

Week8

Table of Contents

Week8	
Unsupervised Learning introduction	
K-means Algorithms	
Optimization Objective	
Random initialization	
Choosing the Number of clusters	
Quiz: Unsupervised Learning	
PCA (Principal Component Analysis)	
Choosing the number of principal components	
Advice for applying PCA	
Quiz: Principle Component Analysis	

节选自《神经网络与深度学习》第九章：无监督学习

无监督学习（Unsupervised Learning, UL）是指从无标签的数据中学习处一些有用的模式。无监督学习算法一般直接从原始数据中学习，不借助任何人工给出标签或者反馈等指导信息。如果监督学习是建立在输入-输出之间的映射关系，那么无监督学习就是发现隐藏数据中的有价值信息，包括有效特征、类别、结构以及概率分布等。

典型的无监督学习分类：

- 无监督特征学习（Unsupervised Feature Learning）是从无标签的训练数据中挖掘有效的特征或表示。无监督特征学习一般用来进行降维、数据可视化或监督学习前期的数据预处理。
- 概率密度估计（Probabilistic Density Estimation）简称**密度估计**，是根据一组训练样本来估计样本空间的概率密度。密度估计可以分为参数密度估计和非参数密度估计。
 - 参数密度估计用假设数据服从某个已知概率密度函数形成的分布（如高斯分布），然后根据训练样本去估计概率密度函数的参数。非参数密度估计是不假设数据服从某个已知分布，只利用训练样本对密度进行估计，可以进行任意形状密度的估计。非参数密度估计的方法只有直方图。
 - 聚类（Clustering）是将一组样本根据一定的准则划分到不同的组（也称为簇（Cluster））。一个比较通用的准则是组内样本的相似性要高于组间样本的相似性。常见的聚类算法包括 K-Means 算法、谱聚类等。

和监督学习一样，无监督学习方法也包含三个基本要素：模型、学习准则和 优化算法。无监督学习的准则非常多，比如最大似然估计、最小重构错误等。在无 监督特征学习中，经常使用的准则为最

小化重构错误，同时也经常对特征进行一些约束，比如独立性、非负性或稀释性等.而在密度估计中，经常采用最大似然估计计来进行学习.

Unsupervised Learning introduction

Question

Which of the following statements are true? Check all that apply.

☒ In unsupervised learning, the training set is of the form $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ without labels $y^{(i)}$.

✓ Correct

☒ Clustering is an example of unsupervised learning.

✓ Correct

☒ In unsupervised learning, you are given an unlabeled dataset and are asked to find "structure" in the data.

✓ Correct

☐ Clustering is the only unsupervised learning algorithm.

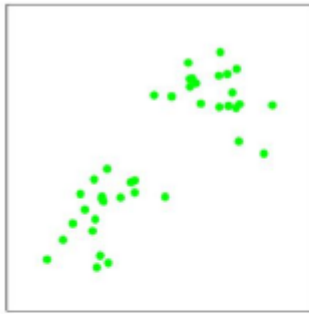
K-means Algorithms

Input:

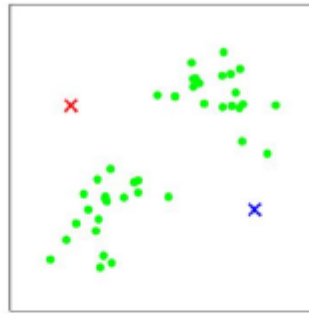
- K (number of clusters)
- Training set. $\{x^1, x^2, \dots, x^m\} x^i \in \mathbb{R}^n$ (Drop $x_0 = 1$ convention)

[K-Means聚类算法原理](#)

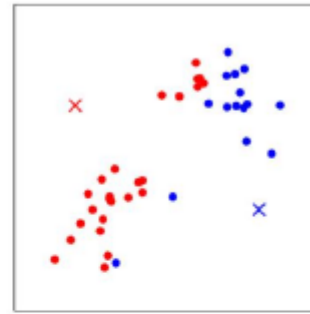
Goal: Minimize squared error



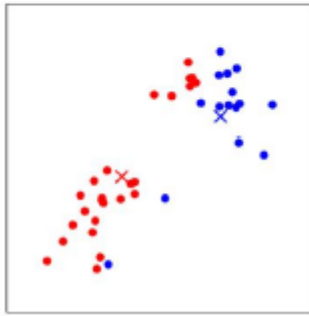
(a)



