2018 EE448, Big Data Mining, Lecture 7

Unsupervised Learning

Weinan Zhang
Shanghai Jiao Tong University
http://wnzhang.net

http://wnzhang.net/teaching/ee448/index.html

ML Problem Setting

- First build and learn $p(x)$ and then infer the conditional dependence $p(x_t|x_i)$
  - Unsupervised learning
  - Each dimension of x is equally treated

- Directly learn the conditional dependence $p(x_t|x_i)$
  - Supervised learning
  - x_t is the label to predict

Definition of Unsupervised Learning

- Given the training dataset

$$D = \{x_i\}_{i=1,2,\ldots,N}$$

let the machine learn the data underlying patterns

  - Latent variables

$$z \longrightarrow x$$

  - Probabilistic density function (p.d.f.) estimation

$$p(x)$$

  - Good data representation (used for discrimination)

$$\phi(x)$$

Uses of Unsupervised Learning

- Data structure discovery, data science
- Data compression
- Outlier detection
- Input to supervised/reinforcement algorithms (causes may be more simply related to outputs or rewards)
- A theory of biological learning and perception

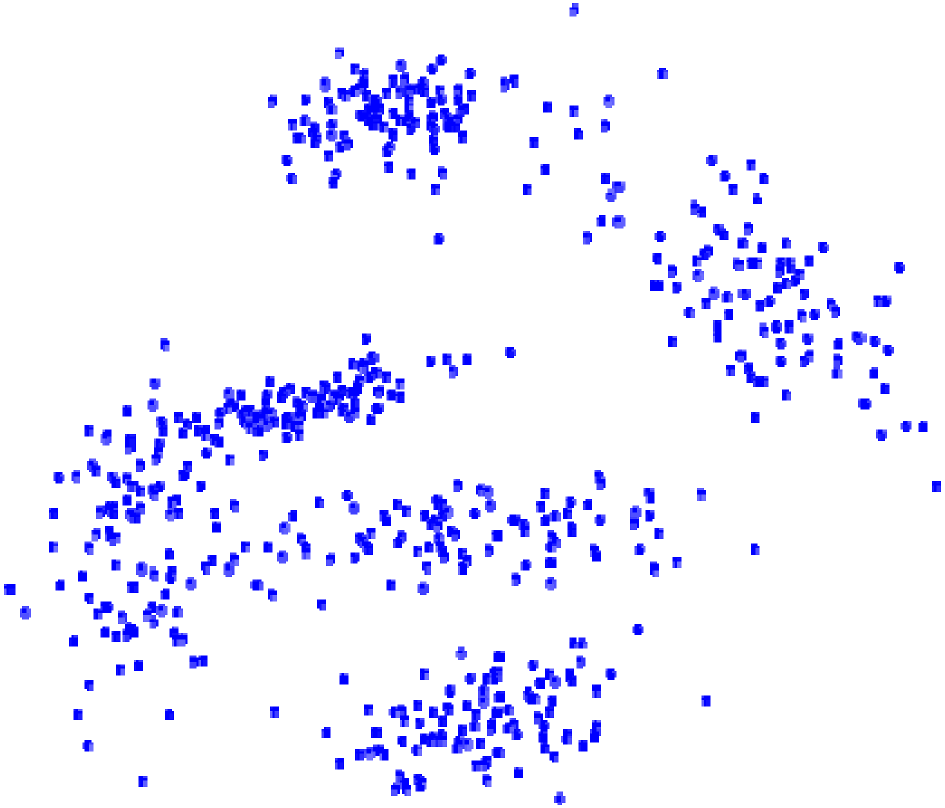
Slide credit: Maneesh Sahani

Content

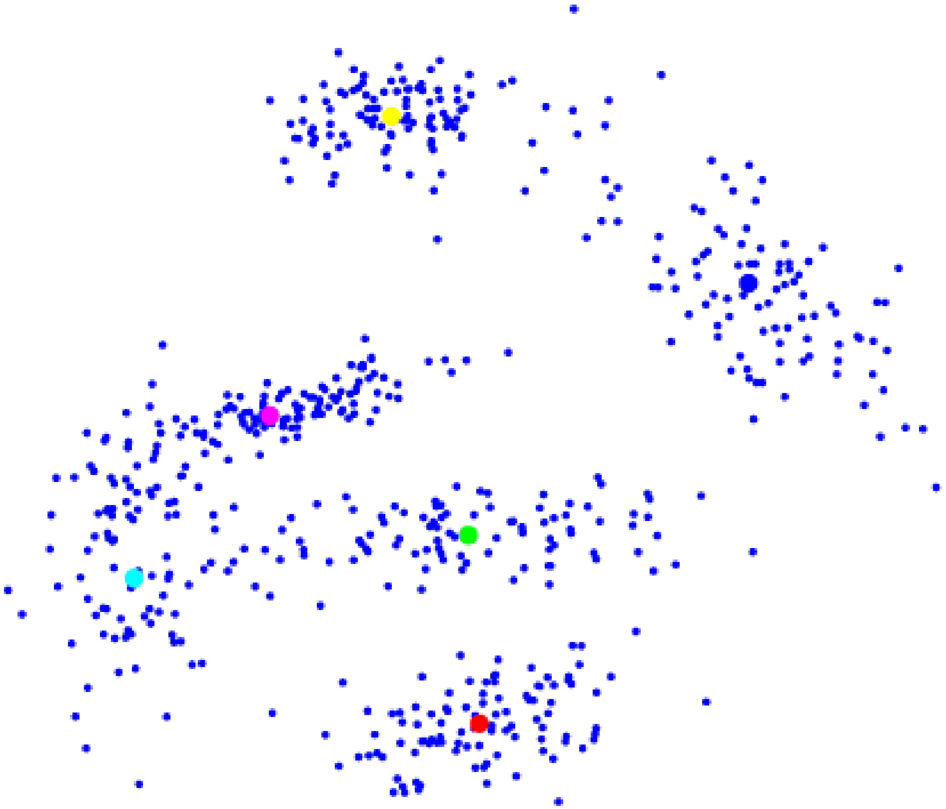
- Fundamentals of Unsupervised Learning
    - K-means clustering
    - Principal component analysis

- Probabilistic Unsupervised Learning
    - Mixture Gaussians
    - EM Methods

K-Means Clustering



K-Means Clustering



K-Means Clustering

- Provide the number of desired clusters *k*
- Randomly choose *k* instances as seeds, one per each cluster, i.e. the centroid for each cluster
- Iterate
  - Assign each instance to the cluster with the closest centroid
  - Re-estimate the centroid of each cluster
- Stop when clustering converges
  - Or after a fixed number of iterations

Slide credit: Ray Mooney

K-Means Clustering: Centriod

- Assume instances are real-valued vectors

$$x \in \mathbb{R}^d$$

- Clusters based on centroids, center of gravity, or mean of points in a cluster C_k

$$\mu^k = \frac{1}{C_k} \sum_{x \in C_k} x$$

Slide credit: Ray Mooney

K-Means Clustering: Distance

- Distance to a centroid $L(x, \mu^k)$
- Euclidian distance (L2 norm)

$$L_2(x, \mu^k) = \|x - \mu^k\| = \sqrt{\sum_{m=1}^{d} (x_i - \mu_m^k)^2}$$

- Euclidian distance (L1 norm)

