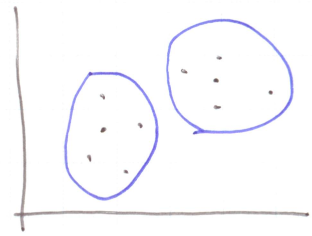
Week8 1

Clustering

Unsupervised learning

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



講義中のQuiz

① unsupervised learning では training set は $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ であり $y^{(i)}$ はない ○

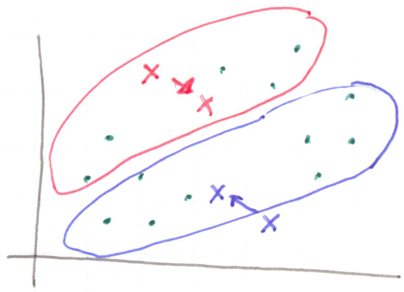
② Clustering は unsupervised learning の例だ ○

③ unsupervised learning では、unlabeled dataset が与えられるので、データの構造を発見するように依頼される ○

④ Clustering は unsupervised learning algorithm だけだ。 X

Week 8　　　2

K-Means アルゴリズム



2点の clustering center を置く

どちらに近いかで色分けする

それぞれのグループの中心に移動する

くり返し

Input:

K　クラスターの数

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

$x^{(i)} \in \mathbb{R}^n$　（$x^{(0)}=1$ を数えないのが慣習）

K個の clustering center $\mu_1, \mu_2, \dots, \mu_K$ をランダムに配置する

Repeat {

各点をクラスタに割りふる
　for $i=1$ to m
　　$c^{(i)}$ = $x^{(i)}$ に最も近い cluster center の index

クラスタ中心を動かす
　for $k=1$ to K
　　μ_k = クラスター k に割り当てられた点の平均

}

[Quiz] k-mean が収束した時 $c^{(1)}=3$, $c^{(2)}=3$, $c^{(3)}=5$, … となった。

① 3番目の example $x^{(3)}$ は クラスタ5に属する　○

② $x^{(1)}$ と $x^{(2)}$ は同じクラスタに割り当てられた　○ クラスタ3で同じ

③ 2番目と3番目の example が同じクラスタに割り当てられた　× クラスタ3と5で異なる

④ $k \in \{1, 2, \dots, K\}$ の全ての k について、$k=3$ が $\|x^{(2)} - \mu_k\|^2$ を最小化した　○ $x^{(2)}$ にとっては μ_3 が最も近い

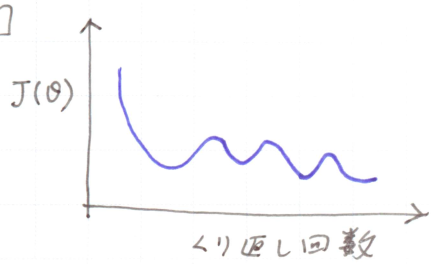
日本総研
The Japan Research Institute, Limited

Week 8　　3

Optimization Objectives

K-means optimization objectives

[Quiz]



① learning rate が大きすぎる　× learning rate など関係ない

② このアルゴリズムは正しく動作している　× 収束していない

③ このアルゴリズムは動作しているが K が大きすぎる　○ ~~μ_i が動き回っている。~~ ~~これは μ_i と μ_j が一点を取り合っていること~~

④ cost function が時々大きくなるのは無理である。コードのバグだ。 ~~×~~ ○

Random Initialization

```
for i=1 to 100{
    ランダムに K-means を初期化する
    c^(1), c^(2), ..., c^(m), μ1, μ2, ..., μK を得る
    cost function J(c^(1), ..., c^(m), μ1, ..., μK) を計算する
}
最も J が小さいものを選択する
```

[Quiz]　$i_1, \ldots, i_k$ を $\{1, \ldots, m\}$ の異なる値から選び　○

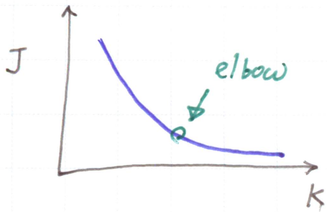
$\mu_1 = x^{(i_1)}, \ldots, \mu_K = x^{(i_K)}$ とする