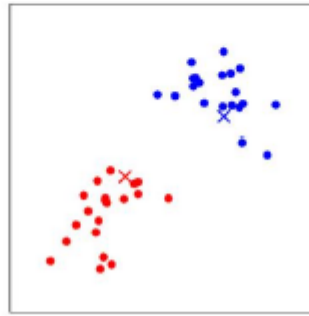
(b)



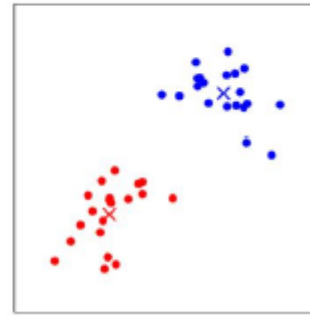
(c)



(d)



(e)



(f)

Repeat the following steps:

K-means algorithm

$$\mu_1 \quad \mu_2$$

Randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$

Repeat {

Cluster assignment step

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

$$\min_k \|x^{(i)} - \mu_k\|^2$$

Move centroid

for $k = 1$ to K

$\rightarrow \mu_k$:= average (mean) of points assigned to cluster k

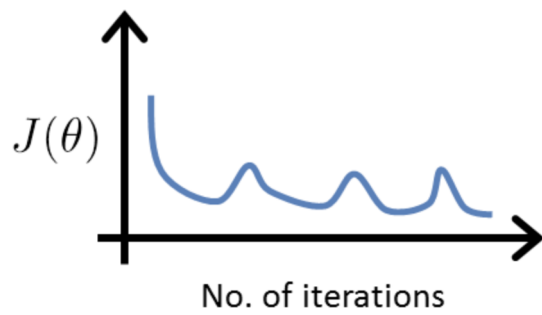
$$x^{(1)}, x^{(5)}, x^{(6)}, x^{(10)}$$

$$\rightarrow c^{(1)}=2, c^{(5)}=2, c^{(6)}=2, c^{(10)}=2$$

$$\mu_2 = \frac{1}{4} [x^{(1)} + x^{(5)} + x^{(6)} + x^{(10)}] \in \mathbb{R}^n$$

Optimization Objective

Suppose you have implemented k-means and to check that it is running correctly, you plot the cost function $J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_k)$ as a function of the number of iterations. Your plot looks like this:



What does this mean?

- ☐ The learning rate is too large.
- ☐ The algorithm is working correctly.
- ☐ The algorithm is working, but k is too large.
- ☒ It is not possible for the cost function to sometimes increase. There must be a bug in the code.

K-means optimization objective

- $c^{(i)}$ = index of cluster $(1, 2, \dots, K)$ to which example $x^{(i)}$ is currently assigned
 - μ_k = cluster centroid k ($\mu_k \in \mathbb{R}^n$)
 - $\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned
- Handwritten notes:* K , $k \in \{1, 2, \dots, K\}$, $x^{(i)} \rightarrow 5$, $c^{(i)} = 5$, $\mu_{c^{(i)}} = \mu_5$

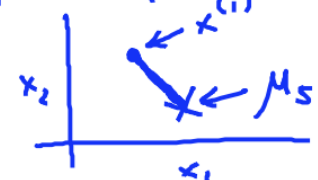
Optimization objective:

$$\rightarrow J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K) = \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

Handwritten notes: \leftarrow (pointing to the squared norm), \uparrow (pointing to $x^{(i)}$), \leftarrow (pointing to $\mu_{c^{(i)}}$)

$$\rightarrow \min_{c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K} J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$$

Handwritten notes: Distortion



Andrew N

K-means algorithm

Randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$

Repeat {

Handwritten notes: Cluster assignment step, Minimize $J(\dots)$ w.r.t $c^{(1)}, c^{(2)}, \dots, c^{(m)}$ (holding μ_1, \dots, μ_K fixed) \leftarrow

