CENTRO UNIVERSITÁRIO FEI
Leonardo Contador Neves – 118315-1

Tópicos Especiais em Aprendizagem:

Principal Component Analysis (Análise da Componente Principal)

São Bernardo do Campo, SP

2018

SUMÁRIO

1 INTRODUÇÃO

Este relatório busca a implementação do algoritimo de Análise das Componentes Principais (PCA) para nos fornecer as componentes principais de um conjunto de dados e assim reduzir as dimenções da nossa amostra, facilitando assim, por exemplo, a classificação das informações.

2 REVISÃO BIBLIOGRÁFICA

A **Análise de Componentes Principais** (PCA) é um procedimento matemático que utiliza uma transformação ortogonal (ortogonalização de vetores) para converter um conjunto de observações de variáveis possivelmente correlacionadas num conjunto de valores de variáveis linearmente não correlacionadas chamadas de **componentes principais**. O número de componentes principais é menor ou igual ao número de variáveis originais. Esta transformação é definida de forma que o primeiro componente principal tem a maior variância possível (ou seja, é responsável pelo máximo de variabilidade nos dados), e cada componente seguinte, por sua vez, tem a máxima variância sob a restrição de ser ortogonal a (i.e., não correlacionado com) os componentes anteriores. Os componentes principais são garantidamente independentes apenas se os dados forem normalmente distribuídos (conjuntamente). O PCA é sensível à escala relativa das variáveis originais. Dependendo da área de aplicação, o PCA é também conhecido como transformada de Karhunen-Loève (KLT) discreta, transformada de Hotelling ou decomposição ortogonal própria (POD).

O PCA foi inventado em 1901 por Karl Pearson. Agora, é mais comumente usado como uma ferramenta de Análise Exploratória de Dados e para fazer modelos preditivos. PCA pode ser feito por decomposição em autovalores (Valores Próprios) de uma matriz covariância, geralmente depois de centralizar (e normalizar ou usar pontuações-Z) a matriz de dados para cada atributo. Os resultados de PCA são geralmente discutidos em termos pontuações (*scores*) de componentes, também chamados de pontuações de fatores (os valores de variável transformados correspondem a um ponto de dado particular), e carregamentos (*loadings*), i.e., o peso pelo qual cada variável normalizada original deve ser multiplicada para se obter a pontuação de componente.

O PCA é a mais simples das verdadeiras análises multivariadas por autovetores (Vetores Próprios). Com frequência, sua operação pode ser tomada como sendo reveladora da estrutura interna dos dados, de uma forma que melhor explica a variância nos dados. Se visualizarmos

um conjunto de dados multivariados em um espaço de alta dimensão, com 1 eixo por variável, o PCA pode ser usado para fornecer uma visualização em dimensões mais baixas dos mesmos dados, uma verdadeira "sombra" do objeto original quando visto de seu ponto mais informativo. Isto é feito usando-se apenas os primeiros componentes principais, de forma que a dimensionalidade dos dados transformados é reduzida.

O PCA é fortemente ligado à análise de fatores (Factorial Analysis); de fato, alguns pacotes estatísticos propositadamente confluem as técnicas. A verdadeira análise de fatores faz suposições diferentes sobre a estrutura subjacente dos dados e encontra os autovetores de uma matriz levemente diferente.

3 METODOLOGIA

Para implementar a Análise das Componentes Principais vamos começar fazendo alguns cálculos básicos da nossa amostra. Primeira mente, vamos calcular a média das variáveis e vamos aplicar para cada par de variáveis a covariação das mesmas. A equação a baixo mostra o calculo da média para n elementos, onde x1 … xn são os valores do conjunto de informações.

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Equação 1: Cáculo da média

Para a variancia, como demostrada na Equação 2, temos que n é o numero de elementos na amostra, xi é cada elemento da amostra e x barrado é a média da amostra. Lembrando que foi se usado (n-1) na equação, pois estamos trabalhando com uma amostra do nosso conjunto de informações, se a média fosse feita com o conjunto todo bastaria usar (n).

