PCA and LDA

jojonki

Feb. 2021

1 はじめに

PCA (Principal Component Analysis) と LDA (Linear Discriminant Analysis) について勉強したのでまとめた．いずれの手法も射影による次元削減を行えるが，LDA はラベル付き事例に対するアプローチという点で大きく異なる．

2 PCA (Principal Component Analysis)

PCA は射影である．PCA ではデータをある軸 $\boldsymbol{u}$ に直角に射影（正射影）したときに，原点（データの平均点）からの距離を最大化するような軸を探す．そのためにもまず正射影ベクトルについて復習する．あるデータ $\boldsymbol{x}$ を軸 $\boldsymbol{u}$ に正射影したときの，影の長さ（スカラ）はその内積で得られる．

$$\boldsymbol{x} \cdot \frac{\boldsymbol{u}}{|\boldsymbol{u}|} \tag{1}$$

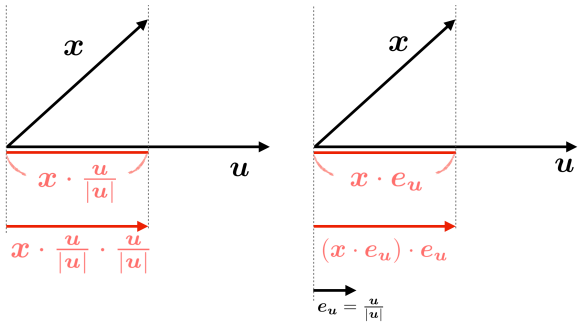

Figure 1: 正射影ベクトルの長さとベクトル（左と右はどちらも同じ）

まずデータは簡単のために平均を 0 にしておく．PCA でまず最初に得たい軸は，射影したときに原点からの距離を最大化するような軸である．図で考えると分かりやすいが，下図では左の図のほうが原点からの距離が遠く，１次元に落としてもその前の次元の位置関係をうまく維持できているのがわかる．

そのため，最適な軸を探す作業は全事例の投射後の距離の 2 乗平均を最大化する問題に帰結できる．分散を最大化するといってもよい．ある軸 $\boldsymbol{u}$ に射影したときの式は以下のようになる．正射影したときの $\boldsymbol{u}$ 軸に対する長さは内積で得られる．事例を $\boldsymbol{x}$，事例数を m，射影する軸を $\boldsymbol{u}$ としている．また $|\boldsymbol{u}|^2 = 1$ としておく．

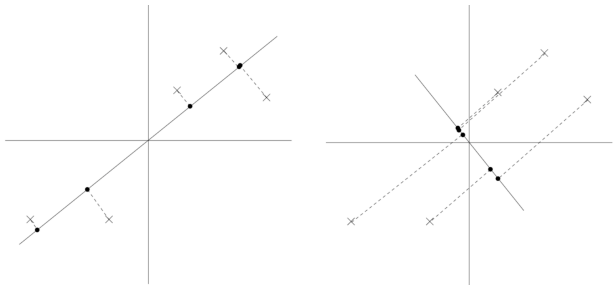

Figure 2: 左：投射後の距離が遠い，右：投射後の距離が近い．図は CS229 より借用.

$$
\begin{aligned}
\frac{1}{m}\sum_i (\boldsymbol{x}^T\boldsymbol{u})^2 &= \frac{1}{m}\sum_i (\boldsymbol{x}^T\boldsymbol{u})^T(\boldsymbol{x}^T\boldsymbol{u}) \\
&= \frac{1}{m}\sum_i (\boldsymbol{u}^T\boldsymbol{x})(\boldsymbol{x}^T\boldsymbol{u}) \\
&= \frac{1}{m}\sum_i \boldsymbol{u}^T\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{u} \\
&= \boldsymbol{u}^T\left(\frac{1}{m}\sum_i \boldsymbol{x}\boldsymbol{x}^T\right)\boldsymbol{u} \\
&= \boldsymbol{u}^T\Sigma\boldsymbol{u}
\end{aligned}
\tag{2}
$$

