TC 01 (2017.2) - Reconhecimento de Padrões

Polycarpo Souza Neto - 401658[1]

[1]Mestrado em Engenharia de Teleinformática - PPGETI - UFC

Questão 01

Estimar a matriz de covariância GLOBAL (i.e. sem considerar os rótulos das classes) para o conjunto de dados escolhido usando os métodos descritos nas Eqs. (68), (69), (70) e (73).Comparar com o resultado produzido pelo comando cov do Matlab/Octave.

Solução

Primeiro temos que inicializar o conjunto de dados e tirarmos suas médias.O conjunto de dados usado foi o DERMATOLOGIA[1]

```
load 'patologias.txt'
dados=patologias; % X[p N]
[p N]=size(dados);
m=mean(dados')';
```

Depois disso, implementamos os métodos de estimação da matriz de covariânica. A matriz de covariância pode ser ainda estimada por meio da seguinte expressão:

$$\hat{C}_x = \frac{1}{N}\sum_{i=1}^{N}[x(i) - \bar{x}][x(i) - \bar{x}]^T \tag{1}$$

o resultado da implementação desta equação é visto na matriz a seguir:

$$C_1 = \begin{bmatrix} 0.2139 & -0.0520 & -0.0615 & -0.0416 & -0.0416 & -0.0173 \\ -0.0520 & 0.1395 & -0.0332 & -0.0225 & -0.0225 & -0.0094 \\ -0.0615 & -0.0332 & 0.1590 & -0.0266 & -0.0266 & -0.0111 \\ -0.0416 & -0.0225 & -0.0266 & 0.1161 & -0.0180 & -0.0075 \\ -0.0416 & -0.0225 & -0.0266 & -0.0180 & 0.1161 & -0.0075 \\ -0.0173 & -0.0094 & -0.0111 & -0.0075 & -0.0075 & 0.0527 \end{bmatrix} \tag{2}$$

```
% equacao 68 - for grande
tic
soma=zeros(p);
for i=1:N,
soma=soma+(dados(:,i)-m)*(dados(:,i)-m)';
end
Cx_1=soma/(N);
toc
```

Uma outra forma de estimar o C_x é utilizando a Eq.3, que como pode ser visto tem notação matemática simples, o que implica numa economia de tempo na execução do método.

$$\hat{C}_x = \hat{R}_x - \bar{x}\bar{x}^T, \tag{3}$$

O resultado da implementação deste equação é dado por:

$$C_2 = \begin{bmatrix} 0.2139 & -0.0520 & -0.0615 & -0.0416 & -0.0416 & -0.0173 \\ -0.0520 & 0.1395 & -0.0332 & -0.0225 & -0.0225 & -0.0094 \\ -0.0615 & -0.0332 & 0.1590 & -0.0266 & -0.0266 & -0.0111 \\ -0.0416 & -0.0225 & -0.0266 & 0.1161 & -0.0180 & -0.0075 \\ -0.0416 & -0.0225 & -0.0266 & -0.0180 & 0.1161 & -0.0075 \\ -0.0173 & -0.0094 & -0.0111 & -0.0075 & -0.0075 & 0.0527 \end{bmatrix} \tag{4}$$

```
% equacao 69 - economico
tic
Rx=(1/N)*dados*dados';
Cx_2=Rx-(m*m');
toc
```

O próximo método implementado é marcado pela replicação da matriz de médias, sendo necessário o uso da função *repmat*, esta equação é vista em 5.

$$\hat{C}_x = \frac{1}{N}[X - M][X - M]^T, \\ M = [m|m|...|m] \tag{5}$$

A matriz de covariância estimada é dada por:

$$C_3 = \begin{bmatrix} 0.2139 & -0.0520 & -0.0615 & -0.0416 & -0.0416 & -0.0173 \\ -0.0520 & 0.1395 & -0.0332 & -0.0225 & -0.0225 & -0.0094 \\ -0.0615 & -0.0332 & 0.1590 & -0.0266 & -0.0266 & -0.0111 \\ -0.0416 & -0.0225 & -0.0266 & 0.1161 & -0.0180 & -0.0075 \\ -0.0416 & -0.0225 & -0.0266 & -0.0180 & 0.1161 & -0.0075 \\ -0.0173 & -0.0094 & -0.0111 & -0.0075 & -0.0075 & 0.0527 \end{bmatrix} \tag{6}$$

```
% equacao 70- replicando amtriz de medias
tic
M=repmat(m,1,N);
Cx_3=(1/N)*(dados-M)*(dados-M)';
toc
```