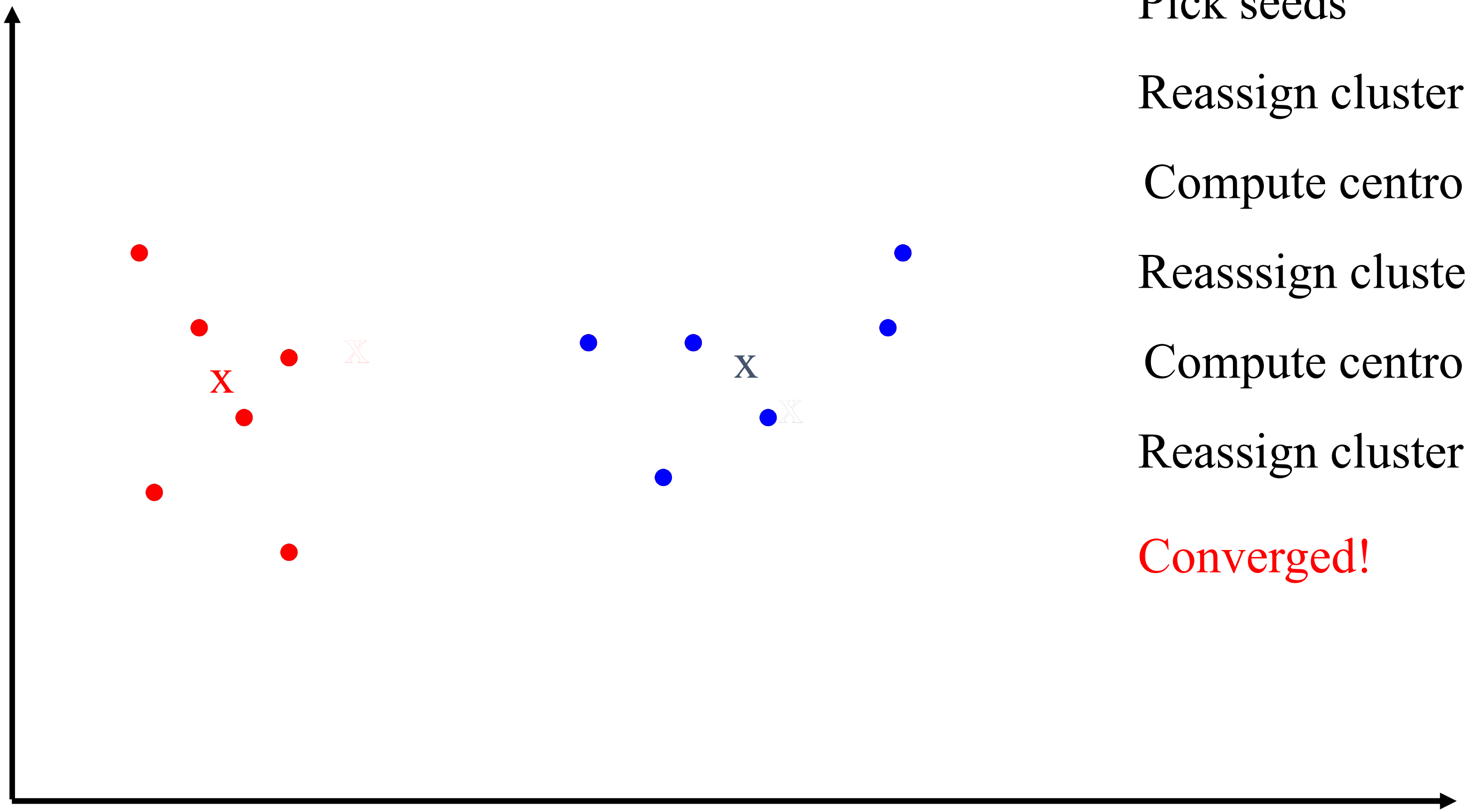
$$L_1(x, \mu^k) = |x - \mu^k| = \sum_{m=1}^{d} |x_i - \mu_m^k|$$

- Cosine distance

$$L_{\cos}(x, \mu^k) = 1 - \frac{x^\top \mu^k}{|x| \cdot |\mu^k|}$$

Slide credit: Ray Mooney

K-Means Example (K=2)



Pick seeds

Reassign clusters

Compute centroids

Reasssign clusters

Compute centroids

Reassign clusters

Converged!

Slide credit: Ray Mooney

K-Means Time Complexity

- Assume computing distance between two instances is $O(d)$ where d is the dimensionality of the vectors
- Reassigning clusters: $O(knd)$ distance computations
- Computing centroids: Each instance vector gets added once to some centroid: $O(nd)$
- Assume these two steps are each done once for I iterations: $O(Iknd)$

Slide credit: Ray Mooney

K-Means Clustering Objective

- The objective of *K*-means is to minimize the total sum of the squared distance of every point to its corresponding cluster centroid

$$\min_{\{\mu^k\}_{k=1}^K} \sum_{k=1}^{K} \sum_{x \in C_k} L(x - \mu^k) \qquad \mu^k = \frac{1}{C_k} \sum_{x \in C_k} x$$

- Finding the global optimum is NP-hard.
- The *K*-means algorithm is guaranteed to converge a local optimum.

Seed Choice

- Results can vary based on random seed selection.

- Some seeds can result in poor convergence rate, or convergence to sub-optimal clusterings.

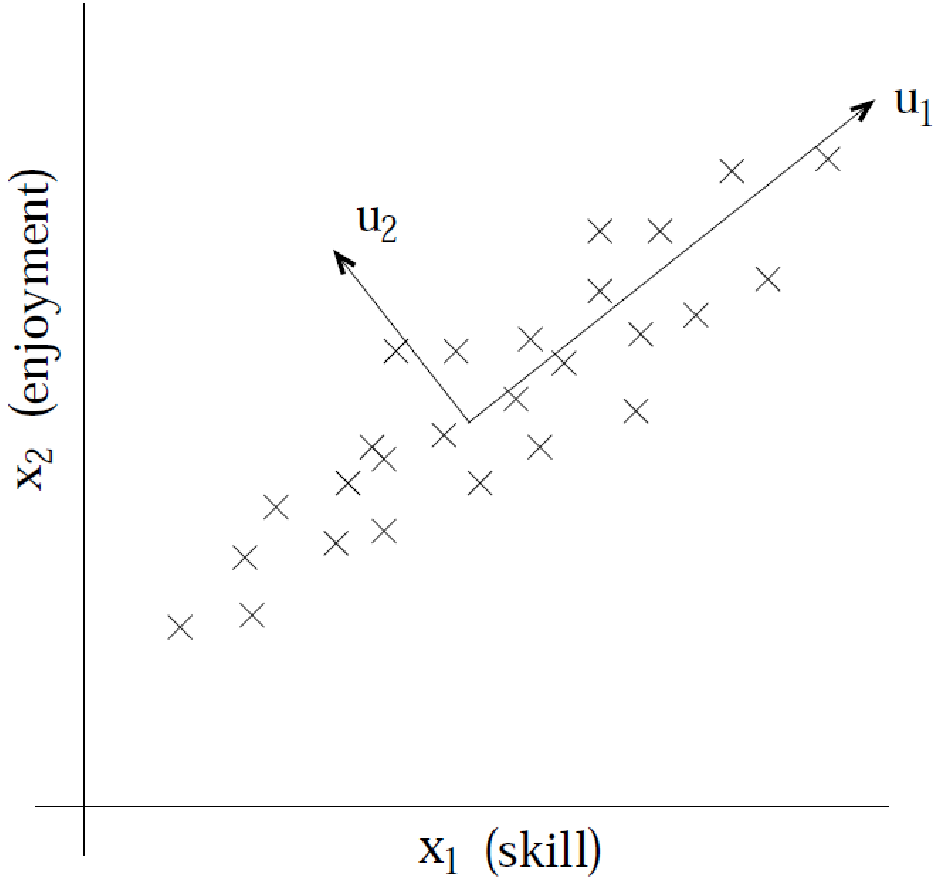
- Select good seeds using a heuristic or the results of another method.

