Análise de Componentes Principais: Aplicação em Pricing e Redução de Imagens

Luiz Gonzaga da Silva Junior[a]

[a]Institute of Mathematical and Computer Sciences, University of São Paulo, São Carlos, Brazil

ARTICLE HISTORY
Compiled November 10, 2019

1. Introdução

Em muitas empresas a área de Pricing é vista como uma área de suporte de dados que executa as decisões de outras áreas. Porém, hoje temos um conceito que vai além de apoiar as demais áreas de negócio com informações, ferramentas e sistemas de análise histórica, as chamadas ferramentas de BI. Atualmente essa área possui mais autonomia, elas tem o dever de montar uma estratégia e levar essa mudança de preço para seus clientes finais.

O preço é o único elemento do mix de marketing que produz receita, sendo também um dos elementos mais flexíveis: pode ser alterado com rapidez, ao contrário das caracteristicas de produtos, dos compromissos com os canais de distribuição e até as promoções. O preço informa ao mercado o posicionamento de valor pretendido pela empresa para seu produto ou marca. Um produto bem desenhado e comercializado pode determinar um preço superior e obter alto lucro (KOTLER e KELLER p.428) [1].

Em um modelo eficiente, a área de Pricing centraliza informações provenientes de diferentes áreas da empresa, e mais recentemente, de fontes externas como redes sociais, para aplicar em suas mudanças de preços.

Na prática a área se apoia em um conjunto de indices que são um conjunto informações quantitativas e qualitativas decorrentes de transações efetuadas no passado (análises ocorrem normalmente em D-1). Informações qualitativas a respeito de uma transação são o produto negociado, cliente que comprou, o canal de vendas desse produto, e muitas outras que podem variar de negócio para negócio. A informações quantitativas que caracteriza a transação é o valor faturado, a quantidade negociada do produto, os descontos aplicado na campanha, valores gastos com impostos e demais taxa, custo do produto e valor gasto em frete.

Apesar da quantidade de informação disponível e complexidade que demanda analisar todas elas, todas são variáveis do mesmo problema e devem ser consideradas simul-

*Corresponding author. Email: luiz.gonzaga.silva@usp.br

taneamente na tomada de decisão. É frequente haver redundância entre as dimensões. A análise de componentes principais pode ser usado na remoção das redundâncias e consequentemente na redução da dimensionalidade por explorar a interdependência em dados multivariados. A forma que essa técnica trabalha é transformar um conjunto de variáveis originais em outro conjunto de variáveis de mesma dimensão denominados de componentes principais REGAZZI [2].

2. Matriz de Dados

A matriz de dados é formada pelo conjunto de n observações das p características que uma determinada população possui. As características são representadas pelo vetores $X_1, X_2, ..., X_p$ de dimensão n.

$$X_1 = \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \\ \vdots \\ x_{n1} \end{bmatrix}, X_2 = \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \\ \vdots \\ x_{n2} \end{bmatrix}, ..., X_p = \begin{bmatrix} x_{1p} \\ x_{2p} \\ x_{3p} \\ \vdots \\ x_{np} \end{bmatrix}$$

Portanto temos uma matriz de n linhas e p colunas, que chamaremos de matriz X.

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & ... & x_{1p} \\ x_{21} & x_{22} & x_{23} & ... & x_{2p} \\ x_{31} & x_{32} & x_{33} & ... & x_{3p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & ... & x_{np} \end{bmatrix}$$

A relação de interdependência que buscamos para análise dos componentes principais encontramos na matriz de covariância S ou na matriz de correlação R. Porém conhecer essa estrutura através dos vetores $X_1, X_2, X_3, ..., X_p$ pode se tornar algo dispendioso. Para contornar tais dificuldades, na análise de componentes principais transformamos os vetores de origem $X_1, X_2, X_3, ..., X_p$ em outros vetores $Y_1, Y_2, Y_3, ..., Y_p$, de modo que nesse novo conjunto de variáveis não haja uma estrutura de interdependência. Outra característica que buscamos é um conjunto de variância decrescente para esse novo conjunto de variáveis, isso permitirá o descarte de variáveis de pouca representividade no conjunto de dados.

Abaixo um exemplo de dados fictícios em 2D onde as variáveis são X1 e X2.