このように分散共分散行列 $\sum$ がでてくる．これを最大化する $\boldsymbol{u}$ を求めたい．そこで $|\boldsymbol{u}|^2 - 1 = 0$ という制約条件を使い，ラグランジュの未定乗数法で求める.

$$
\mathcal{L}(\boldsymbol{u}, \lambda) = \boldsymbol{u}^T\Sigma\boldsymbol{u} - \lambda(|\boldsymbol{u}|^2 - 1) \tag{3}
$$

$\boldsymbol{u}$ で微分して 0 とおくと，

$$
\begin{aligned}
\frac{\partial\mathcal{L}(\boldsymbol{u}, \lambda)}{\partial\boldsymbol{u}} = (\Sigma + \Sigma^T)\boldsymbol{u} - 2\lambda|\boldsymbol{u}| &= 0 \\
\Sigma\boldsymbol{u} &= \lambda\boldsymbol{u}
\end{aligned}
\tag{4}
$$

ベクトル微分の公式 $\frac{d}{dx}\boldsymbol{x}^T A\boldsymbol{x} = (A + A^T)\boldsymbol{x}$，及び分散共分散行列は対称行列であることを利用した.

ところでこれは固有方程式である．つまり分散を最大化する軸は，分散共分散行列に対する固有方程式に対応している．つまり，射影したいベクトル $\boldsymbol{u}$ は の固有ベクトルであり，$k(k < n)$ 次元目までの固有ベクトルを選択すれば，n 次元から k 次元にデータを圧縮できる．よって新しい k 次元のデータ $\boldsymbol{x}'$ は下式で得られる.

$$
\boldsymbol{x}' = \boldsymbol{x} \cdot \boldsymbol{u}[:, k] \tag{5}
$$

また各固有値を固有値の累積和で割って正規化した値を寄与率と呼ぶ.

2.1 Python コード

iris データを使って自前で PCA したコードを下記に記載している．iris データにはラベルが存在するため色付けしているが，PCA の計算にラベルは不要であり，ラベルなしデータに対して潜在的なクラスターを発見できる可能性がある.

`https://github.com/jojonki/DimensionalityReduction/blob/main/PCA_iris.ipynb`

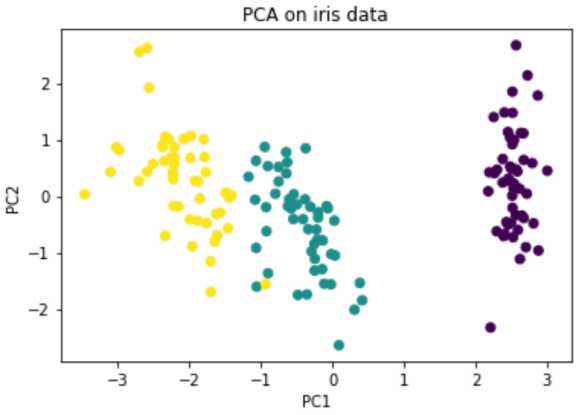

Figure 3: 4 次元の iris データを PCA で 2 次元に圧縮

3 Linear Discriminant Analysis

次に LDA を説明する．LDA はクラスラベル付の事例が与えられたときに，クラスをうまく分離する軸を見つけることができる．PCA ではクラスラベルはなく，データ全体の分散を最大化する軸を探したが，LDA では，投射語の事例のクラス内の分散は小さく，クラス間の分散は大きくするようなうまい分離軸を探す．そこで下記の $J(\boldsymbol{w})$ を最大化する問題に帰着できる．

$$J(\boldsymbol{w}) = \frac{\text{クラス間分散}}{\text{クラス内分散}} \tag{6}$$

式の表現として，例えば 2 クラスのデータを考える下記のようになる．事例 $\boldsymbol{x}$，射影する重み $\boldsymbol{w} = (w_1, w_2)^T$，投影後のデータ $\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}$，クラス 1 のデータ数を n_1，クラス 2 は n_2 個とするとこのように表せる．バーは平均を表す．