for $i = 1$ to m

$c^{(i)} :=$ index (from 1 to K) of cluster centroid closest to $x^{(i)}$

for $k = 1$ to K

$\mu_k :=$ average (mean) of points assigned to cluster k

}

Handwritten notes: minimize $J(\dots)$ w.r.t μ_1, \dots, μ_K

Handwritten note: move centroid

Random initialization

K-means algorithm

Randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$

Repeat {

 for $i = 1$ to m

$c^{(i)} :=$ index (from 1 to K) of cluster centroid
 closest to $x^{(i)}$

 for $k = 1$ to K

$\mu_k :=$ average (mean) of points assigned to cluster k

}

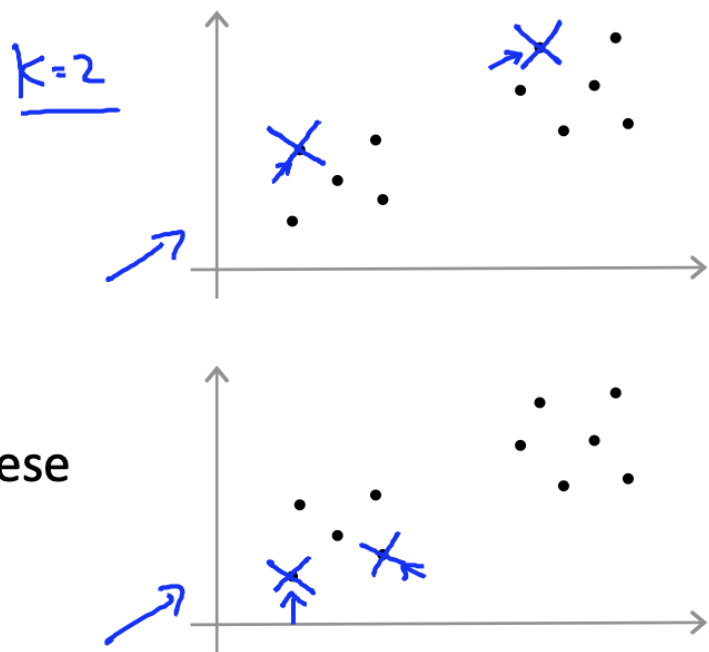
Random initialization

Should have $K < m$

Randomly pick K training examples.

Set μ_1, \dots, μ_K equal to these K examples.

$$\begin{aligned}\mu_1 &= x^{(i)} \\ \mu_2 &= x^{(j)} \\ &\vdots\end{aligned}$$



Done the whole procedure a hundred times. You will have a hundred different ways of clustering the data, pick one give the lowest cost.

Random initialization

For $i = 1$ to 100 {

 Randomly initialize K-means.

 Run K-means. Get $c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K$.

 Compute cost function (distortion)

$$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$$

}

Pick clustering that gave lowest cost $J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$

Which of the following is the recommended way to initialize k-means?

☐ Pick a random integer i from $\{1, \dots, k\}$. Set $\mu_1 = \mu_2 = \dots = \mu_k = x^{(i)}$.

☐ Pick k distinct random integers i_1, \dots, i_k from $\{1, \dots, k\}$.

Set $\mu_1 = x^{(i_1)}, \mu_2 = x^{(i_2)}, \dots, \mu_k = x^{(i_k)}$.

☒ Pick k distinct random integers i_1, \dots, i_k from $\{1, \dots, m\}$.

Set $\mu_1 = x^{(i_1)}, \mu_2 = x^{(i_2)}, \dots, \mu_k = x^{(i_k)}$.

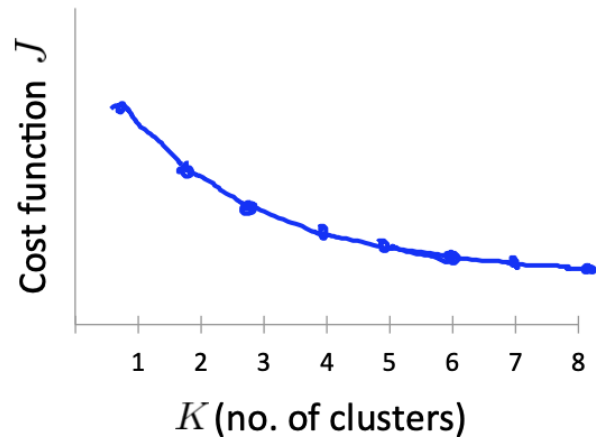
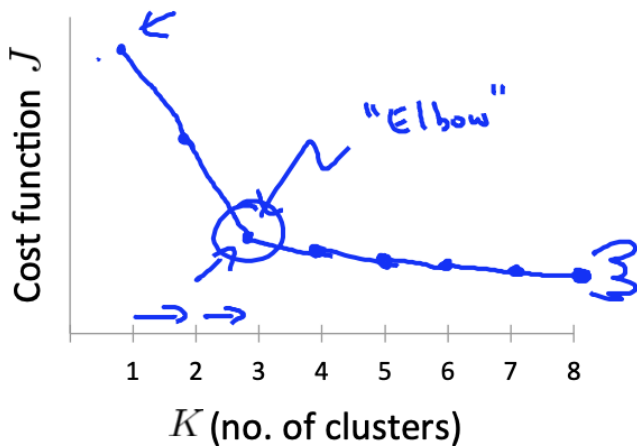
☐ Set every element of $\mu_i \in \mathbb{R}^n$ to a random value between $-\epsilon$ and ϵ , for some small ϵ .