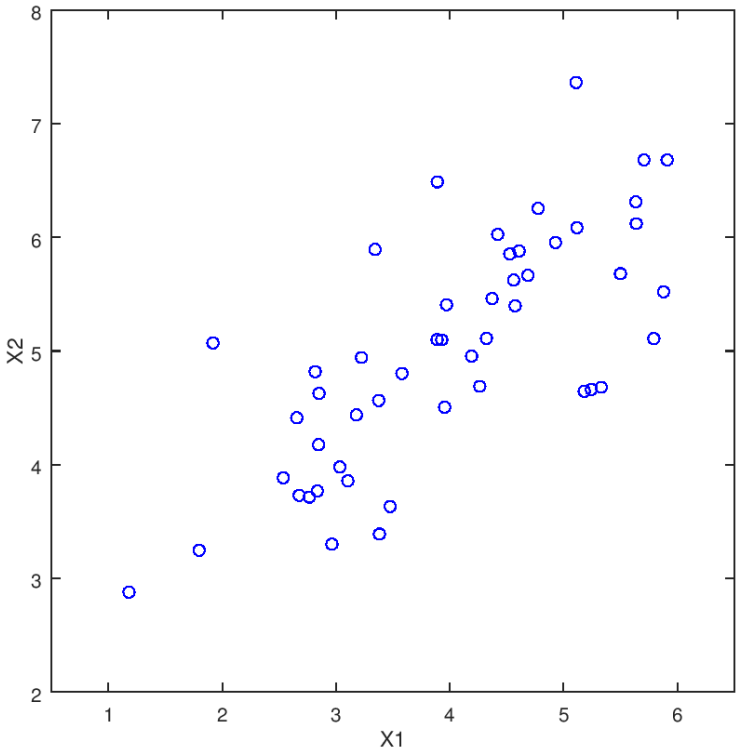

Figure 1. Dados de Exemplo

3. Matriz de Covariância

Como comentado anteriormente, a matriz de covariância revela a estrutura de interdependência entre as veriáveis $X_1, X_2, X_3, ..., X_p$. Essa é uma matriz simétrica de dimensão pxp.

$$S = \begin{bmatrix} cov(X_1, X_1) & cov(X_1, X_2) & cov(X_1, X_3) & ... & cov(X_1, X_p) \\ cov(X_2, X_1) & cov(X_2, X_2) & cov(X_2, X_3) & ... & cov(X_2, X_p) \\ cov(X_3, X_1) & cov(X_3, X_2) & cov(X_3, X_3) & ... & cov(X_3, X_p) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ cov(X_p, X_1) & cov(X_p, X_2) & cov(X_p, X_3) & ... & cov(X_p, X_p) \end{bmatrix},$$

$$Cov(X, Y) = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{n-1}$$

Onde $\overline{X}$ é média aritmética da veriável X, $\overline{Y}$ é média aritmética da veriável Y, n número de observações.

Como é comum as caracteristicas dos individuos serem observadas em unidades de medidas distintas, é recomendado a padronização das variáveis $X_1, X_2, X_3, ..., X_p$

em $Z_1, Z_2, Z_3, ..., Z_p$. Um tipo de padronização muito utilizada possui média *zero* e variância *um*.

$$Z_{ij} = \frac{\sum_{i=1}^{n}(x_{ij} - \overline{x_j})}{S(x_j)}, \quad i = 1, 2, ..., p \quad \text{e} \quad j = 1, 2, ..., p$$

Onde $S(X_j) = \hat{V}ar(X_j)^{\frac{1}{2}} = \frac{\sum_{i=1}^{n}(x_{ij} - \overline{X_j})^2}{n-1}$ é o desvio padrão da característica j.

A matriz padronizada é escrita da seguinte forma:

$$Z = \begin{bmatrix} z_{11} & z_{12} & z_{13} & ... & z_{1p} \\ z_{21} & z_{22} & z_{23} & ... & z_{2p} \\ z_{31} & z_{32} & z_{33} & ... & z_{3p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & z_{n3} & ... & z_{np} \end{bmatrix}$$

Uma propriedade importante da matriz padronizada é que sua matriz de correlação é a mesma da matriz original X. Podemos, portanto, determinar os componentes principais a partir da matriz de correlação R.