$$\begin{aligned}
\boldsymbol{y}_i^{(1)} &= \boldsymbol{w}^T \boldsymbol{x}_i^{(1)} \\
\boldsymbol{y}_i^{(2)} &= \boldsymbol{w}^T \boldsymbol{x}_i^{(2)} \\
\overline{\boldsymbol{y}}^{(1)} &= \frac{1}{n_1} \sum_i^{n_1} \boldsymbol{w}^T \boldsymbol{x}_i^{(1)} = \boldsymbol{w}^T \overline{\boldsymbol{x}}^{(1)} \\
\overline{\boldsymbol{y}}^{(2)} &= \frac{1}{n_2} \sum_i^{n_2} \boldsymbol{w}^T \boldsymbol{x}_i^{(2)} = \boldsymbol{w}^T \overline{\boldsymbol{x}}^{(2)}
\end{aligned} \tag{7}$$

それでは多クラスの場合に一般化して，クラス間分散とクラス内分散を導く．

3.1 クラス間分散

クラスとクラスの間はできるだけ離したいクラス間分散．データ全体の平均 $\overline{\boldsymbol{y}}$ と各クラス c の平均 $\overline{\boldsymbol{y}}^{(c)}$ の差の 2 乗を求めれば良い．n_c はクラス c の事例数でクラス c の分散の重みに該当する．本当は全事例数で割る必要があるが，$J(w)$ のクラス間分散とクラス内分散の分母分子どちらにも現れて打ち消されるので計算しなく

て良い.

$$
\begin{aligned}
J_B(\boldsymbol{w}) = \sum_c n_c(\overline{\boldsymbol{y}}^{(c)} - \overline{\boldsymbol{y}})^2 &= \sum_c n_c \left(\boldsymbol{w}^T(\overline{\boldsymbol{x}}^{(c)} - \overline{\boldsymbol{x}})\right)^2 \\
&= \sum_c n_c \left((\overline{\boldsymbol{x}}^{(c)} - \overline{\boldsymbol{x}})^T\boldsymbol{w}\right)^2 \\
&= \sum_c n_c \left((\overline{\boldsymbol{x}}^{(c)} - \overline{\boldsymbol{x}})^T\boldsymbol{w}\right)^T \left((\overline{\boldsymbol{x}}^{(c)} - \overline{\boldsymbol{x}})^T\boldsymbol{w}\right) \\
&= \sum_c n_c \boldsymbol{w}^T(\overline{\boldsymbol{x}}^{(c)} - \overline{\boldsymbol{x}})(\overline{\boldsymbol{x}}^{(c)} - \overline{\boldsymbol{x}})^T\boldsymbol{w} \\
&= \boldsymbol{w}^T \sum_c n_c(\overline{\boldsymbol{x}}^{(c)} - \overline{\boldsymbol{x}})(\overline{\boldsymbol{x}}^{(c)} - \overline{\boldsymbol{x}})^T\boldsymbol{w} \\
&= \boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}
\end{aligned}
\tag{8}
$$

3.2 クラス内分散

クラス内はできるだけ密にしたいクラス内分散．各クラス内において，各事例とクラス平均 $\overline{\boldsymbol{x}}^{(c)}$ の差の 2 乗を求めれば良い．こちらも事例数に応じて重みが大きくなる．全事例数で割っていないのは先ほど同様にクラス間分散でも割っていないから．

$$
\begin{aligned}
J_W(\boldsymbol{w}) &= \sum_c \sum_i^{n_c} \left(\boldsymbol{w}^T\boldsymbol{x}_i^{(c)} - \boldsymbol{w}^T\overline{\boldsymbol{x}}^{(c)}\right)^2 \\
&= \sum_c \sum_i^{n_c} \left(\boldsymbol{w}^T\left(\boldsymbol{x}_i^{(c)} - \overline{\boldsymbol{x}}^{(c)}\right)\right)^2 \\
&= \sum_c \sum_i^{n_c} \boldsymbol{w}^T\left(\boldsymbol{x}_i^{(c)} - \overline{\boldsymbol{x}}^{(c)}\right)\left(\boldsymbol{x}_i^{(c)} - \overline{\boldsymbol{x}}^{(c)}\right)^T\boldsymbol{w} \\
&= \boldsymbol{w}^T\left(\sum_c \sum_i^{n_c} \left(\boldsymbol{x}_i^{(c)} - \overline{\boldsymbol{x}}^{(c)}\right)\left(\boldsymbol{x}_i^{(c)} - \overline{\boldsymbol{x}}^{(c)}\right)^T\right)\boldsymbol{w} \\
&= \boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}
\end{aligned}
\tag{9}
$$