Clustering Applications

- Text mining
  - Cluster documents for related search
  - Cluster words for query suggestion
- Recommender systems and advertising
  - Cluster users for item/ad recommendation
  - Cluster items for related item suggestion
- Image search
  - Cluster images for similar image search and duplication detection
- Speech recognition or separation
  - Cluster phonetical features

Principal Component Analysis (PCA)

- An example of 2-dimensional data
  - x_1: the piloting skill of pilot
  - x_2: how much he/she enjoys flying
- Main components
  - u_1: intrinsic piloting "karma" of a person
  - u_2: some noise



Example credit: Andrew Ng

Principal Component Analysis (PCA)

- PCA tries to identify the subspace in which the data approximately lies
- PCA uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.
  - The number of principal components is less than or equal to the smaller of the number of original variables or the number of observations.

$$\mathbb{R}^d \to \mathbb{R}^k \qquad k \ll d$$

PCA Data Preprocessing

- Given the dataset

$$D = \{x^{(i)}\}_{i=1}^{m}$$

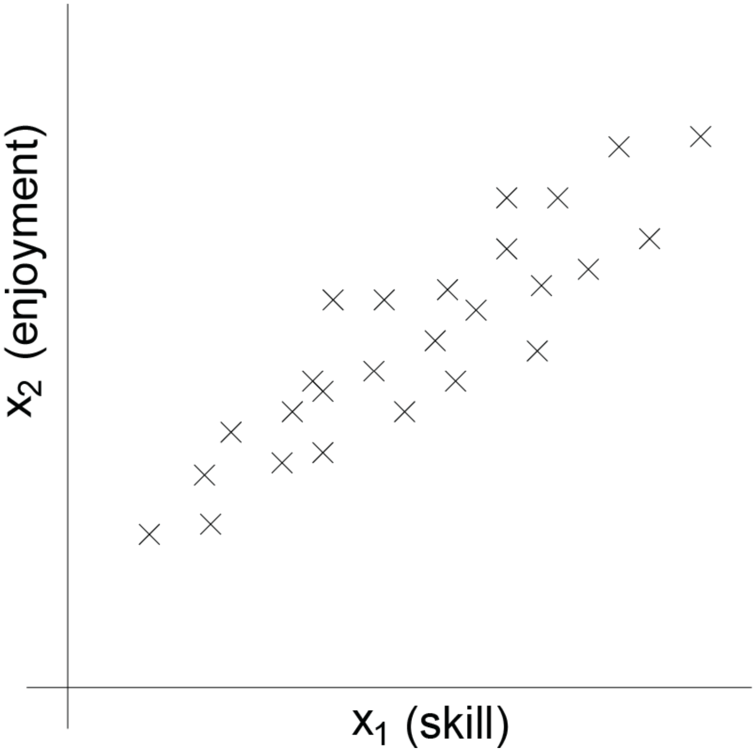
- Typically we first pre-process the data to normalize its mean and variance
  1. Move the central of the data set to 0

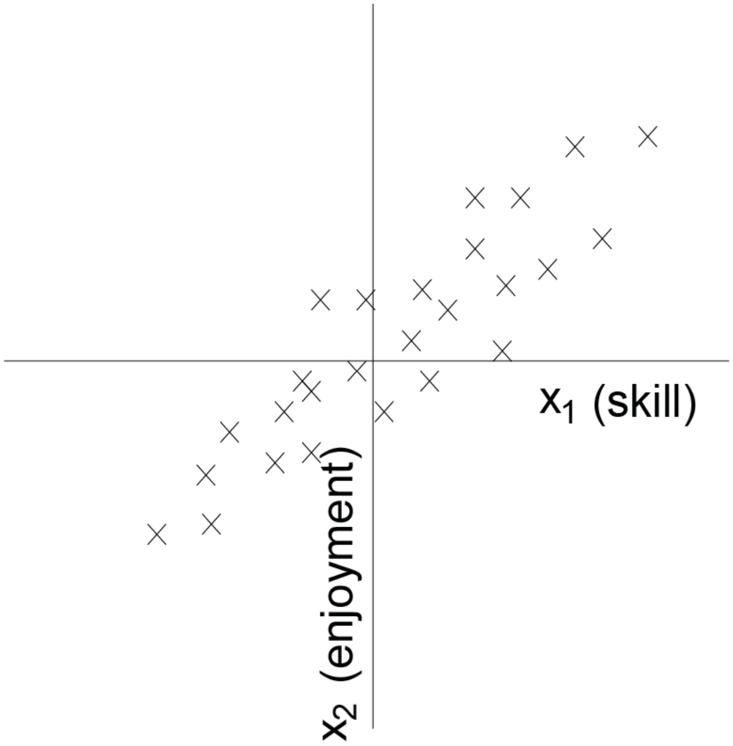
$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} \qquad\qquad x^{(i)} \leftarrow x^{(i)} - \mu$$

  2. Unify the variance of each variable

$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m} (x_j^{(i)})^2 \qquad\qquad x^{(i)} \leftarrow x^{(i)}/\sigma_j$$

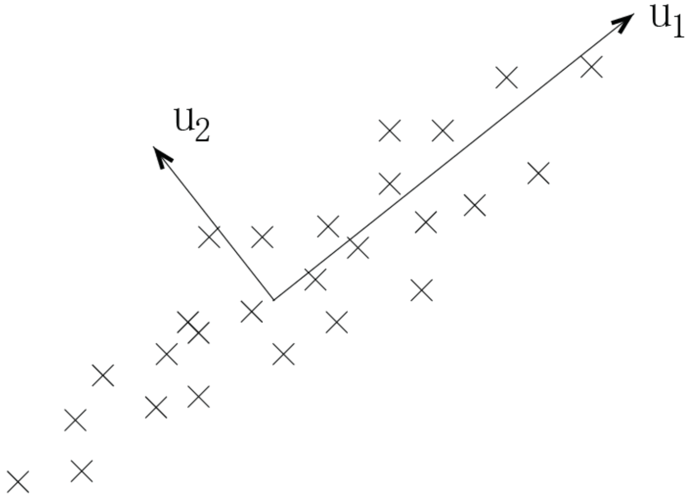
PCA Data Preprocessing

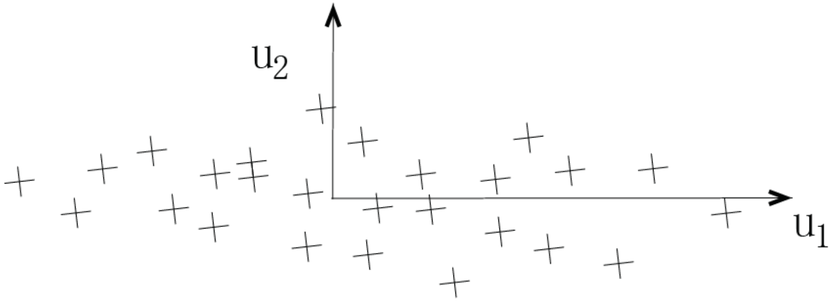




- Zero out the mean of the data
- Rescale each coordinate to have unit variance, which ensures that different attributes are all treated on the same "scale".

PCA Solution





- PCA finds the directions with the largest variable variance
  - which correspond to the eigenvectors of the matrix $X^{\mathsf{T}}X$ with the largest eigenvalues

PCA Solution: Data Projection

- The projection of each point $x^{(i)}$ to a direction u $\quad (\|u\| = 1)$

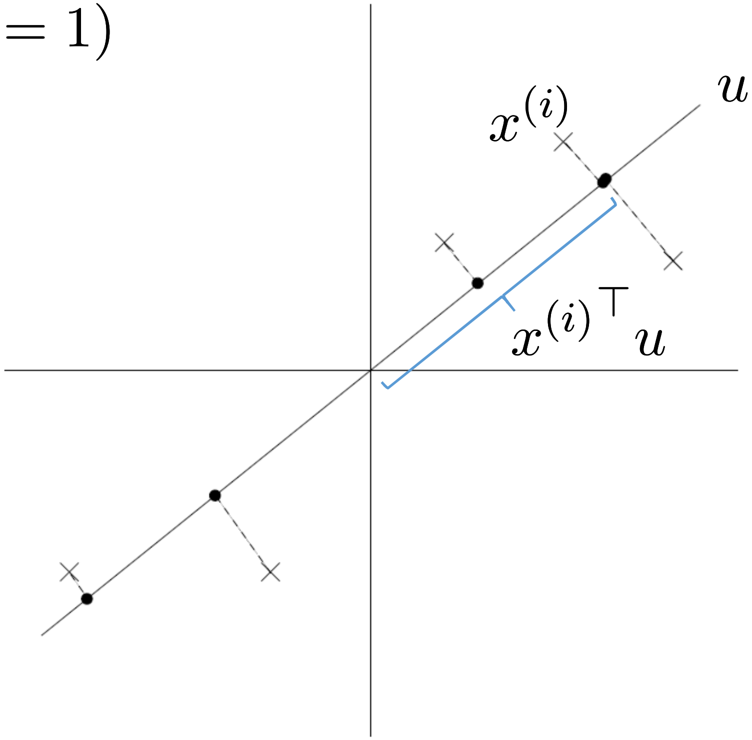
$$x^{(i)\top} u$$

- The variance of the projection

$$\frac{1}{m}\sum_{i=1}^{m}(x^{(i)\top}u)^2 = \frac{1}{m}\sum_{i=1}^{m} u^\top x^{(i)} x^{(i)\top} u$$