Muitas vezes os vetores só vão está disponíveis de um em um, então podemos de forma recursiva estimá-los de forma sequencial. Onde o n será cada iteração, $x(n)$ é o vetor observado e o α está entre zero e 1, denotando uma constante, um *fator de esquecimento*.Esta equação permite estimar um vetor-protótipo como se fosse uma técnica de filtragem passa-baixa, ao invés de uma mera média aritimética [2]. A equação pode ser visto em 7:

$$\hat{R}_x(n) = \alpha\hat{R}_x(n-1) + (1-\alpha)x(n)x(n)^T \\ m(n) = \alpha * m(n-1) + (1-\alpha)x(n) \tag{7}$$

```
% recursivo - alpha
tic
mx=dados(:,1);
Rx=mx*mx';
for n=2:N
a=n/(n+1);
Rx=a*Rx+(1-a)*dados(:,n)*dados(:,n)';
mx=a*mx+(1-a)*dados(:,n);
end
Cx_4=Rx-(mx*mx');
toc
```

O resultado da aplicação de $\hat{R}_x$ e m estimados recursivamente na equação para obter $\hat{C}_x$, gera o resultado visto a seguir na matriz:

$$C_4 = \begin{bmatrix} 0.2136 & -0.0525 & -0.0611 & -0.0413 & -0.0413 & -0.0172 \\ -0.0525 & 0.1410 & -0.0336 & -0.0227 & -0.0227 & -0.0095 \\ -0.0611 & -0.0336 & 0.1587 & -0.0264 & -0.0264 & -0.0110 \\ -0.0413 & -0.0227 & -0.0264 & 0.1158 & -0.0179 & -0.0074 \\ -0.0413 & -0.0227 & -0.0264 & -0.0179 & 0.1158 & -0.0074 \\ -0.0172 & -0.0095 & -0.0110 & -0.0074 & -0.0074 & 0.0526 \end{bmatrix} \tag{8}$$

Como forma de comparação e verificação se as estimações acima estavam corretas, foi usado o comando *cov* do Matlab, que calcula a matriz de coavariância segunda a equação abaixo:

$$C_{(A,B)} = \frac{1}{N-1} \sum_{i=1}^{N} [A_i - \mu_A]^* [B_i - \mu_B] \tag{9}$$

onde μ_A e μ_B são as respectivas médias e * é o complexo do conjugado.

A matriz de covariância resolvida pelo *cov* foi:

$$C_{cov} = \begin{bmatrix} 0.2145 & -0.0521 & -0.0617 & -0.0417 & -0.0417 & -0.0174 \\ -0.0521 & 0.1399 & -0.0333 & -0.0225 & -0.0225 & -0.0094 \\ -0.0617 & -0.0333 & 0.1594 & -0.0267 & -0.0267 & -0.0111 \\ -0.0417 & -0.0225 & -0.0267 & 0.1164 & -0.0180 & -0.0075 \\ -0.0417 & -0.0225 & -0.0267 & -0.0180 & 0.1164 & -0.0075 \\ -0.0174 & -0.0094 & -0.0111 & -0.0075 & -0.0075 & 0.0529 \end{bmatrix} \tag{10}$$

```

%cov nativo do matlab
tic
Cx_5=cov(dados');
toc
```

Conclusão

Como podemos ver, as matrizes de covariância estimadas pelos 4 métodos, deram iguais, diferindo alguns valores da matriz obtida com o comando *cov* apenas na terceira

ou quarta casa decimal, o que acontece devido a divisão do somatório ser por $N-1$ (9), diferente dos outros 4 métodos. No entando, podemos dizer que o resultado foi igual e a estimação foi correta.