Week 8　　4

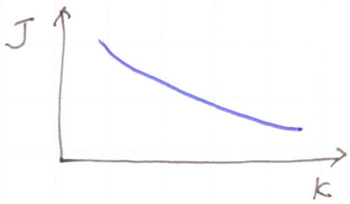
Choosing the Number of Clusters

Elbow method

Jの減少が緩かになったところが elbow



elbowがはっきりしない時もある。



[Quiz] K-means, k=3 と k=5　$J_{k=5} > J_{k=3}$ となった。理由は？

① 数学的にありえない。bug　×

② k=3が正しかった　×

③ k=5の時 local minimumにおちた。複数のrandom initializationで再try せよ　○

④ k=3の時がラッキーだった。k=3でもっと悪い結果のk=5のときと同じになるまで random initialization で再try せよ ×

クラスタの数は、分けた後どう使うかによって変わる。

例　T-shirt サイズ　　S, M, L → k=3

XS, S, M, L, XL → k=5

Week 8　　5

Motivation : Data Compression

Reduce data from 2D to 1D
3D　2D

[講義中のQuiz]
m個 データセット $\{x^{(1)}, \dots, x^{(m)}\}$　$x^{(i)} \in \mathbb{R}^n$
dimensionality reduction
↓
$\{z^{(1)}, \dots, z^{(m)}\}$　$z^{(i)} \in \mathbb{R}^k$ where $k \le n$　答

Motivation : Visualization

[講義中のQuiz]
データセット $\{x^{(1)}, \dots, x^{(m)}\}$ where $x^{(i)} \in \mathbb{R}^n$

$\{z^{(1)}, \dots, z^{(m)}\}$ where $z^{(i)} \in \mathbb{R}^k$, $k \le n$

① $k > n$ ×

② $k \le n$ ○

③ $k \ge 4$ × visualize しにくい

④ $k = 2$ or 3 ○　2次元か3次元なら視覚化できる

日本総研 The Japan Research Institute, Limited

Week 8 6

Principal Component Analysis (主成分分析)

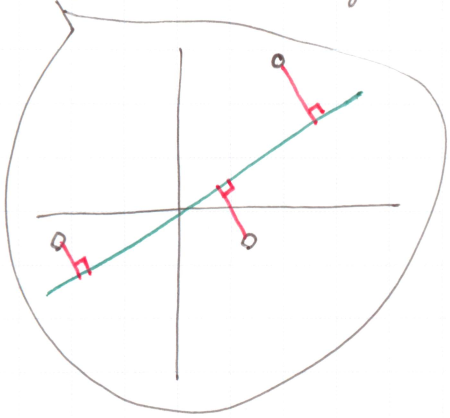
2次元から1次元へのreduction

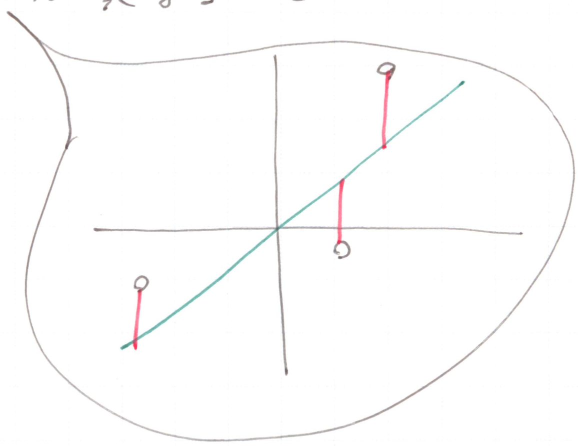
projection error が最小になる方向ベクタ $u^{(1)} \in \mathbb{R}^n$ を見つける
(射影誤差)

n次元から k次元への reduction

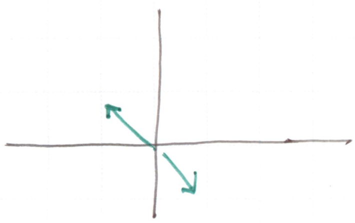
$u^{(1)}, u^{(2)}, \ldots, u^{(k)}$ where $u^{(i)} \in \mathbb{R}^n$