$$= u^\top \Big(\frac{1}{m}\sum_{i=1}^{m} x^{(i)} x^{(i)\top}\Big) u$$

$$\equiv u^\top \Sigma u$$



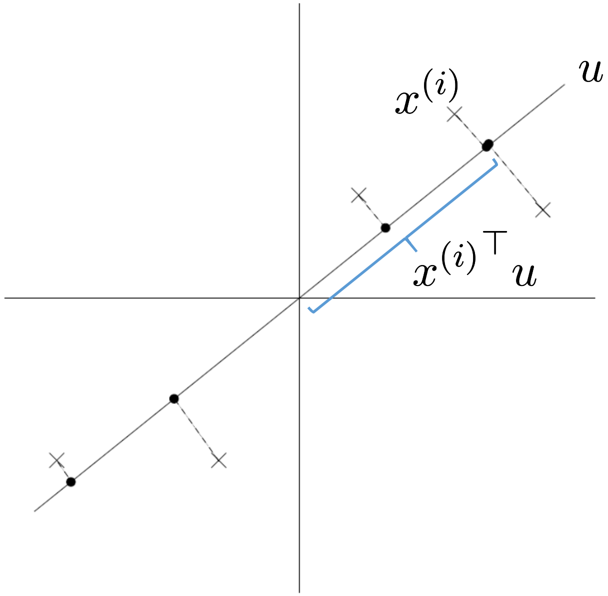
PCA Solution: Largest Eigenvalues

$$\max_{u} \quad u^\top \Sigma u \qquad\qquad \Sigma = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} {x^{(i)}}^\top$$

$$\text{s.t.} \quad \|u\| = 1$$

- Find *k* principal components of the data is to find the *k* principal eigenvectors of Σ
  - i.e. the top-*k* eigenvectors with the largest eigenvalues
- Projected vector for $x^{(i)}$

$$y^{(i)} = \begin{bmatrix} u_1^\top x^{(i)} \\ u_2^\top x^{(i)} \\ \vdots \\ u_k^\top x^{(i)} \end{bmatrix} \in \mathbb{R}^k$$



Eigendecomposition Revisit

- For a semi-positive square matrix $\Sigma_{d\times d}$
  - suppose *u* to be its eigenvector $(\|u\| = 1)$
  - with the scalar eigenvalue *w* $\Sigma u = wu$
  - There are *d* eigenvectors-eigenvalue pairs (u_i, w_i)
  - These *d* eigenvectors are orthogonal, thus they form an orthonormal basis

$$\sum_{i=1}^{d} u_i u_i^\top = I$$

  - Thus any vector *v* can be written as

$$v = \Big(\sum_{i=1}^{d} u_i u_i^\top\Big) v = \sum_{i=1}^{d} (u_i^\top v) u_i = \sum_{i=1}^{d} v_{(i)} u_i$$

$$U = [u_1, u_2, \ldots, u_d]$$

  - $\Sigma_{d\times d}$ can be written as

$$\Sigma = \sum_{i=1}^{d} u_i u_i^\top \Sigma = \sum_{i=1}^{d} w_i u_i u_i^\top = UWU^\top \qquad W = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & w_d \end{bmatrix}$$

Eigendecomposition Revisit

- Given the data $X = \begin{bmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_n^\top \end{bmatrix}$ and its covariance matrix $\Sigma = X^\top X$

  (here we may drop *m* for simplicity)

- The variance in direction u_i is

$$\|Xu_i\|^2 = u_i^\top X^\top X u_i = u_i^\top \Sigma u_i = u_i^\top w_i u_i = w_i$$

- The variance in any direction *v* is

$$\|Xv\|^2 = \left\|X\left(\sum_{i=1}^{d} v_{(i)} u_i\right)\right\|^2 = \sum_{ij} v_{(i)} u_i^\top \Sigma u_i v_{(j)} = \sum_{i=1}^{d} v_{(i)}^2 w_i$$

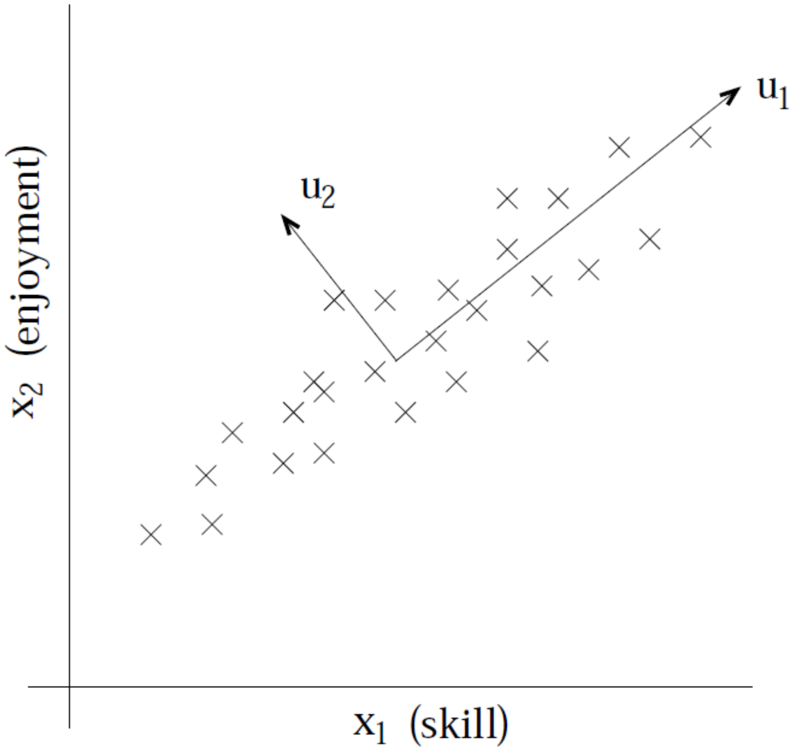
  where $v_{(i)}$ is the projection length of *v* on u_i

- If $v^\top v = 1$, then $\arg\max_{\|v\|=1} \|Xv\|^2 = u_{(\max)}$

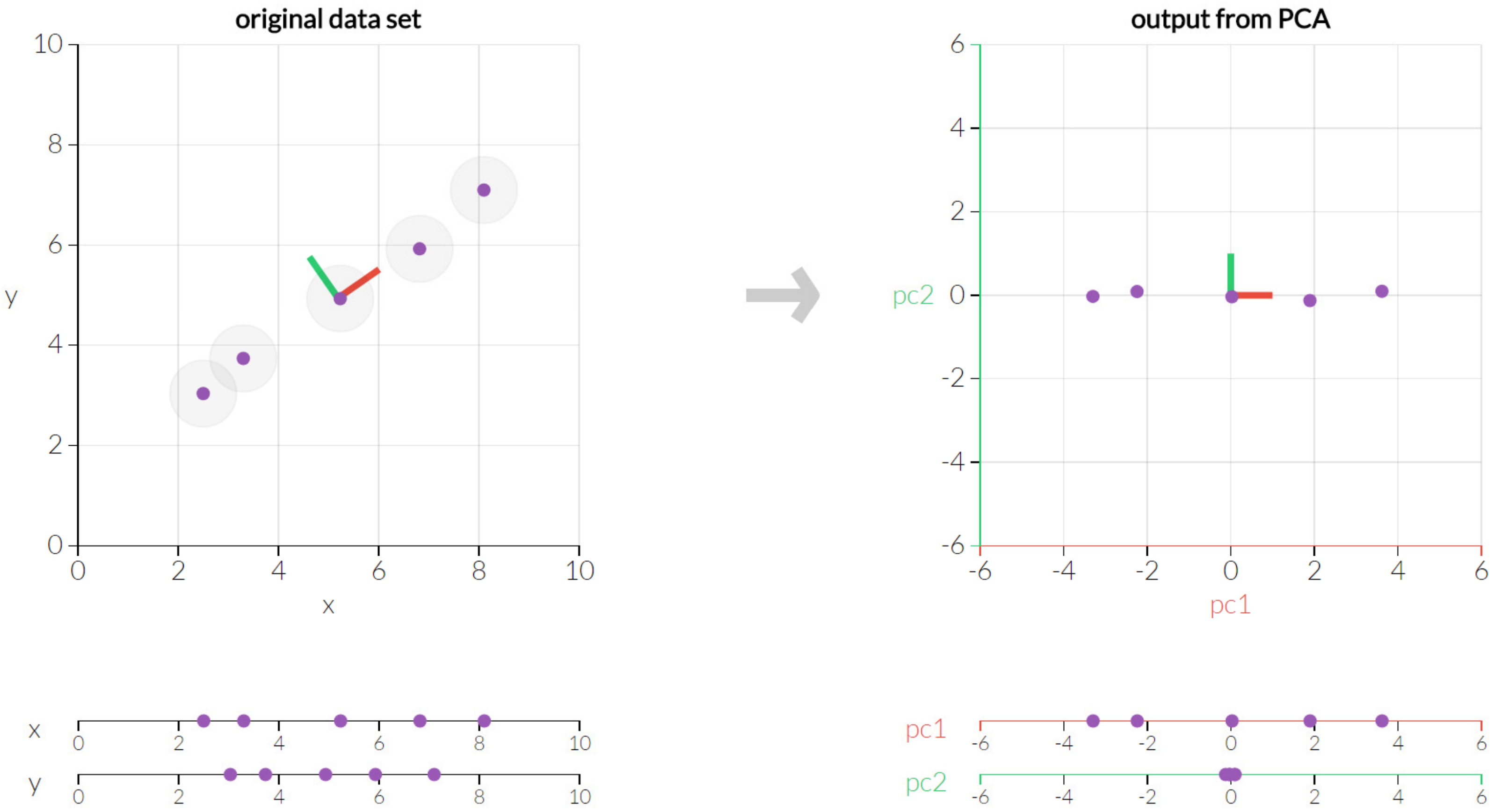
The direction of greatest variance is the eigenvector with the largest eigenvalue