 Correct

Choosing the Number of clusters

Choosing the value of K

Elbow method:



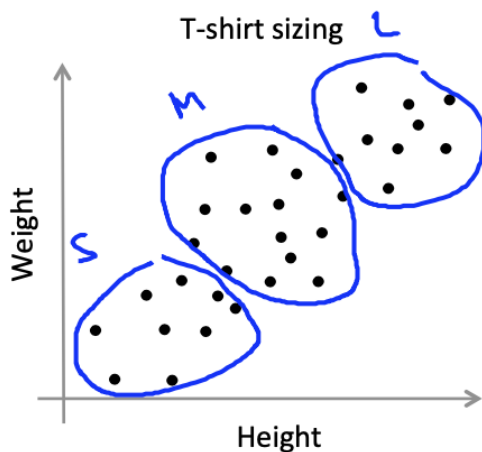
Andrew

Choosing the value of K

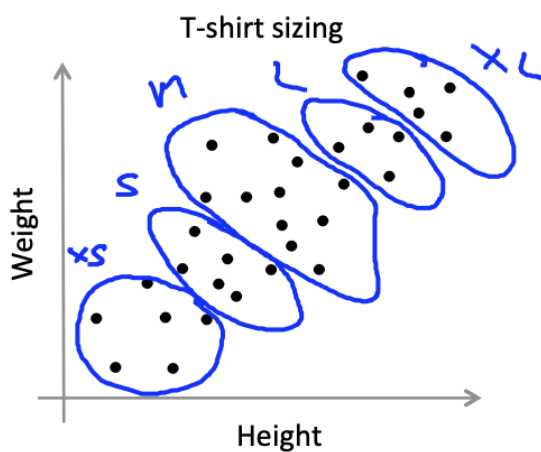
Sometimes, you're running K-means to get clusters to use for some later/downstream purpose. Evaluate K-means based on a metric for how well it performs for that later purpose.

$k=3$ S, M, L

E.g.



$k=5$ XS, S, M, L, XL



Suppose you run k-means using $k=3$ and $k=5$. You find that the cost function J is much higher for $k=5$ than for $k=3$. What can you conclude?

- ☐ This is mathematically impossible. There must be a bug in the code.
- ☐ The correct number of clusters is $k=3$.
- ☒ In the run with $k=5$, k-means got stuck in a bad local minimum. You should try re-running k-means with multiple random initializations.
- ☐ In the run with $k=3$, k-means got lucky. You should try re-running k-means with $k=3$ and different random initializations until it performs no better than with $k=5$.

Quiz: Unsupervised Learning

1. For which of the following tasks might K-means clustering be a suitable algorithm? Select all that apply.

1 / 1 point

- ☒ Given a database of information about your users, automatically group them into different market segments.



Correct

You can use K-means to cluster the database entries, and each cluster will correspond to a different market segment.

- ☒ Given sales data from a large number of products in a supermarket, figure out which products tend to form coherent groups (say are frequently purchased together) and thus should be put on the same shelf.



Correct

If you cluster the sales data with K-means, each cluster should correspond to coherent groups of items.

- ☐ Given historical weather records, predict the amount of rainfall tomorrow (this would be a real-valued output)

- ☐ Given sales data from a large number of products in a supermarket, estimate future sales for each of these products.

2. Suppose we have three cluster centroids $\mu_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} -3 \\ 0 \end{bmatrix}$ and $\mu_3 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$. Furthermore, we have a training example $x^{(i)} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$. After a cluster assignment step, what will $c^{(i)}$ be?