4. Componentes Principais

Para encontrar os componentes principais devemos resolver a equação característica da matriz de correlação R, que é dada resolvendo o seguinte determinante:

$$det[R - \lambda I] = 0$$

$$R = \begin{bmatrix} r(Z_1, Z_1) & r(Z_1, Z_2) & r(Z_1, Z_3) & ... & r(Z_1, Z_p) \\ r(Z_2, Z_1) & r(Z_2, Z_2) & r(Z_2, Z_3) & ... & r(Z_2, Z_p) \\ r(Z_3, Z_1) & r(Z_3, Z_2) & r(Z_3, Z_3) & ... & r(Z_3, Z_p) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r(X_p, X_1) & r(Z_p, Z_2) & r(Z_p, Z_3) & ... & r(Z_p, Z_p) \end{bmatrix},$$

$$r(X, Y) = \frac{Cov(X, Y)}{S(X)S(Y)} = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2 \sum_{i=1}^{n}(Y_i - \overline{Y})^2}}$$

Se as colunas da matriz R forem independentes, ou seja, nenhuma coluna é combinação linear de outra coluna, a solução do determinante terá p raízes que são os autovalores da matriz R.

A raízes $\lambda_1, \lambda_2, ..., \lambda_p$ da equação possuem a seguinte característica:

$$\lambda_1 > \lambda_2 > \lambda_3 > ... > \lambda_p$$

Para cada autovalor λ_i existe um autovetor $\mathbf{v_i}$, que é dado solucionando o sistema $A\mathbf{v} = \lambda\mathbf{v}$.

$$\mathbf{v_i} = \begin{bmatrix} v_{i1} \\ v_{i2} \\ v_{i3} \\ \vdots \\ v_{ip} \end{bmatrix}$$

O autovetores são ortonormais, ou seja, eles são ortogonais entre si e a soma dos quadrados dos coeficientes é igual a 1.

$$\sum_{i=1}^{p} v_{ij}^2 = 1 \quad (\mathbf{v_i}' \cdot \mathbf{v_i} = 1) \quad \text{e}$$

$$\sum_{i=1}^{p} v_{ij} v_{kj} = 0 \quad (\mathbf{v_i}' \cdot \mathbf{v_k} = 0 \quad \text{para} \quad i \neq k)$$

O i-ésimo componente principal Y_i é dado como uma combinação das variaveis normalizadas $Z_1, Z_2, Z_3, ..., Z_p$ ponderada pelas componentes dos autovetores $\mathbf{v_i}$.

$$Y_i = v_{i1}Z_1 + v_{i2}Z_2 + v_{i3}Z_3 + ... + v_{ip}Z_p$$

As propriedades dos componentes principais são:

(i) Variância do componente principal Y_i é igual ao valor do autovalor λ_i.

$$Var(Y_i) = \lambda_i$$

(ii) Os componentes principais possuem variância decrescente.

$$Var(Y_1) > Var(Y_2) > Var(Y_3) > ... > Var(Y_p)$$

(iii) A soma das variâncias das variáveis padronizadas é igual a soma dos autovalores, que por sua vez é igual a soma das variâncias dos componentes principais.

$$\sum Var(Z_i) = \sum \lambda_i = \sum Var(Y_i)$$

(iv) Os componentes principais não são correlacionados entre si.

$$Cov(Y_i, Y_j) = 0$$

Para nosso de conjunto de dados de exemplo temos os seguintes valores:

$$\lambda_1 = 1.7 \quad \text{e} \quad \lambda_2 = 0.3,$$

$$v_1 = \begin{bmatrix} -0.70711 \\ -0.70711 \end{bmatrix} \quad \text{e} \quad v_2 = \begin{bmatrix} -0.70711 \\ 0.70711 \end{bmatrix}$$