PCA と linear regression は異なることに注意





[講義中のQuiz]



単位ベクトルなので

$u^{(1)} = \frac{1}{\sqrt{2}}(1, -1)$ or $\frac{1}{\sqrt{2}}(-1, 1)$

Principal Component Analysis Algorithm

Data Processing

training set $\{x^{(1)}, \dots, x^{(m)}\}$

前処理 (feature scaling / mean normalization)

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

Replace: 各 $x_j^{(i)}$ を $\boxed{x_j^{(i)} - \mu_j}$ で ← 中心を原点にもってくる

feature によって scale が異なる時は 比較可能な範囲になるようにスケールする

$x_j^{(i)}$ を $\boxed{\dfrac{x_j^{(i)} - \mu_j}{s_j}}$ で置き替える

$$s_j = \max_i (x_j^{(i)}) - \min_i (x_j^{(i)})$$

PCAを探すアルゴリズム

1. n次元をk次元に落とす (reduce)
2. covariance matrix (共分散行列) Σ を求める

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

$$\underset{n \times m}{X^T} \; \underset{m \times n}{X} \to n \times n$$

3. Σ 行列の eigenvectors (固有ベクトル) を求める

$$[U, S, V] = \text{svd}(\underbrace{\text{Sigma}}_{n \times n})$$

svd ← singular value decomposition (特異値分解)

(注) eig(Sigma) でも計算できる

$$U = \Big(\underbrace{u^{(1)} \; u^{(2)}}_{k\text{個}} \dots u^{(m)}\Big), \quad U \in \mathbb{R}^{n \times n}, \quad u^{(1)}, \dots, u^{(k)} \in \mathbb{R}^n$$

最初から k個のベクトル $u^{(1)}, \dots, u^{(k)}$ を取り出す

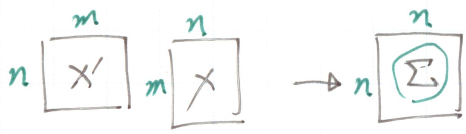
$$x \in \mathbb{R}^n \to z \in \mathbb{R}^k$$

$$z = \underbrace{\begin{pmatrix} u^{(1)} & \dots & u^{(k)} \end{pmatrix}}_{n \times k \text{ 行列 } U_{reduce}}{}^{T} \times \underset{n}{\begin{bmatrix} x \end{bmatrix}} \to \underset{k}{\begin{bmatrix} \; \end{bmatrix}}$$

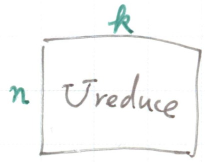
Videoの説明が悪い。
説明と実装で行と列が逆
↑
混乱の原因

Week 8　　　8

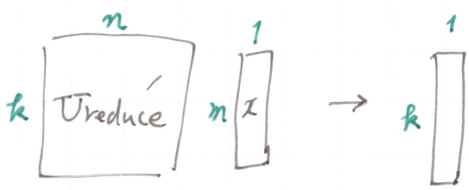
$$X = \begin{array}{cc} & n \\ m & \left(\begin{array}{c} — x^{(1)T} — \\ \vdots \\ — x^{(m)T} — \end{array}\right) \end{array}$$ と表現すると $\text{Sigma} = (1/m) * X' * X$



$U_{reduce} = U(\underset{\text{全ての行}}{:}, \underbrace{1:k}_{1\sim k列});$



$z = U_{reduce}' * x$



[講義中の Quiz]

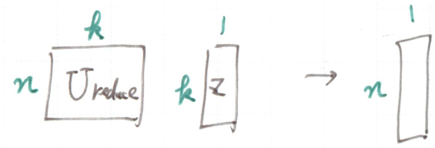
PCAにより $x \in \mathbb{R}^n$ から $z \in \mathbb{R}^k$ を得る.

$$z = \begin{pmatrix} z_1 \\ \vdots \\ z_k \end{pmatrix} = \begin{pmatrix} — u^{(1)T} — \\ \vdots \\ — u^{(k)T} — \end{pmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$\therefore \underline{z_j = u^{(j)T} \cdot x}$ (答)

compress されたデータを元の次元に戻す

$$X_{approx} = U_{reduce} \cdot z$$



[講義中のQuiz]

次元 reduction なしの $k=n$ で PCA を実行した場合、
variance の percent/fraction は $\dfrac{\sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{n} S_{ii}}$ で与えられる。

正しいのはどれか?