PCA Discussion

- PCA can also be derived by picking the basis that minimizes the approximation error arising from projecting the data onto the *k*-dimensional subspace spanned by them.

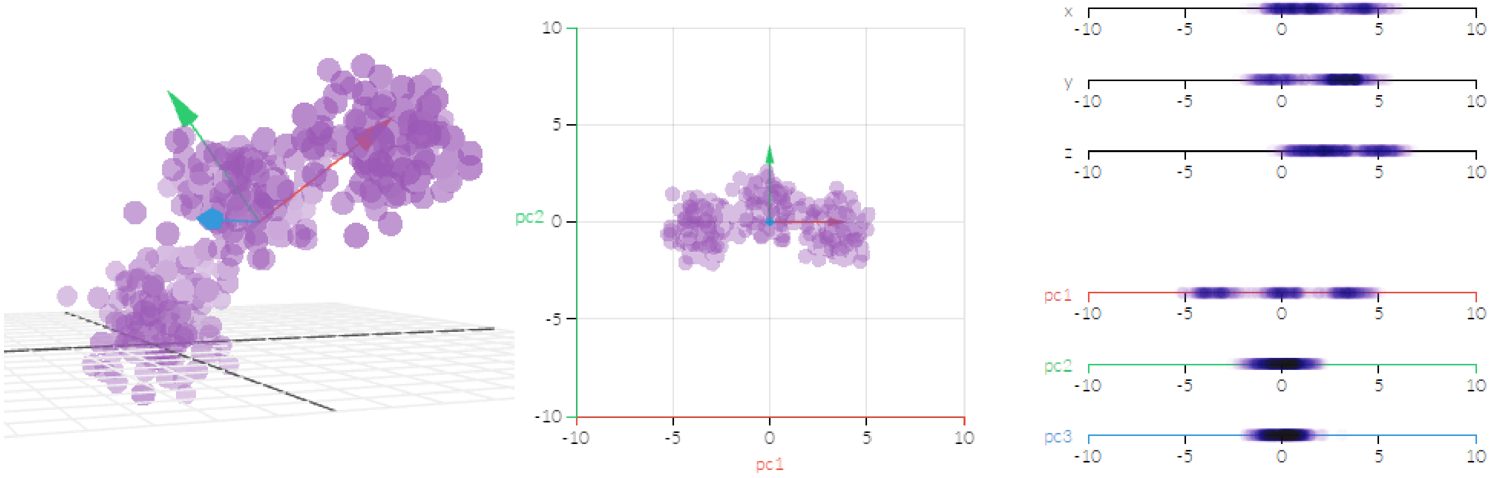


PCA Visualization



http://setosa.io/ev/principal-component-analysis/

PCA Visualization



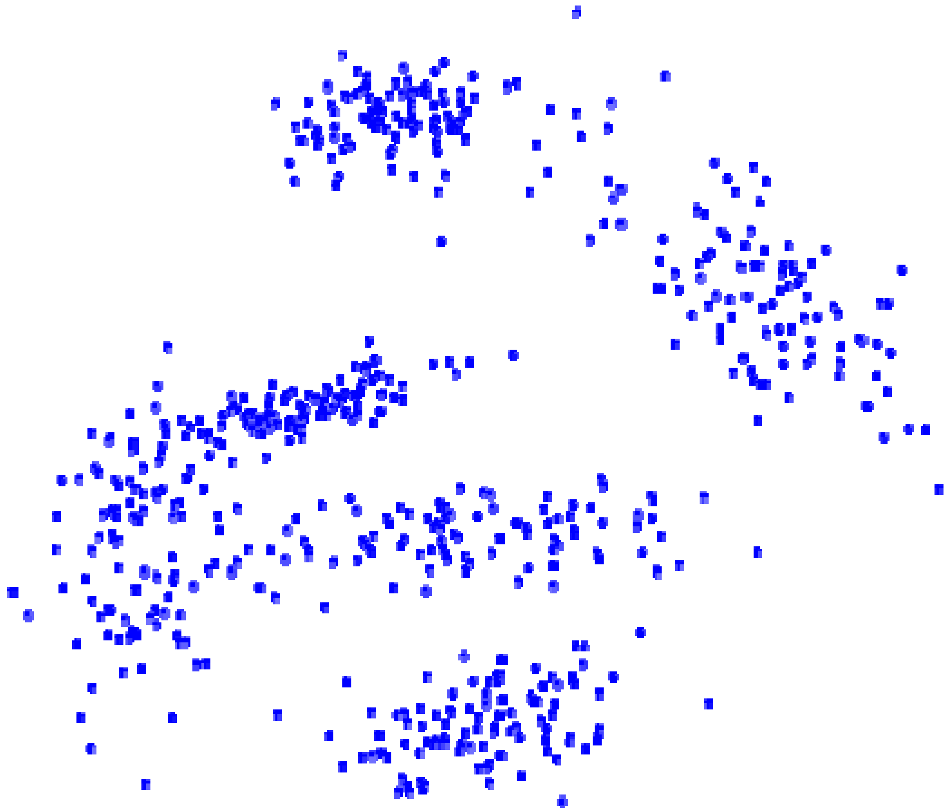
http://setosa.io/ev/principal-component-analysis/

Content

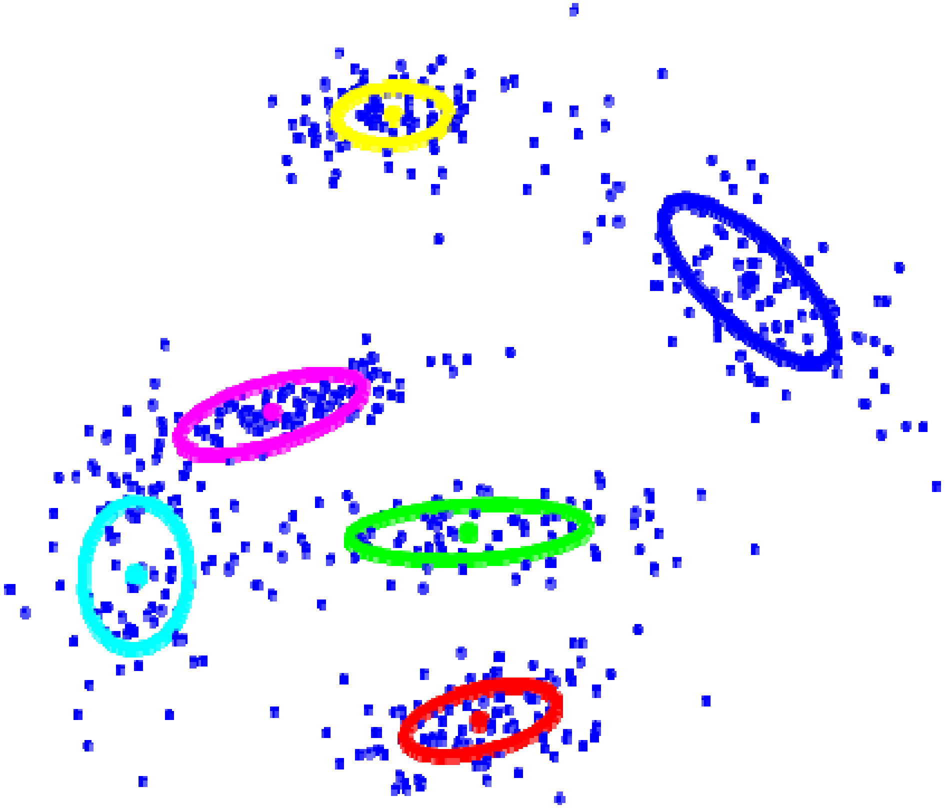
- Fundamentals of Unsupervised Learning
  - K-means clustering
  - Principal component analysis