1 / 1 point

- ☒ $c^{(i)} = 1$
- ☐ $c^{(i)}$ is not assigned
- ☐ $c^{(i)} = 2$
- ☐ $c^{(i)} = 3$



Correct

$x^{(i)}$ is closest to μ_1 , so $c^{(i)} = 1$

3. K-means is an iterative algorithm, and two of the following steps are repeatedly carried out in its inner-loop. Which two?

1 / 1 point

- ☒ The cluster assignment step, where the parameters $c^{(i)}$ are updated.

✓ **Correct**

This is the correct first step of the K-means loop.

- ☐ Randomly initialize the cluster centroids.

- ☒ Move the cluster centroids, where the centroids μ_k are updated.

✓ **Correct**

The cluster update is the second step of the K-means loop.

- ☐ Test on the cross-validation set.

4. Suppose you have an unlabeled dataset $\{x^{(1)}, \dots, x^{(m)}\}$. You run K-means with 50 different random initializations, and obtain 50 different clusterings of the data. What is the recommended way for choosing which one of these 50 clusterings to use?

1 / 1 point

- ☒ Compute the distortion function $J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_k)$, and pick the one that minimizes this.
- ☐ Manually examine the clusterings, and pick the best one.
- ☐ Plot the data and the cluster centroids, and pick the clustering that gives the most "coherent" cluster centroids.
- ☐ Use the elbow method.

✓ **Correct**

A lower value for the distortion function implies a better clustering, so you should choose the clustering with the smallest value for the distortion function.

5. Which of the following statements are true? Select all that apply.

1 /

- ☒ If we are worried about K-means getting stuck in bad local optima, one way to ameliorate (reduce) this problem is if we try using multiple random initializations.



Correct

Since each run of K-means is independent, multiple runs can find different optima, and some should avoid bad local optima.

- ☐ The standard way of initializing K-means is setting $\mu_1 = \dots = \mu_k$ to be equal to a vector of zeros.
- ☐ Since K-Means is an unsupervised learning algorithm, it cannot overfit the data, and thus it is always better to have as large a number of clusters as is computationally feasible.
- ☒ For some datasets, the "right" or "correct" value of K (the number of clusters) can be ambiguous, and hard even for a human expert looking carefully at the data to decide.



Correct

In many datasets, different choices of K will give different clusterings which appear quite reasonable. With no labels on the data, we cannot say one is better than the other.

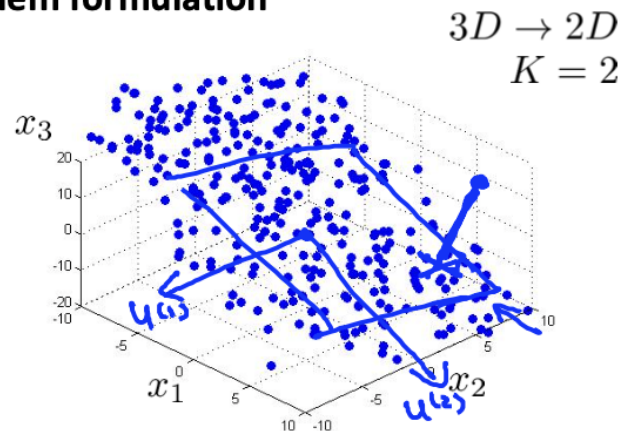
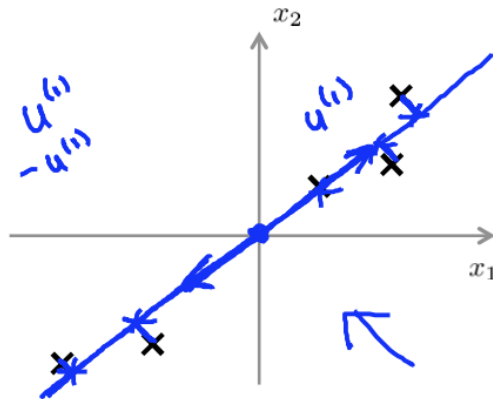
PCA (Principal Component Analysis)

主成分分析，是一种最常用的数据降维方法，使得在转换后的空间中数据的方差最大。

本章难度较大，涉及数学知识较广，参考相关文章。

[PCA主成分分析学习总结](#)

Principal Component Analysis (PCA) problem formulation



Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.

Reduce from n -dimension to k -dimension: Find k vectors $u^{(1)}, u^{(2)}, \dots, u^{(k)}$ onto which to project the data, so as to minimize the projection error.