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2,$$

Equação 2: Cálculo da variação

Para proceguirmos, vamos calcular agora a covariação das variáveis, ou seja, a variação de uma variável com a sua susequênte e assim podemos montar uma matriz de covariação. A equação a seguir, mostra o cálculo das corariâncias e a subsequênte mostra a formação da matriz de covariação.

$$cov(x,y) = \sqrt{\frac{\sum(x_i - \bar{x}_i)(y_i - \bar{y}_i)}{N-1}}$$

Equação 3: Cálculo da covariação

$$C^{n\times n} = (c_{i,j},\ c_{i,j} = cov(Dim_i, Dim_j))$$

Equação 4: Formação da matriz de covariação

Agora, a partir da matriz de covariação, precisamos descobrir os autovetores e autovalores dessa matriz. Para isso, vamos olhar para a definição, onde existe uma matriz A (matriz de autovetores) transposta, que multiplicada pela matriz de covariação B e multiplicada mais uma vez A se iguala à uma matriz de zeros, ond a diagonal principal é caracterizada pelos autovalores de B. A equação é demonstrada a seguir:

$$\Phi^T \Sigma_x \Phi = \Lambda$$

Equação 5: Demonstração da matriz de autovetores e autovalores

Para o calculo dos autovetores e autovalores vamos usar um método implementado na biblioteca que vamos usar no aplicação que faz o uso de método de Jacobi. Esse método iterativo é descrito como:

Dado uma matriz A:

$$A\mathbf{x} = \mathbf{b}$$

Onde, pode ser de n dimensões:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

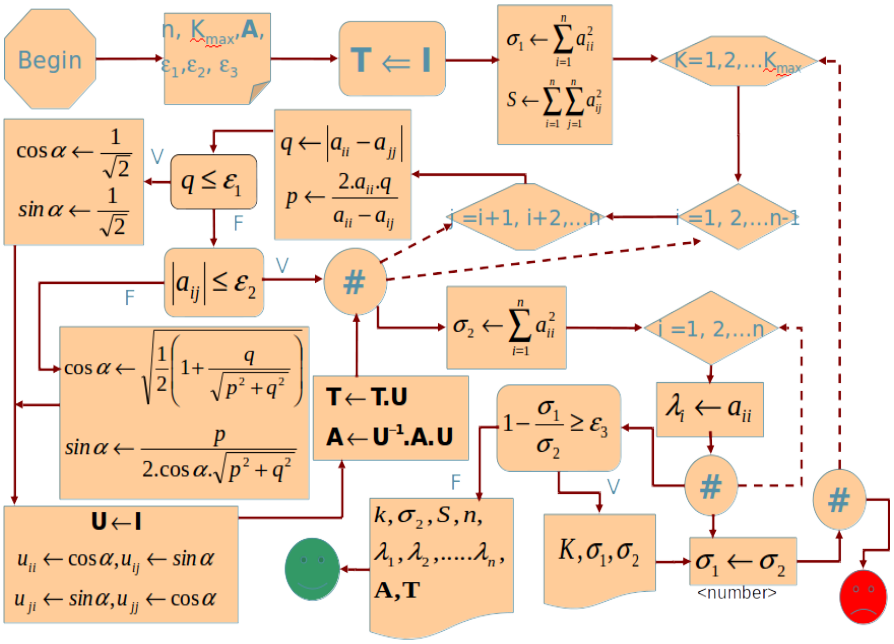
A matriz A, pode ser decomposta em:

$$A = D + R \qquad \text{where} \qquad D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \text{ and } R = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{bmatrix}.$$

E a solução é:

$$\mathbf{x}^{(k+1)} = D^{-1}(\mathbf{b} - R\mathbf{x}^{(k)}).$$

E assim, o método iterativo de Jacobi, da-se por :



E assim, os autovetores da nossa matriz de covariação são os componentes principais que descrevem nosso conjunto de variáveis. Com os cáculos, chegamos que dada n variáveis, o número de componentes principais da nossa amostra vai ser sempre igual a n e assim, os autovetores descobertos, estão em ordem de significancia, onde a primeira componente principal representa a maior covariância dentre as variáveis, a segunda menos que a primeira e assim sucessivamente.

