2019年8月27日 火曜日

教師なし学習に関するノート

主成分分析（PCA）

1．主成分分析の目的

多変量(高次元)データをできるだけ情報を損なわず低次元空間に表現したい。表現したいモチベーションは例えば次の通り。

・高次元のデータを２次元平面や３次元空間にマッピングして可視化したい
・データの容量を小さくしたい
などなど

どんな情報を損なわずに主成分分析を行いたいかで手法が変わってくる。

2. フロベニウスノルムを最小化する方法

データ行列$X:\{x_{ij}\}_{n\times p}$に対して、目的として次のような式を考え、これを満たすようなYを求めることにする。

$$\min_{rank(Y)\leq q} ||X-Y||_F^2$$

ここで、フロベニウスノルムは、中の行列の各要素の二乗和。

この最適化問題を解くにあたっては、次のEckart-Youngの定理を利用する。

Eckart-Youngの定理

Xの特異値分解$X=U\Delta V^T$を考える。ただしΔはσ_jを対角成分とする対角行列、対角成分は行列の特異値。U,Vの要素ベクトルはu_i,v_jで表す。このとき、$rank(Y)\leq q$を満たす全てのYに対して、以下が成り立つ。

$$||X-Y||_F^2 \geq ||X-X_q||_F^2$$

ただし、$X_q=U_q\Delta V_q=\sum_{j=1}^{q}\sigma_j u_j v_j^T$

つまり、この定理で言うところのX_qを求めてやれば良いと言うことになる。

例　画像データの次元削減

$q = rank(X) - p$とし、pを5,10,...20と動かした。元の画像データは画像１で、これをグレースケールに変換して利用した。

画像１

動かした際に出力された画像は次の画像3,4。

画像3,4

かなり早い段階で元の画像に近い状態になっていることがわかる。
一連のコードはFrobeniusPcm.Rに保存した。

3.情報損失最小化する方法

次のような目的に従う未知行列Aを用いて、XAA^Tを考えたい。

$$\min_A ||X - XAA^T||_F^2$$

$\min_A ||X - XAA^T||_F^2 = ||X||_F^2 - tr(A^TX^TXA)$であることに注目すると、$tr(A^TX^TXA)$を最大化する$A$を考えれば良いことに気づく。$tr(A^TX^TXA)$はデータ$XA$に関する標本分散共分散行列の推定量をn倍したものであると解釈すると、次元削減後のデータに関する分散最大化問題と考えることもできる。つまり、次のように考える。次元削減により取り出した主成分をXAA^T、主成分以外のデータの残滓を$X - XAA^T$とみる。

例　　Bostonデータ(rank=13)の主成分を２次元取り出し、biplotする。
一連のコードはBiplot.Rに保存した。

例　　動画の主成分(ほとんどのフレームでは写っているもの)を取り出すこともできる。一連のコードはmovie.Rに保存した。

４. カーネル主成分分析

　　当然、カーネルを使って主成分分析を非線形に拡張することもできる。コードはカーネルPCA.Rに示した。

多様体学習

1. 多様体学習の目的

　　高次元空間上のデータなんだけど、実は多様体上に乗っかっただけのデータで、実は低次元データであるような場合がある。適当な多様体上に乗ったデータの場合、普通のユークリッド距離を考えると、本来の性質をうまく反映させることができない。当然そのようなデータの場合、PCAではうまくいかない。そこで、d次元多様体上のデータであると仮定し、そのデータをうまく多様体上の構造を残したまま、d次元ユークリッド空間に表現し直したいみたいなモチベが発生する。

2. 測地線

測地線は、多様体上の直線のことをさす。多様体上しか進めないと仮定した場合の最短距離が二点間の測地線に該当する。

測地線を推定するためには、近傍グラフ上の最短経路を計算する必要がある。と言うのも、適当な正則条件のもとで、近傍グラフ上の最短経路は$n \to \infty$で測地線に収束すると言うことが知られている。

近傍には次のものがある。

- ε 近傍グラフ：ε を与えた上で、イプシロンボールを考える。イプシロンボールの中心と、その内部の点の間に辺を与える。
- k近傍グラフ：近傍内に含まれる点の個数kを事前に与えた上で、近傍内に含まれる点に対して辺を与える。

2019年8月27日 火曜日

３. 測地線の推定

1　まず近傍を定義し、辺を与える。
2　そして辺の重みをデータ間のユークリッド距離で与える
3　最後に、近傍グラフ上の最短経路で測地線距離を推定する。

例　　Rにはisomapメソッドがあるが、いいデータがなかったため、参考コードだけまとめて省略(isomap.R)