Data preprocessing

Training set: $x^{(1)}, x^{(2)}, \dots, x^{(m)}$

Preprocessing (feature scaling/mean normalization):

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

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

If different features on different scales (e.g., x_1 = size of house, x_2 = number of bedrooms), scale features to have comparable range of values.

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

Principal Component Analysis (PCA) algorithm

Reduce data from n -dimensions to k -dimensions

Compute "covariance matrix":

$$\Sigma = \frac{1}{m} \sum_{i=1}^n \underbrace{(x^{(i)})}_{n \times 1} \underbrace{(x^{(i)})^T}_{1 \times n} \quad \text{--- } n \times n \quad \text{Sigma}$$

Compute "eigenvectors" of matrix Σ :

$$\rightarrow [U, S, V] = \text{svd}(\text{Sigma});$$

\rightarrow Singular value decomposition
 $\text{svd}(\text{Sigma})$

$n \times n$ matrix.

$$U = \begin{bmatrix} | & | & | & \dots & | \\ u^{(1)} & u^{(2)} & u^{(3)} & \dots & u^{(m)} \\ | & | & | & & | \\ \underbrace{\hspace{1.5cm}}_k & & & & \end{bmatrix}$$

$$U \in \mathbb{R}^{n \times n}$$

$$u^{(1)}, \dots, u^{(k)}$$

Principal Component Analysis (PCA) algorithm

From $[U, S, V] = \text{svd}(\text{Sigma})$, we get:

$$\rightarrow U = \begin{bmatrix} | & | & & | \\ u^{(1)} & u^{(2)} & \dots & u^{(n)} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times n}$$

$\underbrace{\hspace{10em}}_k$

$$x \in \mathbb{R}^n \rightarrow z \in \mathbb{R}^k$$

$$z^{(i)} = \underbrace{\begin{bmatrix} | & | & & | \\ u^{(1)} & u^{(2)} & \dots & u^{(k)} \\ | & | & & | \end{bmatrix}^T}_{n \times k} x^{(i)} = \underbrace{\begin{bmatrix} \text{---} (u^{(1)})^T \text{---} \\ \vdots \\ \text{---} (u^{(k)})^T \text{---} \end{bmatrix}}_{k \times n} \underbrace{x^{(i)}}_{n \times 1}$$

$z \in \mathbb{R}^k$ U_{reduce} $k \times 1$

Andrew I

Principal Component Analysis (PCA) algorithm summary

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

$$\text{Sigma} = \frac{1}{m} \sum_{i=1}^m (x^{(i)})(x^{(i)})^T$$

→ $[U, S, V] = \text{svd}(\text{Sigma})$;

→ $U_{\text{reduce}} = U(:, 1:k)$;

→ $z = U_{\text{reduce}}' * x$;

↑

↑

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

$$\cancel{x_0 = 1}$$

$$X = \begin{bmatrix} \text{---} x^{(1)T} \text{---} \\ \vdots \\ \text{---} x^{(m)T} \text{---} \end{bmatrix}$$

→ $\text{Sigma} = (1/m) * X' * X$

Choosing the number of principal components

Choosing k (number of principal components)

Average squared projection error: $\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2$

Total variation in the data: $\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2$

Typically, choose k to be smallest value so that

$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq \frac{0.01}{0.10} \quad \frac{(1\%)}{(10\%)}$$

→ "99% of variance is retained"
95% to 90%

Choosing k (number of principal components)

Algorithm:

Try PCA with $k=1$ $k=2$ $k=3$ $k=4$ \dots

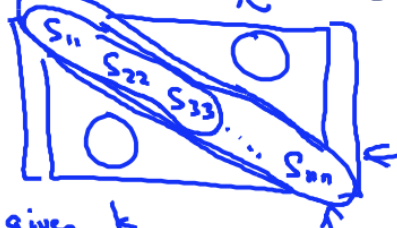
Compute $U_{reduce}, z^{(1)}, z^{(2)}, \dots, z^{(m)}, x_{approx}^{(1)}, \dots, x_{approx}^{(m)}$

Check if

$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq 0.01?$$

$k=17$

$$\rightarrow [U, S, V] = \text{svd}(\text{Sigma})$$

→ $S =$ 

For given k

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

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

Choosing k (number of principal components)

→ $[U, S, V] = \text{svd}(\text{Sigma})$

Pick smallest value of k for which

$$\frac{\sum_{i=1}^k S_{ii}}{\sum_{i=1}^m S_{ii}} \geq 0.99$$

$k=100$

(99% of variance retained)