4 DESENVOLVIMENTO

Para implementar os algoritmos a cima descritos, foi feita uma Classe em *Python* que tem duas funções principais, onde uma delas é a *fit()* que basicamente implementa o algoritmo de PCA e a segunda é uma auxiliar que retorna a matriz de covariação. Os outros cáculos envolvidos no algoritmo, foram feitos através da biblioteca do *Python para operações matemáticas, a Numpy*.

Com a biblioteca *Numpy* fazemos uso da função que calcula os autovetores e autovalores e de operações de multiplicação de matrizes simples e transposição.

4.1 CLASSE PARA CÁLCULO DO PCA

4.1.1 Método de *fit* (Cálculo do PCA)

Esse é o método principal do algoritmo, nele é implementado o cálculo do PCA descrito no item de Metodologia deste relatório. Primeiramente, dado os vetores de entrada (nossas variáveis a serem reduzidas dimensionalmente), vamos calcular um vetor de médias das variáveis e um vetor que contem as variáveis onde cada elemento se subtrai a média. Despois disso, é calculada uma matriz de covariação dessas variáveis, usando o método de covariação desenvolvido. Esses procedimentos são descritos na imagem à seguir:

```
self.vectors = vectors

#creating a mean vector
self.vect_med = [np.array([x]).mean() for x in vectors]

#subtracting the mean
vectors = (np.array(vectors).T - self.vect_med).T

#Covariance Matrix
covarMatrix = np.ones([len(vectors),len(vectors)])
for i in range(len(vectors)):
    for j in range(len(vectors)):
        covarMatrix[i][j] = self.getCovariance(vectors[i], vectors[j])
```

Com a matriz de covariação, vamos agora descobrir os auto vetores e autovalores dessa matriz. Para descobrí-los, vamos usar o método *eig()* da biblioteca *Numpy* do *Python*, onde o retorno é a matriz de autovetores e um metor de autovalores, como mostra na imagem a seguir:

```
#Computing the eigenvalues and right eigenvectors of a square array.
self.W, self.v = np.linalg.eig(covarMatrix)
```

Com os autovetores e autovalores, precisamos agora criar um vetor de componentes principais (a partir dos autovetores) para dada as variáveis de entrada, ajustá-las no nosso novo eixo de componentes principais. A imagem a seguir mostra a criação do vetor de características (vetor de autovetores escolhidos), sua multiplicação pelo vetor com continha a amostra subtraída da média e a soma da média de novo, ajustando a amostra ao novo sistema de coordenadas.

```
#Computing the feature vector with the number of principal components
feature_vector = self.v[:,0:n_components]
#Multiplying the feature vector by a vector with data minus the mean
final_data = feature_vector.T.dot(np.array(vectors))
#getting the data fitted to principal component
original_data = np.add(feature_vector.dot(final_data), np.array([self.vect_med]).T)
```

4.1.2 Método de Cálculo da Covariância

Esse método faz a aplicação direta da equação de covariância. A imagem a seguir mostra sua implementação.

```
def getCovariance(self, x, y):
    if len(x) == len(y):
        #subtracting the mean
        cov = 0
        for i in range(len(x)):
            cov = cov + (x[i] - np.array([x]).mean()) * (y[i] - np.array([y]).mean())
        return (cov/(len(x)-1))
    else:
        print("There are vectors with different lenghts!!")
        return "error"
```

5 RESULTADOS

O algoritmo implementado foi testado em três conjuntos de informações. Para o primeiro conjunto, existem duas variáveis, onde a redução de dimensão faz com que as informações se ajustem em um eixo prinipal. A imagem a seguir mostra a dispersão das variáveis ao longo dos dois eixos e em seguida, a figura mostra a aplicação do algorítimo e a descoberta da componente principal e o ajuste da amostra na componente principal para a primeira base (alpswater.xlsx).