クラスタリング

1. クラスタリング概要

古典的なクラスタリングには階層、非階層といった手法が存在する。階層クラスタリングは逐次的にクラスタ数を変えていく方法でRのhclustなどが該当する。hclustの”ward”はウォード法ではないと言うことが指摘されているので使うならward.D2の方が良い。k-meansは非階層クラスタリングに該当する。当然カーネルを乗せてカーネルk-meansを考えることもできるが、Rに既存の関数は現状ない(kkmeansはカーネルk-meansではなく、normalized cutであったりする)

2. スペクトラルクラスタリング

データに対して、グラフを考えて、いい塩梅になるようcutすることでクラスタリングを行う方法。

いい塩梅になるようcut といえばGraph cutが考えられるが、、、

3. Graph cut

Graph cutとは様々考えられるが、例えば
・クラスタ間のコネクションが少ない
・クラスタのサイズ(クラスタ内のノードの個数)がある程度大きくなる
ようなcutがよく使われる。

まず、データの隣接行列Kの要素を$k_{i,j}$とする。

Ratio cut

Ratio cut は次のような目的関数を与えるcutである。ただし、クラスタ数2(A,Bと言う排反のクラスタに分けられる)の場合を考える。クラスタ数が３以上の場合でも、クラスタAとそれ以外に分ければ、クラスタ数２のアルゴリズムと変わらない。

$$Rcut(A,B) = Mcut(A,B)\{\frac{1}{\#(A)} + \frac{1}{\#(B)}\}$$

ただし$Mcut(A,B) = \sum_{i\in A}\sum_{j\in B} k_{i,j}$

$\#(A)$ = {Aにふくまれる頂点数}

Normalized cut

Normalized cutでは、Ratio cutで用いた頂点数をクラスタ内の総degreeに置き換えることで目的関数が得られる。

ただし、これらのcutはNP困難なので最適化が難しい。そこでスペクトラルクラスタリングが利用される。

このcutのideaをうまく活用して、スペクトラルクラスタリングを行う。

４.Unnormalized スペクトラルクラスタリング

まず、Ratio cutを一般化すると次のようにかける。
クラスタの集合$\{C_1,\dots C_M\}$を考える。この時、Ratio cutの目的関数は次のようになる。

$$Rcut(C) = \sum_{m=1}^{M} \frac{vol(C_m) - Mcut(C_m, C_m)}{\#(C_m)} = tr\{\hat{U}^T(D-K)\hat{U}\}$$

ただし、
$vol(C_m)$ = クラスタmに含まれるノードのdegreeの総和
D = 各ノードのdegreeを対角成分とした対角行列
Uはn(頂点数)×M(クラスタ数)行列であり、頂点のクラスタに対する帰属を0,1で示す行列。
$\hat{U} = U(U^TU)^{-\frac{1}{2}}$

ここで、$L = D - K$は重要で、Graph LaplacianとかUnnormalized Laplacianと呼ばれる。以上より結局、

$$tr\{\hat{U}^T L \hat{U}\}$$

を最小化すれば良いと言うことになる。離散だと組み合わせが爆発するので、**連続最適化にrelaxした**ものをUnnormalized スペクトラルクラスタリングと呼ぶ。

グラフラプラシアンの重要な性質
非負重み付きグラフを考えた場合、Null spaceは0,1を要素とするM個の固有ベクトルで張られ、各固有ベクトルが各クラスタに属する要素に対応する。

クラスタリング方法
よって次のような方法でクラスタリングを行う
1　グラフラプラシアンLを計算する
2　Lの最小のM個の固有値に対応する固有ベクトルを列荷物行列Vを計算する
3　Vに対してk-meansを適用する

5.Normalized スペクトラルクラスタリング

Normalized cutを一般化すると目的関数は次のようになる。

$$Nvut(C_1, \dots C_M) = M - \sum_{m=1}^{M} \frac{Mcut(C_m, C_m)}{vol(C_m)} = tr\{\hat{U}^T D^{-\frac{1}{2}} K D^{-\frac{1}{2}} \hat{U}\}$$

先ほどと同様に連続化し、次のような問題を考える

$$\min_{\hat{U}} tr\{\hat{U}^T D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \hat{U}\} \qquad s.t.\ \hat{U}^T \hat{U} = I_M$$

