

Unsupervised Learning:Introduction

- 教師あり学習 データセットにはデータを示す x とそのデータのラベルを示す y があり、新しいデータがどのラベルに属しているかを問われる。
- 教師なし学習 データセットにはデータを示す x しかなく、データのラベルを示す y はない。与えられたデータセット内のデータ構造を調べるために使われる。
。マーケティングのセグメントを見つけたり、SNS分析で人をグルーピングするのに用いられる。

K-Means algorithm

K-Meansとは

データセットを任意の数のグループに分類する手法である。

1. 分類するクラスターの数(k)を決める。
2. k 個の重心の座標を決める
3. 重心とのユークリッド距離をすべて計算し、すべてのデータセットに対してどの重心が一番近いか求める
4. クラスタリングしたデータセットの中でそれぞれの平均を求め、新たな重心を決める
5. 重心が動かなくなるまで3.と4.を繰り返す

もしクラスタリングの途中でデータが振り分けられない重心が存在したらその重心を取り除き、 $k-1$ 個のクラスターに分けるようにするのが普通。

Optimization Objective

以下のように変数をきめる

- $c^{(i)}$: それぞれのデータセットが属するクラスターの番号
- μ_k : k 番目の重心
- $\mu_{c^{(i)}}$: i 番目のデータセットが属するクラスターの重心

このときK-Meansのコスト関数は次の通りとなる。

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

$$\min_{\theta} C[\sum_{i=1}^m y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)})] + \frac{1}{2} \sum_{j=1}^n \theta_j^2$$

Random Initialization

最初の k 個の重心の座標の決め方

最初の座標の決め方によっては、誤ったクラスタリングをしてしまうがあるので以下の初期化を実行することで正しいクラスタリングをする。

1. データセットの中から k (クラスターの数)個の標本を選び、それを最初のクラスターの重心に設定する
2. k-Meansを実行し目的関数を最小化させる
3. 1.2.を何度も(100回くらい?)実行させコスト関数がグローバル最小となる値を見つける。

Choosing the Number of Clusters

クラスターの数を最適化する絶対的な手法は存在しない。1つの方法はエルボ一法という方法である。

縦軸にK-Meansのコスト関数、横軸にクラスターの数を設定しコスト関数をプロットする。コスト関数の傾きが大きく変化する点があればそれが最適なクラスターの数となる。

しかし、急激な傾きの変化なくコスト関数が最小化するとき、エルボ一法は使

えない。何のためにクラスタリングをしているのかという目的から、クラスターの数を決めるこの法が多い。Tシャツのサイズの数を決めるために人の身体データをクラスタリングするときのように。

Motivation:date compression

実際の開発においてデータの次元はとても多くなるが、次元が多くなるにつれて計算は遅くなる。従ってよりよいアルゴリズムにするためには次数削減が必要である。それぞれの変数の定義から不必要なものは他の変数で代用することが必要である。また、2つの変数をプロットするとある直線上にのる、もしくは3つの変数をプロットするとある平面上になる場合には射影を用いることで2次元を1次元に、または3次元を2次元にという次数削減が可能となる。

Motivation: Visualization

多くの次元をもつデータセットから学習させるアルゴリズムを作るためにはデータを可視化できるほうがよい。たとえデータの次元が50とかあったとしても自分が分析したい内容から考えて必要な2,3の変数を選び、それらの値でデータセットを代表させることでデータを可視化できる。

Principal Component Analysis Problem Formulation

PCA(主成分分析)の概要

よりよい次元削減を考える。具体的にはn次元のデータをk次元のデータに次元削減することになるがわかりやすいように2次元で考える。

x_1, x_2 からなる座標平面上でデータをプロットしたとき、データはある直線に射影する形で1次元の値で代表される。その直線の選び方はその直線と各データとの距離(射影誤差)の総和が最小になるときでなければならない。これは3次元→2次元の次元削減でも同様で、射影誤差の総和が小さくなるように平面(平面を表す2つのベクトル)を導かなくてはならない。