- Probabilistic Unsupervised Learning
  - Mixture Gaussians
  - EM Methods

Mixture Gaussian



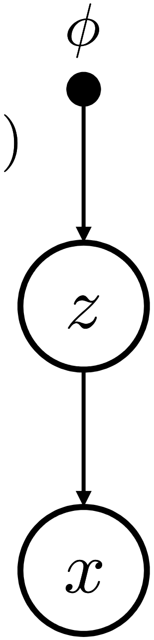
Mixture Gaussian



Graphic Model for Mixture Gaussian

- Given a training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$
- Model the data by specifying a joint distribution

$$p(x^{(i)}, z^{(i)}) = p(x^{(i)}|z^{(i)})p(z^{(i)})$$



Parameters of latent variable distribution

$$z^{(i)} \sim \text{Multinomial}(\phi)$$

$$p(z^{(i)} = j) = \phi_j$$

Latent variable: the Gaussian cluster ID
Indicates which Gaussian each *x* comes from

$$x^{(i)} \sim \mathcal{N}(\mu_j, \Sigma_j)$$

Observed data points

Data Likelihood

- We want to maximize

$$
\begin{aligned}
l(\phi, \mu, \Sigma) &= \sum_{i=1}^{m} \log p(x^{(i)}; \phi, \mu, \Sigma) \\
&= \sum_{i=1}^{m} \log \sum_{z^{(i)}=1}^{k} p(x^{(i)} | z^{(i)}; \mu, \Sigma) p(z^{(i)}; \phi) \\
&= \sum_{i=1}^{m} \log \sum_{j=1}^{k} \mathcal{N}(x^{(i)} | \mu_j, \Sigma_j) \phi_j
\end{aligned}
$$

- No closed form solution by simply setting

$$
\frac{\partial l(\phi, \mu, \Sigma)}{\partial \phi} = 0 \qquad \frac{\partial l(\phi, \mu, \Sigma)}{\partial \mu} = 0 \qquad \frac{\partial l(\phi, \mu, \Sigma)}{\partial \Sigma} = 0
$$

Data Likelihood Maximization

- For each data point $x^{(i)}$, latent variable $z^{(i)}$ indicates which Gaussian it comes from
- If we knew $z^{(i)}$, the data likelihood

$$\begin{aligned} l(\phi, \mu, \Sigma) &= \sum_{i=1}^{m} \log p(x^{(i)}; \phi, \mu, \Sigma) \\ &= \sum_{i=1}^{m} \log p(x^{(i)} | z^{(i)}; \mu, \Sigma) p(z^{(i)}; \phi) \\ &= \sum_{i=1}^{m} \log \mathcal{N}(x^{(i)} | \mu_{z^{(i)}}, \Sigma_{z^{(i)}}) + \log p(z^{(i)}; \phi) \end{aligned}$$

Data Likelihood Maximization

- Given *z*$^{(i)}$, maximize the data likelihood

$$\max_{\phi,\mu,\Sigma} l(\phi,\mu,\Sigma) = \max_{\phi,\mu,\Sigma} \sum_{i=1}^{m} \log \mathcal{N}(x^{(i)} | \mu_{z^{(i)}}, \Sigma_{z^{(i)}}) + \log p(z^{(i)}; \phi)$$

- It is easy to get the solution

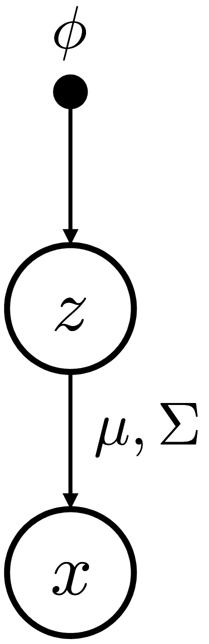
$$\phi_j = \frac{1}{m} \sum_{i=1}^{m} 1\{z^{(i)} = j\}$$

$$\mu_j = \frac{\sum_{i=1}^{m} 1\{z^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{z^{(i)} = j\}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{m} 1\{z^{(i)} = j\} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^\top}{\sum_{i=1}^{m} 1\{z^{(i)} = j\}}$$

Latent Variable Inference

- Given the parameters *μ, Σ, ϕ,* it is not hard to infer the posterior of the latent variable $z^{(i)}$ for each instance



$$p(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma) = \frac{p(z^{(i)} = j, x^{(i)}; \phi, \mu, \Sigma)}{p(x^{(i)}; \phi, \mu, \Sigma)}$$

$$= \frac{p(x^{(i)} | z^{(i)} = j; \mu, \Sigma) p(z^{(i)} = j; \phi)}{\sum_{l=1}^{k} p(x^{(i)} | z^{(i)} = l; \mu, \Sigma) p(z^{(i)} = l; \phi)}$$

where

- The prior of $z^{(i)}$ is $p(z^{(i)} = j; \phi)$
- The likelihood is $p(x^{(i)} | z^{(i)} = j; \mu, \Sigma)$

Expectation Maximization Methods

- E-step: infer the posterior distribution of the latent variables given the model parameters
- M-step: tune parameters to maximize the data likelihood given the latent variable distribution

- EM methods
  - Iteratively execute E-step and M-step until convergence

EM Methods for Mixture Gaussians

- Mixture Gaussian example

Repeat until convergence: {

(E-step) For each *i, j,* set

$$w_j^{(i)} = p(z^{(i)} = j, x^{(i)}; \phi, \mu, \Sigma)$$

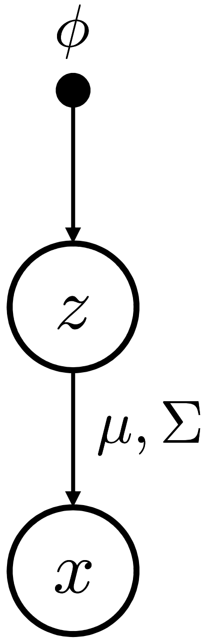
(M-step) Update the parameters

$$\phi_j = \frac{1}{m}\sum_{i=1}^{m} w_j^{(i)}$$

$$\mu_j = \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{m} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^\top}{\sum_{i=1}^{m} w_j^{(i)}}$$

}



General EM Methods

- Claims:

1. After each E-M step, the data likelihood will not decrease.
2. The EM algorithm finds a (local) maximum of a latent variable model likelihood

- Now let’s discuss the general EM methods and verify its effectiveness of improving data likelihood and its convergence

Jensen’s Inequality

Theorem. Let *f* be a convex function, and let *X* be a random variable.

Then:

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X])$$

- Moreover, if *f* is strictly convex, then

  $$\mathbb{E}[f(X)] = f(\mathbb{E}[X])$$

  holds true if and only if

  $$X = \mathbb{E}[X]$$

  with probability 1 (i.e., if *X* is a constant).

