faces, SIGGRAPH 1999

Shape modeling

SMPL: Skinned multi-person linear model, SIGGRAPH Asia 2015

Dimensionality Reduction

- Motivation
- Principal component analysis
  - Formulation
  - Algorithm
  - Reconstruction
- **Choosing the number of principal components**
- Applying PCA

How do we choose k
(number of principal components)

- Average squared projection error: $\frac{1}{\mathrm{m}}\sum_i \left\| x^{(i)} - x_{\mathrm{approx}}^{(i)} \right\|^2$
- Total variation in the data: $\frac{1}{\mathrm{m}}\sum_i \left\| x^{(i)} \right\|^2$

- Typically, choose k to be the smallest value so that

$$\frac{\frac{1}{\mathrm{m}}\sum_i \left\| x^{(i)} - x_{\mathrm{approx}}^{(i)} \right\|^2}{\frac{1}{\mathrm{m}}\sum_i \left\| x^{(i)} \right\|^2} \leq 0.01 \ (1\%)$$

"99% of variance is retained"

How do we choose k (number of principal components)

- Try PCA with $\mathrm{k} = 1, 2, \cdots$
- Compute $\mathrm{U}_{\mathrm{reduce}}, \mathrm{z}^{(1)}, \mathrm{z}^{(2)}, \cdots, \mathrm{z}^{(m)}$,

$$x_{\mathrm{approx}}^{(1)}, x_{\mathrm{approx}}^{(2)}, \cdots, x_{\mathrm{approx}}^{(m)}$$

- Check if

$$\frac{\frac{1}{\mathrm{m}}\sum_i \left\| x^{(i)} - x_{\mathrm{approx}}^{(i)} \right\|^2}{\frac{1}{\mathrm{m}}\sum_i \left\| x^{(i)} \right\|^2} \leq 0.01 \text{ ?}$$

- [U, S, V] = svd(Sigma)
- $S = \begin{bmatrix} s_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & s_{nn} \end{bmatrix}$

- For given k

$$1 - \frac{\sum_{i=1}^{k} s_{ii}}{\sum_{i=1}^{n} s_{ii}} \leq 0.01$$

$$\frac{\sum_{i=1}^{k} s_{ii}}{\sum_{i=1}^{n} s_{ii}} \geq 0.99$$

Dimensionality Reduction

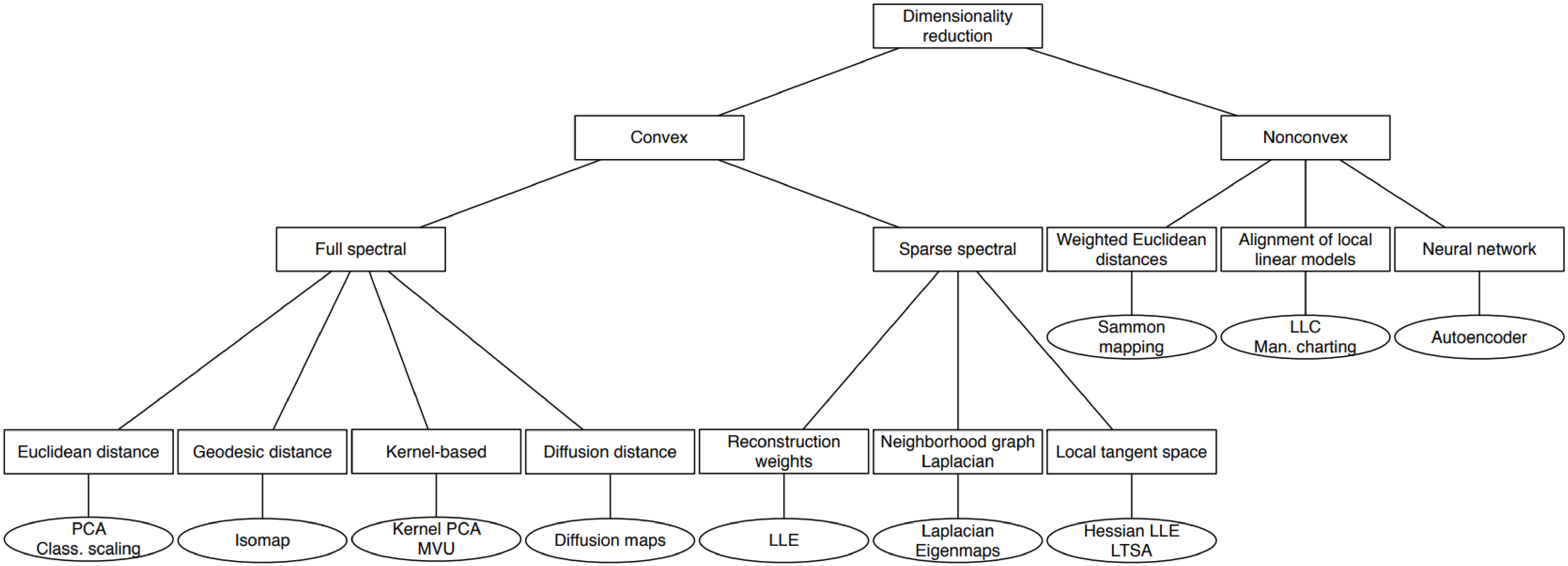
- Motivation
- Principal component analysis
  - Formulation
  - Algorithm
  - Reconstruction
- Choosing the number of principal components
- **Applying PCA**

Application of PCA

- Compression
  - Reduce memory/disk needed to store data
  - Speed up learning algorithm
- Visualization (k=2, k=3)

- Bad use of PCA
  - Reduce the number of features -> less likely to overfit?
  - Use regularization instead.

Taxonomy for dimensionality reduction



http://www.math.chalmers.se/Stat/Grundutb/GU/MSA220/S18/DimRed2.pdf

Things to remember

- Compression, visualization
- Principal component analysis
  - Formulation
  - Algorithm
  - Reconstruction
- Choosing the number of principal components
- Applying PCA