3.3 最大化

$$
\begin{aligned}
J(\boldsymbol{w}) &= \frac{\text{クラス間分散}}{\text{クラス内分散}} \\
&= \frac{\boldsymbol{J}_B(\boldsymbol{w})}{\boldsymbol{J}_W(\boldsymbol{w})} \\
&= \frac{\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w}}
\end{aligned}
\tag{10}
$$

この目的関数を微分して 0 になるような $\boldsymbol{w}$ を求める．商の微分公式などを使う．

$$
\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = \frac{(\boldsymbol{S}_B + \boldsymbol{S}_B^T)\boldsymbol{w}(\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w}) - (\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w})(\boldsymbol{S}_W + \boldsymbol{S}_W^T)\boldsymbol{w}}{(\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w})^2} = 0 \tag{11}
$$

分子だけ見れば良い，また $\boldsymbol{S}_B, \boldsymbol{S}_W$ は対称行列であるので，

$$
\begin{aligned}
\boldsymbol{S}_B\boldsymbol{w}(\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w}) &= (\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w})\boldsymbol{S}_W\boldsymbol{w} \\
&\quad \boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w} \text{ はスカラであるので，両辺をこの値で割る} \\
\boldsymbol{S}_B\boldsymbol{w} &= \frac{\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w}}\boldsymbol{S}_W\boldsymbol{w} \\
&\quad \text{両辺に左から } \boldsymbol{S}_W^{-1} \text{ をかける} \\
\boldsymbol{S}_W^{-1}\boldsymbol{S}_B\boldsymbol{w} &= \frac{\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w}}\boldsymbol{w} \\
\boldsymbol{S}_W^{-1}\boldsymbol{S}_B\boldsymbol{w} &= J(\boldsymbol{w})\boldsymbol{w}
\end{aligned}
\tag{12}
$$

ここでもまた固有方程式が出てきた．これは行列 $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$ の固有方程式であり，目的関数 $\frac{\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w}}$ は固有値に該当する．

3.4 Python コード

iris データを使って自前で LDA したコードを下記に記載している．スケール及び符号が反対ではあるが，scikit-learn の LDA とほぼ同じ図が得られた．

`https://github.com/jojonki/DimensionalityReduction/blob/main/LDA_iris.ipynb`

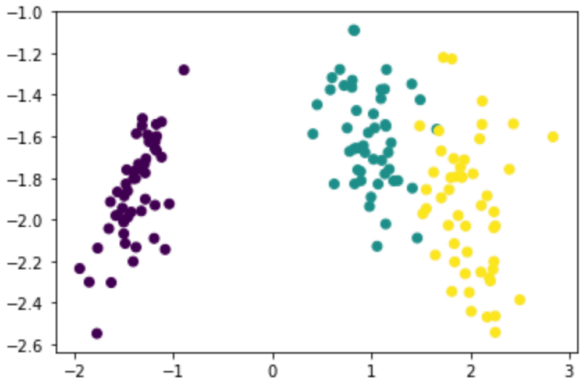

Figure 4: 4 次元の iris データを LDA で 2 次元に圧縮

4 参考

- 多変量解析入門，小西貞則
- ソースコード `https://github.com/jojonki/DimensionalityReduction`
- CS229 Lecture notes 10 (ミラー) `https://github.com/Kivy-CN/Stanford-CS-229-CN/blob/master/CS229%E5%AE%98%E7%BD%91%E5%BD%93%E5%89%8D%E6%96%87%E6%A1%A3/notes/cs229-notes10.pdf`
- Technical Note- 線形判別分析 (LDA) `https://hkawabata.github.io/technical-note/note/ML/Preprocess/lda.html`