Jensen’s Inequality

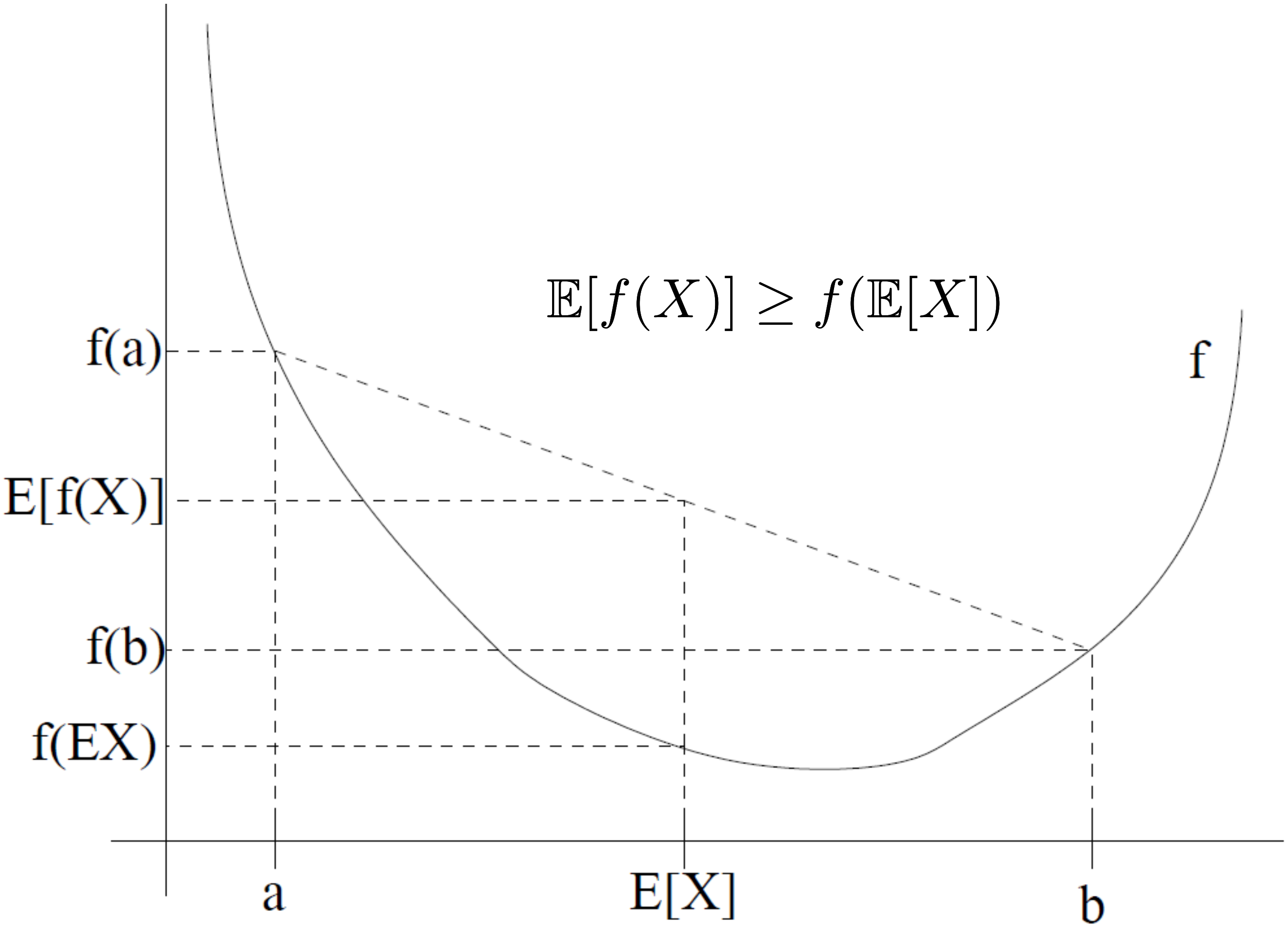


Figure credit: Andrew Ng

Jensen's Inequality

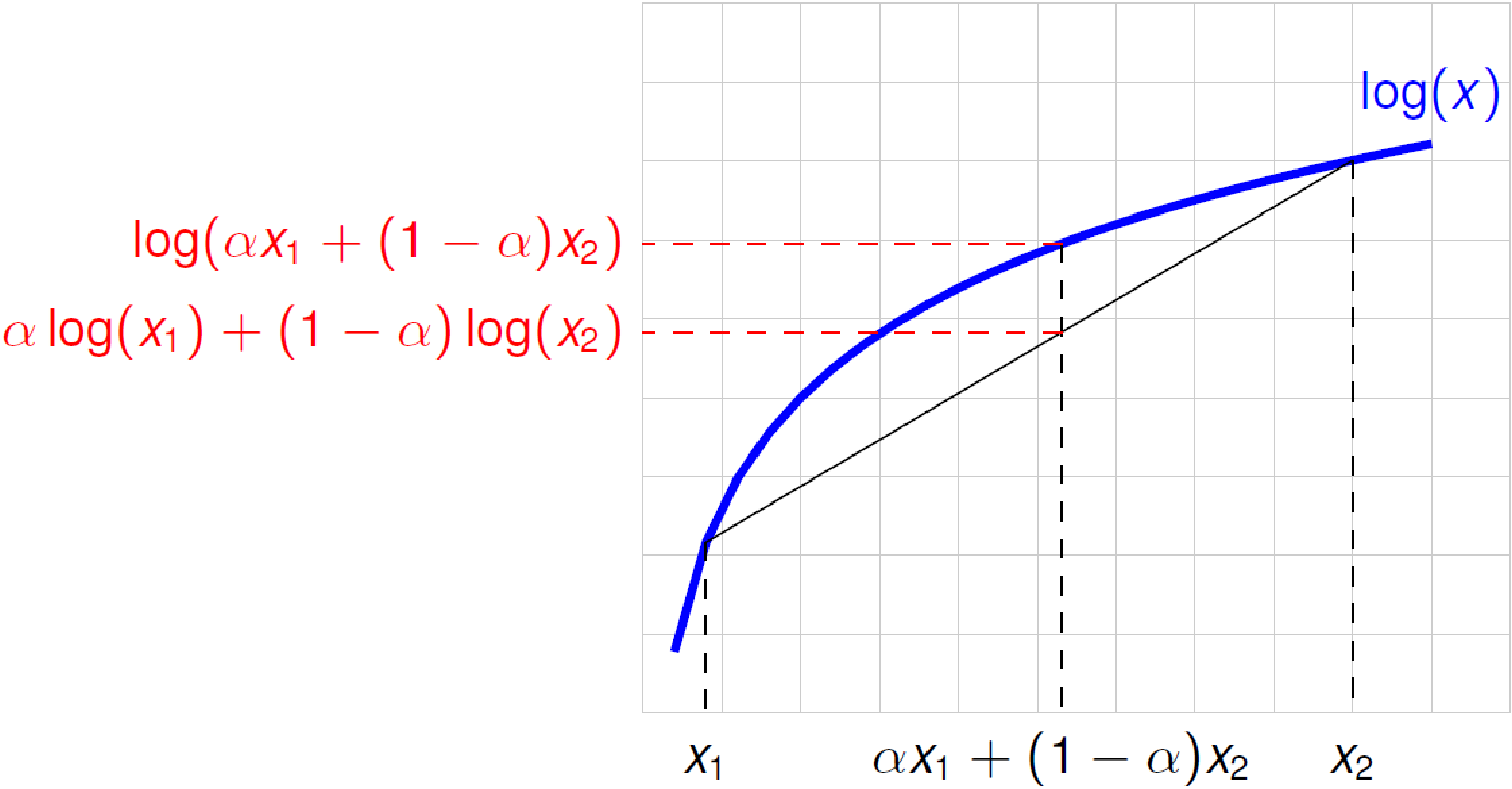


Figure credit: Maneesh Sahani

General EM Methods: Problem

- Given the training dataset

$$D = \{x_i\}_{i=1,2,\ldots,N}$$

  let the machine learn the data underlying patterns

- Assume latent variables

$$z \longrightarrow x$$

- We wish to fit the parameters of a model *p*(*x*,*z*) to the data, where the log-likelihood is

$$\begin{aligned} l(\theta) &= \sum_{i=1}^{N} \log p(x;\theta) \\ &= \sum_{i=1}^{N} \log \sum_{z} p(x,z;\theta) \end{aligned}$$

General EM Methods: Problems

- EM methods solve the problems where
  - Explicitly find the maximum likelihood estimation (MLE) is hard

$$\theta^* = \arg\max_\theta \sum_{i=1}^{N} \log \sum_z p(x^{(i)}, z^{(i)}; \theta)$$