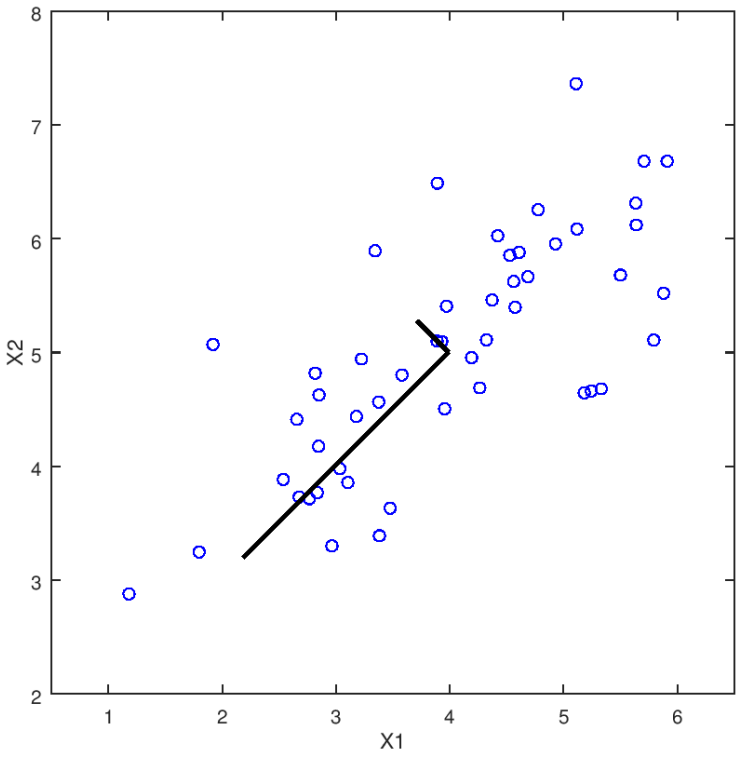

Figure 2. Autovalores e Autovetores

5. Variância explicada

A contribuição C_i de cada componente principal Y_i representa a proporção de variância total explicada pelo componente principal Y_i e é calculada dividindo a variância de Y_i pela variância total.

$$C_i = \frac{Var(Y_i)}{\sum_{i=1}^{p} Var(Y_i)} \cdot 100 = \frac{\lambda_i}{\sum_{i=1}^{p} \lambda_i} \cdot 100$$

Quanto maior o valor da contribuição maior será a importância do componente principal. A soma dos k maiores autovalores representa a proporção de informação retida na redução de p para k dimensões. Isso permite decidir quantos componentes principais vamos manter na análise.

Na figura abaixo estamos projetando os dados de exemplo no primeiro k autovetor: saindo de uma representação 2D para uma 1D.

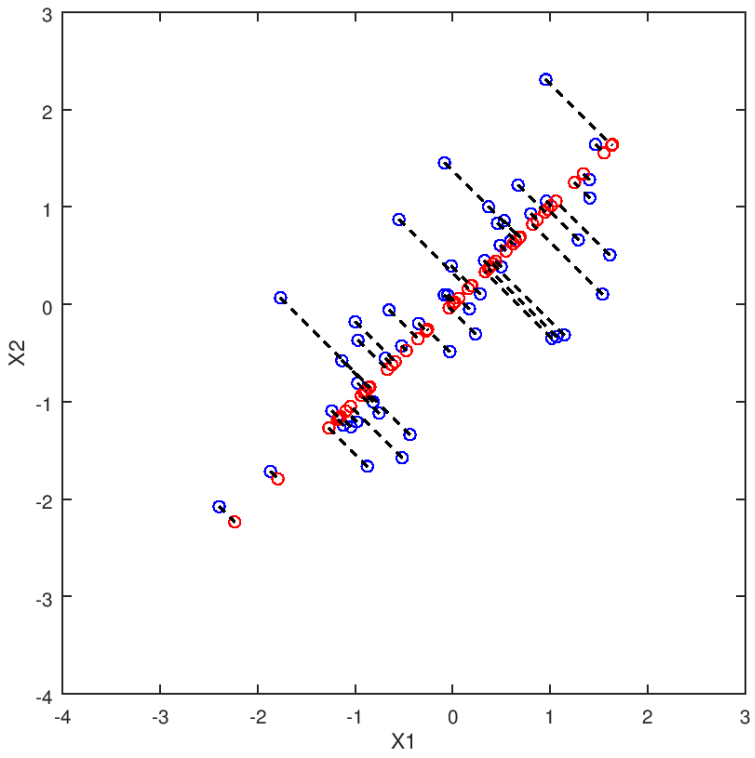

Figure 3. Projeção dados de exemplo na direção do primeiro autovetor

6. Interpretação de cada componente

Verifica-se o grau de influência que cada variável X_j tem sobre o componente Y_i. O grau de influência é dado pela correlação entre cada X_j e o componente Y_i que está sendo interpretado. Por exemplo a correlação entre X_j e Y_1 é:

$$r(X_j, Y_i) = v_{1j} \cdot \frac{\sqrt{Var(Y_1)}}{\sqrt{Var(X_j)}} = \sqrt{\lambda_1} \cdot \frac{v_{1j}}{\sqrt{Var(X_j)}}$$