ただしLはUnnormalized ラプラシアン

他に重要な考え方として、Normalized ラプラシアンL_{norm}と言うものがある。

$$L_{norm} = D^{-\frac{1}{2}} L D^{\frac{1}{2}}$$

クラスタリング方法1
1　　グラフラプラシアンLを計算する
2　　一般固有値問題$Lv = \lambda D v$に対して(最小一般化固有値から数えて)最初のM個の一般化固有ベクトルを求める
3　　Vをそれらを列に持つ行列とする
4　　Vに対してk-meansを適用する

クラスタリング方法２

1　normalized laplacian L_{norm}を計算する

2　L_{norm}の最小のM個の固有値に対応する固有ベクトルを列に持つ行列Vを計算する

3　Vの各行のノルムを１に標準化する

4　求めた行列に対してk-meansを適用する

６.スペクトラルクラスタリングまとめ

応用上は、無向化したk近傍グラフを作成して、Normalized Laplacianを用いるのが無難。k近傍グラフを作成するにあたってのkはk>log(n)が良い。Rのkernlabにはspectral.clustering関数があるが、これはUnnormalized スペクトラルクラスタリングであることに注意。normalizedスペクトラルクラスタリングはpythonで行う。(SpectralClustering.py)

７.Reduced k-means Clustering

クラスタリングは、探索的、教師なしといった特徴があるため、クラスタリングを行なった後に、クラスタの特徴を知ることが重要となる。その際、高次元データのクラスタリングでは、その特徴がとらえにくい、二次元プロットができず、可視化が行えないといった欠点がある。そこで、主成分分析を適用した上で、データを低次元空間に縮約した上でクラスタリングを行うと言う方法(Tandem clustering)が考えられるが、これはあまりうまくない。何故なら、最初の主成分がクラスタ構造を反映するとは限らないからである。そこで、reduced k-means clusteringと言う手法が用いられる。ちなみにクラスタ数kと次元数qが等しい場合、k-meansと同値になる。Rでは簡単に実行できるメソッドがある(RKM.R参照)。

８.クラスタ数や次元数等のパラメータ決定法

Ben-David et.al,2006(https://cs.uwaterloo.ca/~shai/sober.pdf)では、クラスタリング距離という概念が提案されている。これは、２クラスタの不同意確率であり、クラスタ数が異なる場合も評価可能で、しかも距離の公理を満たすという点で使い勝手が良い。

クラスタリング距離

データ$X_{(n)} = (X_1, \ldots X_n)$について、２つのクラスタリング$c_1, c_2$を考える。ここで、次のような２つのクラスタリングに対する距離$d_C(c_1, c_2)$を次のように定義する。

$$d_C(c_1, c_2) = Prob\{1_{(c_1(X)=c_1(Y))} + 1_{(c_2(X)=c_2(Y))} = 1\}$$

これは、クラスタリングc_1, c_2を用いて独立の二種類のデータセットX, Yをクラスタリングした場合に、c_1では同一クラスタにクラスタリングされるにも関わらず、c_2では異なるクラスタにクラスタリングされている(又はその逆)の場合の確率(２クラスタの不同意確率)を２クラスタリングの距離として定式化したものである。二種類のクラスタリング方法が両方異なるクラスタにクラスタリングしていたり、両方が同じクラスタにクラスタリングしている場合はカウントしない。この距離を用いてパラメータを決定する。

パラメータ数決定アルゴリズム(クラスタ数、次元数等)

１　何回アルゴリズムを繰り返すか決めておく(R回)

２　データ$X_{(n)} = (X_1, \dots X_n)$をランダムに並び替える$(X_1^{(r)}, \dots X_n^{(r)})$

３　並び替えたデータをm,m,n-2mの三種類に分ける。つまり

$Y_1^{(r)} = (X_1^{(r)}, \dots X_m^{(r)}), Y_2^{(r)} = (X_{m+1}^{(r)}, \dots X_{2m}^{(r)}), Z^{(r)} = (X_{(2m+1)}^{(r)}, \dots X_n^{(r)})$

４　$Y_1^{(r)}$で作成したクラスタリングc_1,$Y_2^{(r)}$で作成したクラスタリングc_2を用いて$Z^{(r)}$をクラスタリングする。

５　$Z^{(r)}$内のデータから二個ずつ取り出し、すべてのパターンで

$1_{\{1_{(c_1(X)=c_1(Y))}+1_{(c_2(X)=c_2(Y))}=1\}}$を計算し、総和を最小化するパラメータの組を求める。

６　２から５をR回繰り返し、パラメータの組の最頻値(ベクトル)を採用する

(RKM.RとclustCV.pyにコードを書いた)