  - But given $z^{(i)}$ observed, the MLE is easy

$$\theta^* = \arg\max_\theta \sum_{i=1}^{N} \log p(x^{(i)} | z^{(i)}; \theta)$$

- EM methods give an efficient solution for MLE, by iteratively doing
  - E-step: construct a (good) lower-bound of log-likelihood
  - M-step: optimize that lower-bound

General EM Methods: Lower Bound

- For each instance *i*, let q_i be some distribution of $z^{(i)}$

$$\sum_z q_i(z) = 1, \quad q_i(z) \geq 0$$

- Thus the data log-likelihood

$$\begin{aligned} l(\theta) = \sum_{i=1}^{N} \log p(x^{(i)};\theta) &= \sum_{i=1}^{N} \log \sum_{z^{(i)}} p(x^{(i)}, z^{(i)};\theta) \\ &= \sum_{i=1}^{N} \log \sum_{z^{(i)}} q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)};\theta)}{q_i(z^{(i)})} \\ &\geq \sum_{i=1}^{N} \sum_{z^{(i)}} q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)};\theta)}{q_i(z^{(i)})} \end{aligned}$$

Lower bound of $l(\vartheta)$

Jensen's inequality
-log(*x*) is a convex function

General EM Methods: Lower Bound

$$l(\theta) = \sum_{i=1}^{N} \log p(x^{(i)}; \theta) \geq \sum_{i=1}^{N} \sum_{z^{(i)}} q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{q_i(z^{(i)})}$$

- Then what $q_i(z)$ should we choose?

REVIEW

Jensen's Inequality

Theorem. Let *f* be a convex function, and let *X* be a random variable.

Then:

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X])$$

- Moreover, if *f* is strictly convex, then

  $$\mathbb{E}[f(X)] = f(\mathbb{E}[X])$$

  holds true if and only if

  $$X = \mathbb{E}[X]$$

  with probability 1 (i.e., if *X* is a constant).

General EM Methods: Lower Bound

$$l(\theta) = \sum_{i=1}^{N} \log p(x^{(i)};\theta) \geq \sum_{i=1}^{N} \sum_{z^{(i)}} q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)};\theta)}{q_i(z^{(i)})}$$

- Then what *q_i*(z) should we choose?
- In order to make above inequality tight (to hold with equality), it is sufficient that

$$p(x^{(i)}, z^{(i)};\theta) = q_i(z^{(i)}) \cdot c$$

- We can derive

$$\log p(x^{(i)};\theta) = \log \sum_{z^{(i)}} p(x^{(i)}, z^{(i)};\theta) = \log \sum_{z^{(i)}} q(z^{(i)})c = \sum_{z^{(i)}} q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)};\theta)}{q_i(z^{(i)})}$$

- As such q_i(z) is written as the posterior distribution

$$q_i(z^{(i)}) = \frac{p(x^{(i)}, z^{(i)};\theta)}{\sum_z p(x^{(i)}, z;\theta)} = \frac{p(x^{(i)}, z^{(i)};\theta)}{p(x^{(i)};\theta)} = p(z^{(i)}|x^{(i)};\theta)$$

General EM Methods

Repeat until convergence: {

(E-step) For each *i*, set

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

(M-step) Update the parameters

$$\theta = \arg\max_{\theta} \sum_{i=1}^{N} \sum_{z^{(i)}} q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{q_i(z^{(i)})}$$

}

Convergence of EM

- Denote $\vartheta^{(t)}$ and $\vartheta^{(t+1)}$ as the parameters of two successive iterations of EM, we prove that

$$l(\theta^{(t)}) \leq l(\theta^{(t+1)})$$

which shows EM always monotonically improves the log-likelihood, thus ensures EM will at least converge to a local optimum.

Proof of EM Convergence

- Start from $\vartheta^{(t)}$, we choose the posterior of latent variable

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

- This choice ensures the Jensen's inequality holds with equality

$$l(\theta^{(t)}) = \sum_{i=1}^{N} \log \sum_{z^{(i)}} q_i^{(t)}(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{q_i^{(t)}(z^{(i)})} = \sum_{i=1}^{N} \sum_{z^{(i)}} q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{q_i^{(t)}(z^{(i)})}$$

- Then the parameters $\vartheta^{(t+1)}$ are then obtained by maximizing the right hand side of above equation
- Thus $l(\theta^{(t+1)}) \geq \sum_{i=1}^{N} \sum_{z^{(i)}} q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t+1)})}{q_i^{(t)}(z^{(i)})}$ [lower bound]

$$\geq \sum_{i=1}^{N} \sum_{z^{(i)}} q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{q_i^{(t)}(z^{(i)})}$$ [parameter optimization]

$$= l(\theta^{(t)})$$

Remark of EM Convergence

- If we define

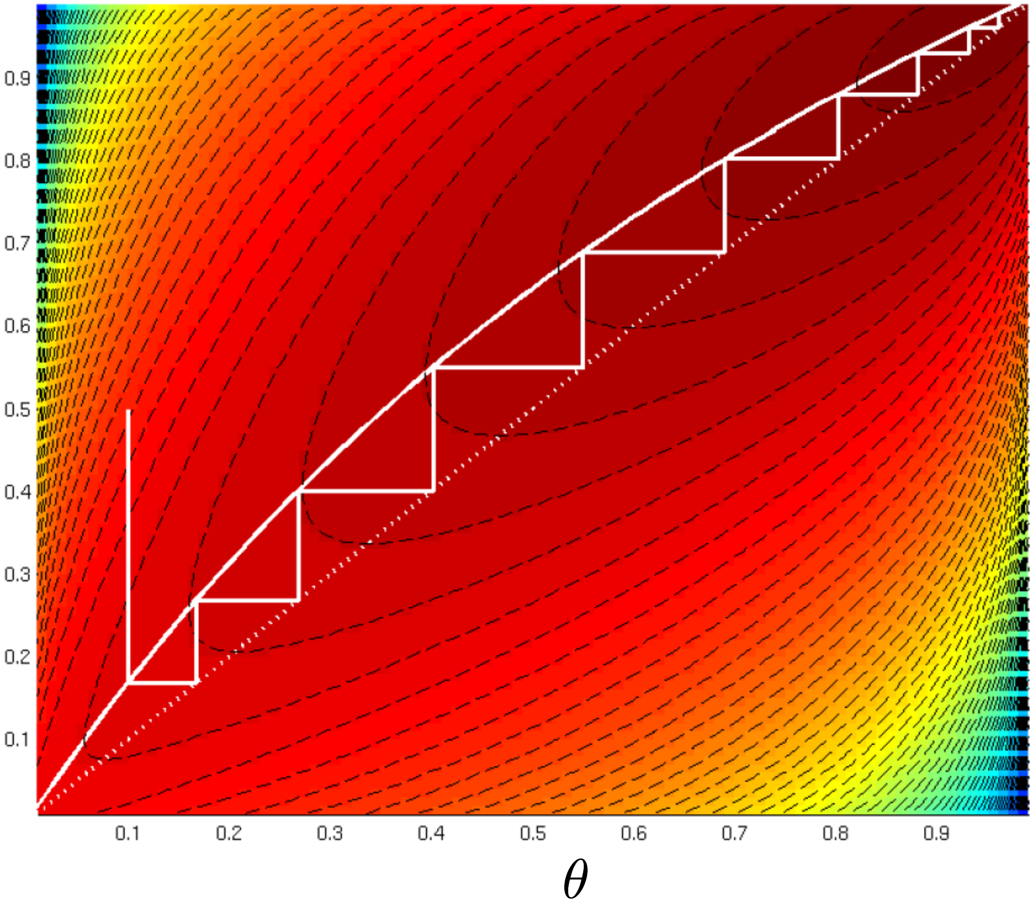
$$J(q,\theta) = \sum_{i=1}^{N} \sum_{z^{(i)}} q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{q_i(z^{(i)})}$$

  Then we know

$$l(\theta) \geq J(q,\theta)$$

- EM can also be viewed as a coordinate ascent on *J*
  - E-step maximizes it w.r.t. *q*
  - M-step maximizes it w.r.t. ϑ

Coordinate Ascent in EM



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

$$\theta^{(t+1)} = \arg\max_{\theta} \sum_{i=1}^{N} \sum_{z^{(i)}} q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{q_i^{(t)}(z^{(i)})}$$

Figure credit: Maneesh Sahani