Comparamos a influência do conjunto $X_1, X_2, X_3, ..., X_p$ sobre Y_1 analisando o peso de cada variável sobre o componente Y_1. O peso de cada variável sobre um componente específico é dado por:

$$w_1 = \frac{v_{11}}{\sqrt{Var(X_1)}}, w_2 = \frac{v_{12}}{\sqrt{Var(X_2)}}, ..., w_p = \frac{v_{1p}}{\sqrt{Var(X_p)}},$$

Sendo w_1 o peso de X_1 sobre Y_1, w_2 o peso de X_2 sobre Y_1, e assim por diante.

7. Aplicação

7.1. *PCA aplicado em Pricing*

Como comentado anteriormemnte, esta área tem que lidar com diversas variáveis em suas análises, porém uma boa parte dessas variáveis podem ser despresadas pois possuem pouca representividade dentro do conjunto de dados (datasetpricing.xlsx). Com intuíto de reduzir o número de variáveis vamos aplica PCA nos dados de transações utilizado pela àrea.

Este conjunto de dados possui 104807 observações com 72 variáveis e trás as informações das transações efetuadas pela empresa no perido nos dias 01/06/2019 e 02/06/2019. O conjunto original possui muito mais variáveis do que as apresentadas aqui, porém a maioria são descritivas e foram descartadas ma análise.

Para nossas análises vamos utilizar o python 3 e as bibiliotecas pandas e numpy.

```
In [1]: import pandas as pd
        import numpy as np

In [261]: df_tran = pd.read_excel('datasetpricing.xlsx', sheet_name='transacoes')

In [290]: df_tran.head(5)
```

Out[290]:

	F_DAY	SKU_ID	TXN_TYPE	QTY_KG	P1	REALIZED_PAYMENT_TERMS	P1_WITH_TERMS	MAX_INFORPRO	UNUSED_INFORPRO	DCA_BOLETO	...
0	20190601	500744	ESTORNO	-10.0	-140.50	-1.500000	-142.000000	NaN	NaN	NaN	...
1	20190601	487171	ESTORNO	-15.0	-154.20	-1.650000	-155.850000	NaN	NaN	NaN	...
2	20190601	42579	ESTORNO	-8.0	-101.92	-1.040000	-102.960000	NaN	NaN	NaN	...
3	20190601	500731	ESTORNO	-12.0	-120.24	-3.840000	-124.080000	NaN	NaN	NaN	...
4	20190601	973500	ESTORNO	-20.0	-223.20	-6.818803	-230.018803	-0.02	-0.02	NaN	...

5 rows × 72 columns

Figure 4. Conjunto completo

A três primeiras variáveis também são descritivas: data, produto e tipo de transação, porém foram mantidas pois não queremos analisar todo o conjunto, vamos restringir a data para 01/02/2016 o produto para 44644 e tipo de transação CAMPO.

Após aplicar os filtros acima e remover campos descritivos, uma parte das variáveis ficaram vazias, ou seja, todas as observações nula. Para esses casos foi removido a coluna toda restando um total de 23 colunas para tratamento.

```
In [267]: X = df_tran[(df_tran['F_DAY']==20190601) & (df_tran['SKU_ID']==44644) & (df_tran['TXN_TYPE']=='CAMPO')].drop(['F_DAY','
          #Ao invés apenas de tratar os NA vou remover as colunas que so possuem NA´s
          X = X.dropna(axis=1, how='all')
          X = X.fillna(value=0)

In [269]: X.head()

Out[269]:
```

	QTY_KG	P1	REALIZED_PAYMENT_TERMS	P1_WITH_TERMS	MAX_INFORPRO	UNUSED_INFORPRO	P2	MAX_D2	D2	SP	...	FINANCIAL_DISC
3091	14.0	262.50	8.26	270.76	27.076	0.0	0.0	72.56	40.404	0.0	...	0.0
3115	21.0	410.97	13.02	423.99	0.000	0.0	0.0	58.47	58.380	0.0	...	0.0
3131	14.0	273.98	5.74	279.72	0.000	0.0	0.0	38.57	36.260	0.0	...	0.0
3145	7.0	131.25	4.13	135.38	13.538	0.0	0.0	36.28	22.652	0.0	...	0.0
3170	35.0	656.25	7.00	663.25	66.325	0.0	0.0	177.75	111.125	0.0	...	0.0