Questão 02

Comparar os métodos implementados no Item 1 com o comando cov do Matlab/Octave em termos de tempo de processamento. Para isso, usar os comandos tic/toc.

Solução

Usando as equações abaixo, foram estimadas 5 matrizes de covariância, pelos trechos de códigos já anexados no item 1, onde para cada equação foi usado um comando *tic-toc* para medição do tempo de estimação, como pode ser visto na Tabela1.

$$\hat{C}_x = \frac{1}{N}\sum_{i=1}^{N}[x(i)-\bar{x}][x(i)-\bar{x}]^T \tag{11}$$

$$\hat{C}_x = \hat{R}_x - \bar{x}\bar{x}^T, \tag{12}$$

$$\begin{aligned}\hat{C}_x &= \frac{1}{N}[X-M][X-M]^T,\\ M &= [m|m|...|m]\end{aligned} \tag{13}$$

$$\begin{aligned}\hat{R}_x(n) &= \alpha\hat{R}_x(n-1) + (1-\alpha)x(n)x(n)^T\\ m(n) &= \alpha * m(n-1) + (1-\alpha)x(n)\end{aligned} \tag{14}$$

Tabela 1. Comparação do tempo de estimação para cada método.

Tempo	For (11)	Econômico(12)	Repmat (13)	Recursivo(14)	Cov (Matlab)
(s)	0.003760	0.000937	0.001745	0.006055	0.009732

Pelo que pode ser visto, assim como já citado no item 1, o método chamado econômico, implementado pela Eq.12, foi o mais rápido, seguido do modelo de replicação matricial da equação (13). O que gerou maior custo computacional foi o método implementado pela função nativa *cov*. Os outros métodos que implementam a estimação da matriz de covariância e fazem uso de laços *for*, são os mais demorados, incluindo o método nativo.

Questão 03

Escolher um dos métodos implementados no Item 1 e estimar as matrizes de covariância de cada classe.

Solução

Utilizando o método da (15), método de notação econômica e mais rápido, implementamos a estimação da matriz de covariância de cada classe.

$$\hat{C}_x = \hat{R}_x - \bar{x}\bar{x}^T, \tag{15}$$

Para efeito de curiosidade, estima-se e mostra-se a matriz GLOBAL.

$$C_{cov} = \begin{bmatrix} 0.2145 & -0.0521 & -0.0617 & -0.0417 & -0.0417 & -0.0174 \\ -0.0521 & 0.1399 & -0.0333 & -0.0225 & -0.0225 & -0.0094 \\ -0.0617 & -0.0333 & 0.1594 & -0.0267 & -0.0267 & -0.0111 \\ -0.0417 & -0.0225 & -0.0267 & 0.1164 & -0.0180 & -0.0075 \\ -0.0417 & -0.0225 & -0.0267 & -0.0180 & 0.1164 & -0.0075 \\ -0.0174 & -0.0094 & -0.0111 & -0.0075 & -0.0075 & 0.0529 \end{bmatrix} \tag{16}$$

Para mostrar a matriz de covariância de cada classe, levamos em consideração o vetor de rótulos e pegamos todos os seus valores. O comando *unique* retorna os mesmos dados que tem na matriz, mas sem repetí-los. Com isso a gente vai ter o retorno dos dados em ordem de seus rótulos, no caso deste *dataset*, de 1 até 6. Depois, fazemos uma célula para fazer a separação em rótulos de 1 até o 6 e outra pra calcular a matriz de covariancia de cada classe. Devido ao conjunto de dados ter 34 atributos, sua matriz de covariância por classe vai ser $34x34$, o que torna inviável colocar neste documento, então, foi usado um comando para pegar só as 5 primeiras linhas e 5 primeiras colunas (1:5,1:5). Seguem as matriz estimadas abaixo:

$$C_{classe1} = \begin{bmatrix} 0.3854 & 0.1050 & 0.0769 & 0.0220 & -0.0146 \\ 0.1050 & 0.3931 & 0.1227 & 0.0179 & 0.0012 \\ 0.0769 & 0.1227 & 0.3307 & -0.0041 & 0.0170 \\ 0.0220 & 0.0179 & -0.0041 & 1.1782 & 0.5980 \\ -0.0146 & 0.0012 & 0.0170 & 0.5980 & 0.7597 \end{bmatrix} \tag{17}$$