線形回帰でも同じようなことをしたが、2次元の例で違いを述べると線形回帰の場合、 x, y からなる平面上で全てのデータとy方向に平行な差の総和が最小になる直線を選ばなくてはならなかった。

従って線形回帰とPCAでは誤差の定義が異なる。

Principal Component Analysis Algorithm

PCAのアルゴリズム実装の前にデータを平均標準化(mean normalization)[データの平均を0にする]する必要がある。

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

を計算し、 $x_j^{(i)}$ を $x_j - \mu_j$ で置き換える必要がある。

さらに必要であればそれぞれの変数を比較できるようにfeature scalingもしなくてはならない。

PCAのアルゴリズム n次元からk次元に次元削減する場合、共分散行列(Σ)の特異値分解をする必要がある。

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

共分散行列は上の式で計算される。

$x^{(i)} \in \mathbb{R}^{n \times 1}$ なので $\Sigma \in \mathbb{R}^{n \times n}$

ここから Σ の固有値ベクトルを求める(svdでもeigでもどちらでもよい)。octaveで固有値ベクトルを求めるコードは次の通り

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

ここでUは固有値ベクトルを表し、 $U \in \mathbb{R}^{n \times n}$ を表す。この固有値ベクトル行列の最初のk列を取り出す。 $(U, reduce$ とする)

k次元に次元削減したデータセット $z \in \mathbb{R}^k$ を得るために $U, reduce$ の転置行列

と(n次元のデータセット)xをかけて上げればよい。

$$(x \in \mathbb{R}^n, z \in \mathbb{R}^k, U_{reduce} \in n \times k)$$

PCAをoctaveで実装したコードは以下の通り

```
Ureduce = U(:,1:k);
z = Ureduce' * x;
```

Reconstruction from Compressed Regression

データの再構築

上の次元圧縮で示した例において、圧縮されたデータセットzから圧縮する前のデータセットxを求めるを考える。 $z = U_{reduce}^T x$ なので、xは $U_{reduce} x$ で近似できる。しかし、 $k=n$ でない限りに二乗射影誤差が含まれてしまうため全く同じ値にはならない。

Choosing the Number of Principal Components

kの選び方

$$\text{二乗射影誤差: } \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2$$

$$\text{全データの分散: } \frac{1}{m} \|x^{(i)}\|^2$$

kは以下の不等式を満たさなくてはならない

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

$$\theta \min \sum_{j=1}^n \theta_j^2$$

この条件をkが満たしているとき「この主成分分析は99%の分散を保持している」と言える。

実際の計算

共分散行列の特異値分解したときのsvdコマンドで得られるSという行列は対角成分以外は0の行列であり、以下の式を満たす

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

従って99%の分散を保持した主成分分析を行うためには
 $\frac{\sum_{i=1}^k S_{ii}}{\sum_{i=1}^n S_{ii}} \leq 0.99\sigma^2$ を満たす最小のkを見つければよい

Advise for Applying PCA

主成分分析は教師あり問題にも用いることができる。与えられたデータセットの中で入力項xの次元を主成分分析を用いて削減してからニューラルネットワークや論理回帰のアルゴリズムを適用させることができる。しかし、主成分分析はトレーニングセットにのみ用いるべきで、クロスバリデーションセットやテストセットに用いてはいけない。

主成分分析を使うとき

- 使って良い時 機械学習のプログラムのスピードを速くしたい時 メモリや記憶容量の使用量を減らしたい時 可視化したいとき
- 使うのがあまりよくないとき アルゴリズムがoverfittingしているとき 使うのを前提としている時

- 機械学習のアルゴリズムを計画するときPCAを使ってしまう場合があるが、できる限り生のデータでアルゴリズムを実装し、それはうまくいかないとき、もしくは計算を早めたり、メモリの使用量を減らしたい時にPCAを使うべきである。