5 rows × 23 columns

Figure 5. Conjunto tratado

Esta é a matriz de dados X que vamos trabalhar, porém como podemos ver, as variáveis possuem unidades distintas, portanto vamos fazer a padronização para média zero e variância um.

```
In [271]: Z = (X - X.mean())/X.std()
          Z = Z.fillna(value=0)
          Z.head()

Out[271]:
```

	QTY_KG	P1	REALIZED_PAYMENT_TERMS	P1_WITH_TERMS	MAX_INFORPRO	UNUSED_INFORPRO	P2	MAX_D2	D2	SP	...
3091	-0.221726	-0.208300	-0.132750	-0.206734	0.763835	-0.266336	-0.142463	0.338979	0.409743	-0.039837	...
3115	-0.086879	-0.026099	0.070195	-0.023467	-0.389987	-0.266336	-0.142463	0.201220	0.742043	-0.039837	...
3131	-0.221726	-0.194212	-0.240191	-0.196018	-0.389987	-0.266336	-0.142463	0.006658	0.333139	-0.039837	...
3145	-0.356573	-0.369370	-0.308834	-0.368652	0.186924	-0.266336	-0.142463	-0.015732	0.081585	-0.039837	...
3170	0.182814	0.274907	-0.186470	0.262694	2.436400	-0.266336	-0.142463	1.367423	1.717073	-0.039837	...

5 rows × 23 columns

Figure 6. Conjunto padronizado

Com o conjunto de dados de padronizados vamos calcular a matriz de correlação R.

```
In [272]: R = Z.corr().fillna(value=0)
          R.head()

Out[272]:
```

	QTY_KG	P1	REALIZED_PAYMENT_TERMS	P1_WITH_TERMS	MAX_INFORPRO	UNUSED_INFORPRO	P2	MAX_D2	
QTY_KG	1.000000	0.993462	0.917202	0.993960	0.086213	-0.011468	0.164485	0.549883	0
P1	0.993462	1.000000	0.902774	0.999927	0.120000	0.000714	0.204855	0.597972	0
REALIZED_PAYMENT_TERMS	0.917202	0.902774	1.000000	0.907898	0.058375	-0.016809	0.170810	0.364149	0
P1_WITH_TERMS	0.993960	0.999927	0.907898	1.000000	0.118590	0.000225	0.204444	0.593000	0
MAX_INFORPRO	0.086213	0.120000	0.058375	0.118590	1.000000	0.395884	-0.044141	0.383927	0

5 rows × 23 columns

Figure 7. Matriz de correlação

A partir da matiz de correlação R calculamos seus autovalores e autovetores.

```
In [273]: eigen_val, eigen_vect =np.linalg.eig(R)
```

Ordena os autovalores em ordem decrescente

```
In [274]: idx = np.argsort(eigen_val)
          eigen_vect = eigen_vect.T[idx][::-1]
          eigen_val = eigen_val[idx][::-1]
```

Figure 8. Autovalores e autovetores

Os componentes principais Y são determinado pela combinação das variáveis normalizadas Z ponderada pelos componentes dos autovalores $\mathbf{v}$.

```
In [278]: # Y1 = np.dot(eigen_vect[0,:],Z.T)
          # Y2 = np.dot(eigen_vect[1,:],Z.T)
          Y = np.dot(eigen_vect,Z.T)
```

Figure 9. Calculo PCA

Uma propriedade dos componentes principais que podemos utilizar para validar os cálculos é comparar a variância dos componentes principais com os autovalores, que devem ser o mesmo.

```
In [279]: Y.var(axis=1)

Out[279]: array([6.81890840e+00, 1.97901992e+00, 1.68196300e+00, 1.61943032e+00,
                 1.24450473e+00, 1.12518716e+00, 1.07301811e+00, 1.04660266e+00,
                 1.01519298e+00, 9.49072972e-01, 9.01575915e-01, 7.33723096e-01,
                 7.02620199e-01, 6.75001591e-01, 4.31905707e-01, 3.76537748e-01,
                 3.34130055e-01, 2.01067221e-01, 4.39934821e-02, 2.61357719e-02,
                 1.92020404e-03, 1.63960420e-28, 1.04762185e-26])

In [280]: eigen_val

Out[280]: array([ 6.82439425e+00,  1.98061206e+00,  1.68331615e+00,  1.62073316e+00,
                  1.24550594e+00,  1.12609238e+00,  1.07388136e+00,  1.04744466e+00,
                  1.01600971e+00,  9.49836506e-01,  9.02301238e-01,  7.34313380e-01,
                  7.03185461e-01,  6.75544633e-01,  4.32253177e-01,  3.76840674e-01,
                  3.34398864e-01,  2.01228980e-01,  4.40288750e-02,  2.61567982e-02,
                  1.92174885e-03,  2.35572914e-15, -8.77708438e-16])
```