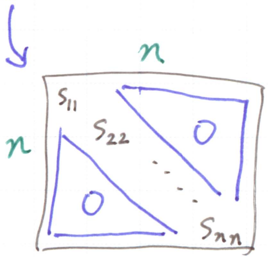
① U_{reduce} は $n \times n$ 行列 ○

② $x_{approx} = x$ が全ての x で成り立つ ○ Uは同じ次元の別の座標系での座標なので元に戻せる

③ percentage of variance $= 100\%$ ○ ∵ $k=n$ なので分母と分子が同じ

④ $\dfrac{\sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{n} S_{ii}} > 1$ × ∵ 1 だから

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



S_{ii} は固有値か? (eigen value)

PCA componentの数kを選ぶ

kは次の条件を満たす中で最小の値

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

← 2乗誤差の和

← varianceの2乗和

"99% of variance is retained"
「99%の分散が保持されている」 ← 他人に説明する時は、この表現を使う

kを求める方法

$k=1$で PCA を実行して ①が満たされているかをcheckする。

↓

kを増やしていく

①の左辺 $= 1-\frac{\sum_{i=1}^{k}S_{ii}}{\sum_{i=1}^{n}S_{ii}}$ である。∴①は $\frac{\sum_{i=1}^{k}S_{ii}}{\sum_{i=1}^{n}S_{ii}} \geqq 0.99$ と同じ

[講義中のQuiz]

PCAは projection error を小さくするように $u^{(1)}$ を選ぶこと。

同じ意味の式は

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

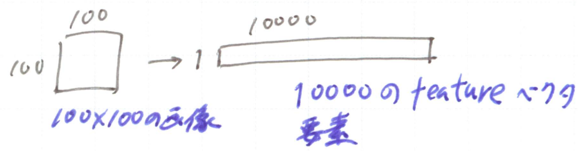
Week 8 11

Advice for Applying PCA

featureの数を減らすのに使える.

Supervised learning speedup

$(x^{(1)}, y^{(1)}), \ldots (x^{(m)}, y^{(m)})$, wher $x^{(i)} \in \mathbb{R}^{10000}$



unlabeled dataset $\{x^{(1)}, \ldots, x^{(m)}\}$ where $x^{(i)} \in \mathbb{R}^{10000}$

↓PCA

$\{z^{(1)}, \ldots, z^{(m)}\}$ where $z^{(i)} \in \mathbb{R}^{1000}$

新しく得られた
training set: $(z^{(1)}, y^{(1)}), \ldots, (z^{(m)}, y^{(m)})$

[Note] $x^{(i)} \to z^{(i)}$ へのmappingは training setのみに対して実行したPCAで定義されるべき。
そのmappingをcross validationやtestの時も使うべき。

PCAを適用する利点.

Compression
- メモリやディスクスペースの節約
- 学習のspeed up

Visualization
$k=2$ or 3 ならば視覚化しやすい

注意 overfittingを直すのには使えない (featuresを減らせるといって).
本質的なfeatureは(ほとんど)減っていないので

overfittingを直すには regularizationの入った を使うべき

注意 最初はPCAを使わずにやってみる。それで駄目ならPCAを適用してみる

Week 8　12

[~~講義~~中の Quiz]

PCA を使うのに適切なのは?

① メモリやディスクの節約のために compression する ○

② 学習を speed up するためにデータの次元を減らす ○

③ regularization の代わりに、feature を減らして overfit を直す ×

④ k=2 か 3 にして visualizing する ○