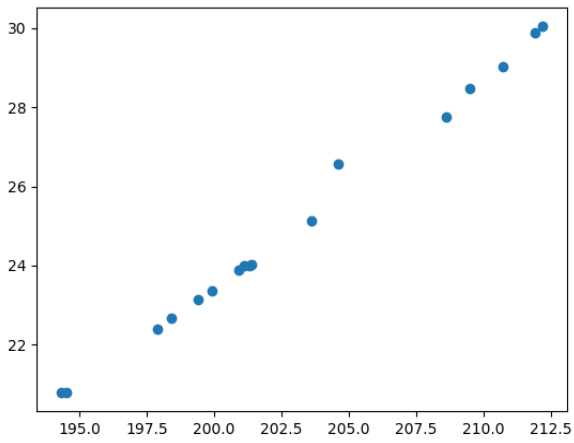


Gráfico 1: Gráfico da amostra

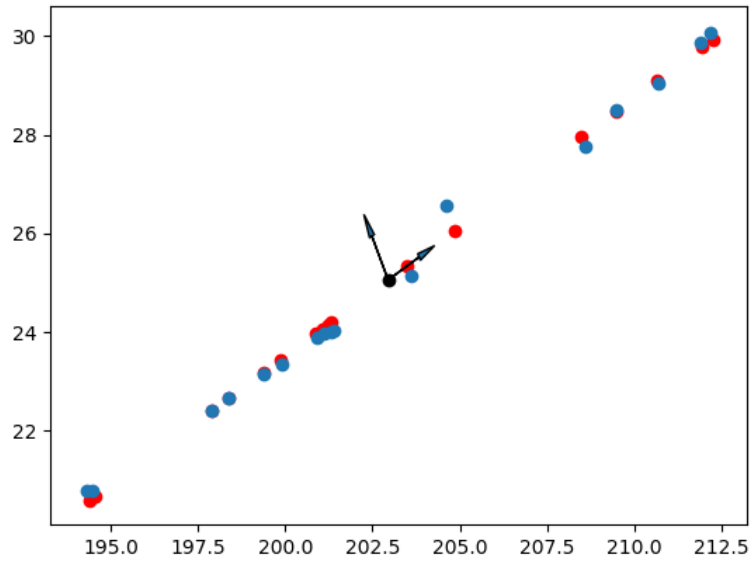


Gráfico 2: Gráfico da amostra (pontod azuis) e amostra ajustada à componente principal (pontos vermelhos)

Para a segunda base (USCensus.xls), os dados tinham um com portamento polinomial, como vemos a curva formada pelos pontos azuis na imagem a seguir. Os pontos em vermelho, mostram a nova amostra ajustada em um eixo da componente principal, como mostra o vetor de componentes na figura.