Advice for applying PCA

Supervised learning speedup

→ $(\underline{x}^{(1)}, y^{(1)}), (\underline{x}^{(2)}, y^{(2)}), \dots, (\underline{x}^{(m)}, y^{(m)})$

Extract inputs:

Unlabeled dataset: $\underline{x}^{(1)}, \underline{x}^{(2)}, \dots, \underline{x}^{(m)} \in \mathbb{R}^{10000}$

↓ PCA

$\underline{z}^{(1)}, \underline{z}^{(2)}, \dots, \underline{z}^{(m)} \in \mathbb{R}^{1000}$

New training set:

$(\underline{z}^{(1)}, y^{(1)}), (\underline{z}^{(2)}, y^{(2)}), \dots, (\underline{z}^{(m)}, y^{(m)})$

Note: Mapping $x^{(i)} \rightarrow z^{(i)}$ should be defined by running PCA only on the training set. This mapping can be applied as well to the examples $x_{cv}^{(i)}$ and $x_{test}^{(i)}$ in the cross validation and test sets

$x^{(i)} \in \mathbb{R}^{10000} \leftarrow 100$



$x \rightarrow z$

$$h_{\theta}(z) = \frac{1}{1 + e^{-\theta^T z}}$$

$x \rightarrow z$

Application of PCA

- Compression

- Reduce memory/disk needed to store data
- Speed up learning algorithm ←

Choose k by % of variance retained

- Visualization

$k=2$ or $k=3$

Quiz: Principle Component Analysis

2. Which of the following is a reasonable way to select the number of principal components k ?

1 point

(Recall that n is the dimensionality of the input data and m is the number of input examples.)

- ☐ Choose k to be 99% of n (i.e., $k = 0.99 * n$, rounded to the nearest integer).
- ☐ Choose the value of k that minimizes the approximation error $\frac{1}{m} \sum_{i=1}^m ||x^{(i)} - x_{\text{approx}}^{(i)}||^2$.
- ☐ Choose k to be the smallest value so that at least 1% of the variance is retained.
- ☒ Choose k to be the smallest value so that at least 99% of the variance is retained.

3. Suppose someone tells you that they ran PCA in such a way that "95% of the variance was retained." What is an equivalent statement to this?

1 point

- ☐ $\frac{\frac{1}{m} \sum_{i=1}^m ||x^{(i)}||^2}{\frac{1}{m} \sum_{i=1}^m ||x^{(i)} - x_{\text{approx}}^{(i)}||^2} \geq 0.05$
- ☒ $\frac{\frac{1}{m} \sum_{i=1}^m ||x^{(i)} - x_{\text{approx}}^{(i)}||^2}{\frac{1}{m} \sum_{i=1}^m ||x^{(i)}||^2} \leq 0.05$
- ☐ $\frac{\frac{1}{m} \sum_{i=1}^m ||x^{(i)}||^2}{\frac{1}{m} \sum_{i=1}^m ||x^{(i)} - x_{\text{approx}}^{(i)}||^2} \leq 0.95$
- ☐ $\frac{\frac{1}{m} \sum_{i=1}^m ||x^{(i)}||^2}{\frac{1}{m} \sum_{i=1}^m ||x^{(i)} - x_{\text{approx}}^{(i)}||^2} \leq 0.05$

4. Which of the following statements are true? Check all that apply.

1 point

- ☐ PCA can be used only to reduce the dimensionality of data by 1 (such as 3D to 2D, or 2D to 1D).
- ☒ If the input features are on very different scales, it is a good idea to perform feature scaling before applying PCA.
- ☒ Given an input $x \in \mathbb{R}^n$, PCA compresses it to a lower-dimensional vector $z \in \mathbb{R}^k$.
- ☐ Feature scaling is not useful for PCA, since the eigenvector calculation (such as using Octave's `svd(Sigma)` routine) takes care of this automatically.

5. Which of the following are recommended applications of PCA? Select all that apply.

1 point

- ☐ Preventing overfitting: Reduce the number of features (in a supervised learning problem), so that there are fewer parameters to learn.
- ☐ To get more features to feed into a learning algorithm.
- ☒ Data compression: Reduce the dimension of your data, so that it takes up less memory / disk space.
- ☒ Data visualization: Reduce data to 2D (or 3D) so that it can be plotted.