Figure 10. Variância de Y_i é igual ao autovalor λ_i

Vamos calcular agora a contribuição que cada componente principal tem sobre a variância total.

```
In [281]: C = eigen_val*100/eigen_val.sum()

In [282]: index = 1
          for i in C:
                  print("Contribuição C%s do componente Y%s: %5.2f" %(str(index),str(index),i))
                  index=index + 1

Contribuição C1 do componente Y1: 29.67
Contribuição C2 do componente Y2:  8.61
Contribuição C3 do componente Y3:  7.32
Contribuição C4 do componente Y4:  7.05
Contribuição C5 do componente Y5:  5.42
Contribuição C6 do componente Y6:  4.90
Contribuição C7 do componente Y7:  4.67
Contribuição C8 do componente Y8:  4.55
Contribuição C9 do componente Y9:  4.42
Contribuição C10 do componente Y10:  4.13
Contribuição C11 do componente Y11:  3.92
Contribuição C12 do componente Y12:  3.19
Contribuição C13 do componente Y13:  3.06
Contribuição C14 do componente Y14:  2.94
Contribuição C15 do componente Y15:  1.88
Contribuição C16 do componente Y16:  1.64
Contribuição C17 do componente Y17:  1.45
Contribuição C18 do componente Y18:  0.87
Contribuição C19 do componente Y19:  0.19
Contribuição C20 do componente Y20:  0.11
Contribuição C21 do componente Y21:  0.01
Contribuição C22 do componente Y22:  0.00
Contribuição C23 do componente Y23: -0.00
```

Figure 11. Contribuição de cada componente principal

A soma dos primeiros k autovalores representa a proporção de informação retida na redução de p para k dimensões. Nesse caso vamos considerar uma redução para 70% da informação.

```
In [283]: index = 1
          acumulador = 0
          while acumulador<70:
              acumulador = acumulador + C[index]
              index = index + 1
          print("Foi possivel reter %5.2f%% da informação reduzindo de %d para %d dimensões" %(acumulador, len(C),index))

          Foi possivel reter 70.02% da informação reduzindo de 23 para 18 dimensões
```

Figure 12. Retenção de 70% da informação original

Como a saída do comando acima mostra, diminuindo de 23 para 18 dimensões foi possível reter 70% da informação.

A influência que cada variável X tem sobre o componente Y é dada pelo seu peso w.

```
In [284]: W = pd.DataFrame()
          for i in range(len(eigen_vect)):
              W[i] = eigen_vect[i,:]/Z.std()

In [285]: W.T.head()
```

Out[285]:

	QTY_KG	P1	REALIZED_PAYMENT_TERMS	P1_WITH_TERMS	MAX_INFORPRO	UNUSED_INFORPRO	P2	MAX_D2	D2	SP	...	F
0	0.375158	0.376973	0.347684	0.377152	0.067152	0.013553	0.047402	0.253441	0.205738	0.001551	...	
1	0.101404	0.074137	0.150112	0.076465	-0.497990	-0.296748	0.252026	-0.364542	-0.280242	0.101105	...	
2	0.038346	0.038151	0.030044	0.038025	0.005630	0.164365	0.058676	-0.047619	-0.165280	-0.611170	...	
3	0.027297	0.009702	-0.009750	0.009182	0.136797	0.522319	-0.098675	-0.038702	-0.341008	0.259671	...	
4	-0.057267	-0.055870	-0.017904	-0.054953	-0.057026	0.144845	-0.288671	0.169271	0.251528	0.012784	...	

5 rows × 23 columns

Neste contexto a análise termina aqui, pois neste caso estamos interessados na obtenção de indices.