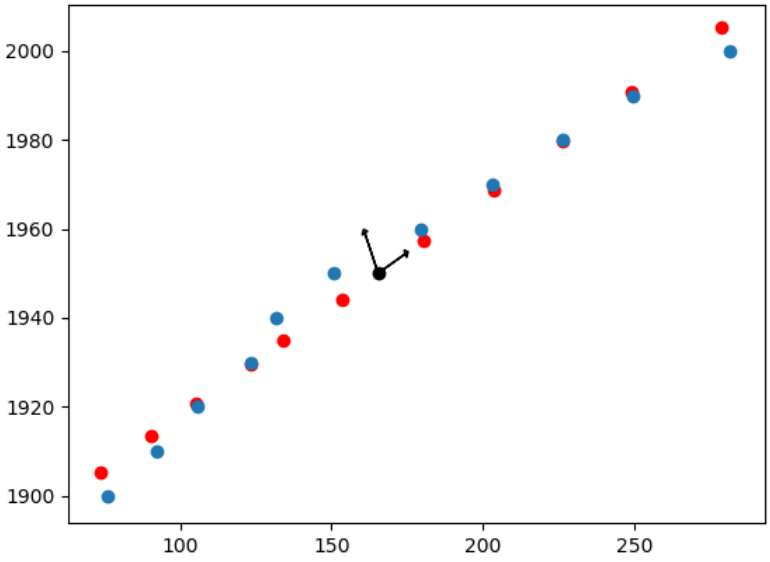


Gráfico 3: Ajuste das variáveis à componente principal

Para a terceira base (Books_attend_grade.xls), temos três variáveis e vamos reduzir para duas componentes principais. A imagem a seguir mostra a dispersão das informações no plano 3d e consequêntemente a redução dimencional para apenas dois eixos, podendo-se assim enquadrar a amostra num plano cartesiano.

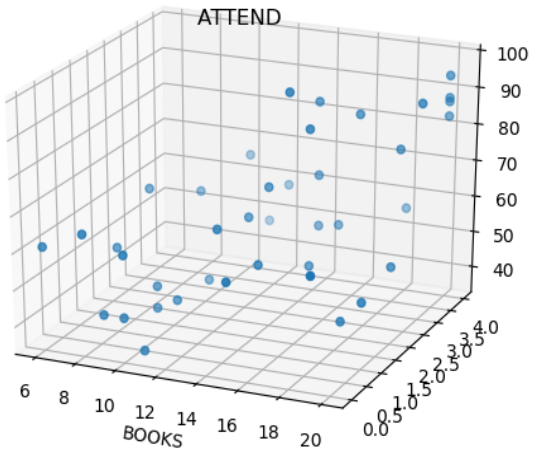


Gráfico 4: Dispersão das variáveis no espaço

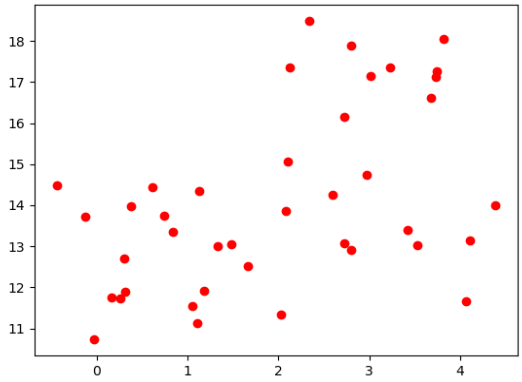


Gráfico 5: Dispersão das variáveis ajustadas à 2 componentes principais

5 CONCLUSÃO

O método PCA se mostrou eficiente para o conjunto de amostras ajustadas, porém para muitas variáveis, dado o método de Jacobi, pode-se exigir um poder computacional um pouco maior. Em comparação com o método dos mínimos quadrados (MMQ), com o PCA, vemos que podemos, na redução de dimensões, achar uma reta que se ajusta com a distribuição de variância da amostra, sendo assim, podemos aplicar PCA para compor uma reta de predição, porém não é muito recomendado, visto que para o cálculo do PCA, mais poder computacional é usado do que o método dos mínimos quadrados e a reta das componentes principais geradas, apresentam em algum ponto erro muito maior que o MMQ, dado que esse erro é corrigido à cada iteração do MMQ e no PCA não nos preocupamos com o erro e sim com a covariância da amostra.

REFERÊNCIAS

VARIÂNCIA. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2017. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Vari%C3%A2ncia&oldid=48416835>. Acesso em: 30 mar. 2017.

ANÁLISE DE COMPONENTES PRINCIPAIS. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2018. Disponível em: <https://pt.wikipedia.org/w/index.php?title=An%C3%A1lise_de_componentes_principais&oldid=53098518>. Acesso em: 10 set. 2018.

NumPy Reference. Disponível em: <https://docs.scipy.org/doc/numpy-1.15.1/reference/index.html>. Acesso em: 10 set. 2018.

MÉTODO DOS MÍNIMOS QUADRADOS. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2018. Disponível em: <https://pt.wikipedia.org/w/index.php?title=M%C3%A9todo_dos_m%C3%ADnimos_quadrados&oldid=51604442>. Acesso em: 24 mar. 2018.