Figure 13. Influência que cada variável X_j tem sobre o componente Y_i

7.2. *PCA aplicado na redução de imagens*

Neste exemplo também vamos aplicar PCA para redução de dimensionalidade, mas neste caso com a finalidade de analisar um grande conjunto imagens de rosto (datasetfaces.mat). O conjunto de dados X é composto por 5000 imagens, cada 32 x 32 em escala de cinza. Cada linha de X corresponde a uma imagem do rosto (um vetor linha de tamanho 1024).

Para nossas análises vamos utilizar o Octave 4.2.1. Diferente do Python, no Octave/Matlab não precisamos carregar nenhuma biblioteca adicional para termos acesso aos métodos e funções de Algebra Linear, esses recursos são carregados nativamente ao iniciar a aplicação.

Primeiramento vamos carregar nosso conjunto de dados contendo os vetores das imagens e apresentar através da função "displayData".

```
%% Inicialização
clear ; close all; clc

%  carrega conjunto de dados de imagens
load ('datasetimages.mat')

% Utiliza a funcao displayData para apresentar o vetor de imagens
displayData(X(1:100, :));

```

Figure 14. Leitura do conjunto de dados

Figure 15. 100 primeiras imagens

O próximo passo é a normalização da matriz de dados X que vamos chamar de Z.

```
%  normalização
mu = mean(X);
X_norm = bsxfun(@minus, X, mu);

sigma = std(X_norm);
Z = bsxfun(@rdivide, X_norm, sigma);

```

Figure 16. Normalização da matriz de dados

Com a matriz normalizada Z vamos calcular sua matriz de correlação R e encontrar os autovalores e autovetores da matriz de correlação.

```
%  Rondando o PCA
[m, n] = size(Z);

U = zeros(n);
S = zeros(n);

R = (1/m)*Z'*Z;
[U, S, V] = svd(R);


%  Visualizacao dos 36 top autovetores encontrados
displayData(U(:, 1:36)');

```

Figure 17. Cálculo da matriz de correlaçao e autovalores/autovetores

A figura abaixo apresenta os 36 top autovetores.

Figure 18. Visualizacao dos 36 top autovetores encontrados

Projeção das imagens para o espaço próprio usando os k=100 vetores principais.

```
% Projeção das imagens para o espaço próprio usando os k=100 vetores principais
K = 100;
Z = zeros(size(X, 1), K);
U_reduce = U(:,1:K);
Z = X*U_reduce;


```

Figure 19. Projeção nos 100 primeiros autovetores

Após projeção dos dados nos 100 autovetores principais vamos recuperar uma aproximação dos dados originais ao usar os dados projetados.

```
% Visualizacao usando somento K=100 dimensoes
K = 100;
X_rec = zeros(size(Z, 1), size(U, 1));

for i=1:size(Z,1)
  v = Z(i, :)';
  for j=1:size(U(:, 1:K),1)
    X_rec(i,j)= v' * U(j, 1:K)';
  end
end

```

Figure 20. Aproximação dos dados originais apartir dos dados projetados

Com a matriz de dados recuperados vamos fazer uma comparação com os dados originais.

```
% Apresenta os dados
subplot(1, 2, 1);
displayData(X(1:100,:));
title('Faces Originais');
axis square;

% Apresenta os dados reconstruido
subplot(1, 2, 2);
displayData(X_rec(1:100,:));
title('Faces Recuperada');
axis square;


```

Figure 21. Código para comparação das imagens

Figure 22. Originais vs. Recuperadas

8. Conclusão

Uma das principais consequências de aplicação dos componentes principais é a compressão dos dados, reduzindo assim a quantidade de memória necessária para processar esses dados, menos dados precisam ser armazenados e aumento na velocidade em algoritmos de aprendizado. Nos casos que analisamos, esse foi o ganho obtido: menos variáveis precisam ser consideradas na analise de uma transação em princing, e menos informação precisa ser armazenadas para as imagens comprimidas.

Outra motivação comum na aplicação de PCA é a redução de dimensionalidade para melhor visualizar os dados. Como no exemplo que demos os dados em 2D foram reduzidos para uma representação 1D.

References

[1] Kotler P, K. K. (2006). *Administração de Marketing: A Biblia do Marketing.* Livraria Cultura.

[2] REGAZZI, A. (2000). *Análise multivariada, notas de aula INF 766.* Viçosa, Brasil: Universidade Federal de Viçosa.