$$C_{classe2} = \begin{bmatrix} 0.3697 & 0.0811 & 0.0475 & 0.0086 & -0.0094 \\ 0.0811 & 0.2956 & 0.0033 & 0.0756 & -0.0022 \\ 0.0475 & 0.0033 & 0.6475 & -0.4525 & 0.0017 \\ 0.0086 & 0.0756 & -0.4525 & 0.9031 & 0.0128 \\ -0.0094 & -0.0022 & 0.0017 & 0.0128 & 0.0656 \end{bmatrix} \tag{18}$$

$$C_{classe3} = \begin{bmatrix} 0.3591 & 0.1436 & 0.0903 & 0.0196 & -0.0286 \\ 0.1436 & 0.4293 & 0.2333 & 0.1262 & 0.0674 \\ 0.0903 & 0.2333 & 0.4551 & 0.1708 & 0.0934 \\ 0.0196 & 0.1262 & 0.1708 & 0.6185 & 0.2617 \\ -0.0286 & 0.0674 & 0.0934 & 0.2617 & 1.0970 \end{bmatrix} \tag{19}$$

$$C_{classe4} = \begin{bmatrix} 0.3433 & 0.0751 & 0.0425 & 0.0082 & -0.0868 \\ 0.0751 & 0.2496 & 0.1415 & -0.0621 & -0.0868 \\ 0.0425 & 0.1415 & 0.4983 & -0.2040 & -0.1597 \\ 0.0082 & -0.0621 & -0.2040 & 0.5829 & 0.1910 \\ -0.0868 & -0.0868 & -0.1597 & 0.1910 & 0.6389 \end{bmatrix} \tag{20}$$

$$C_{classe5} = \begin{bmatrix} 0.4579 & 0.1324 & 0.1584 & -0.0590 & 0 \\ 0.1324 & 0.4162 & 0.1610 & -0.1007 & 0 \\ 0.1584 & 0.1610 & 0.8016 & -0.0590 & 0 \\ -0.0590 & -0.1007 & -0.0590 & 1.0972 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{21}$$

$$C_{classe6} = \begin{bmatrix} 0.2475 & 0.1125 & -0.0525 & -0.0250 & 0 \\ 0.1125 & 0.1875 & -0.0375 & -0.0750 & 0 \\ -0.0525 & -0.0375 & 0.5475 & -0.1250 & 0 \\ -0.0250 & -0.0750 & -0.1250 & 0.3500 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{22}$$

As matrizes de covariância das 6 classes deste conjunto de dados foram anexados no *.zip* deste trabalho e são compartilhadas via Google Drive à partir do link:`https://drive.google.com/open?id=1RVRhVBeDRAoEs-nPV_2E-ZmRZwkIFR-3`.

```
%lendo conjunto de dados
load 'patologias.txt'
load 'pacientes.txt'
%arquivo com os rotulos
labels= patologias;
%diagnosticos
diag=pacientes;
[p N]=size(diag);
%encontrar os rotulos
labels_tot=zeros(1,N);
for i=1:N
labels_tot(i)=find(patologias(:,i)==max(patologias(:,i)));
end
%vetor de rotulos sem repeticao
vet_labels=unique(labels_tot);
%tamanho e quantidade de rotulos
num_labels=length(vet_labels);
%separar as classes por rotulos
separate_class=cell(1,num_labels);
for i=1:num_labels
index=find(labels_tot==vet_labels(i));
separate_class{i}=diag(:,index);
end
%calcular matriz pra cada classe
Matrix_cov_class=cell(1,num_labels)
%calcualr posto por classe
find_rank=cell(1,num_labels);
%encontrar numero de condicionamento por classe
find_number_condit=cell(1,num_labels);
for i=1:num_labels
[~,Ni]=size(separate_class{i});
%media dos dados por classe
m_class=mean(separate_class{i}')';
Matrix_CorE_est=(1/Ni)*separate_class{i}*separate_class{i}'
%calcula matriz de covariancia
Matrix_cov_class{i}=Matrix_CorE_est-(m_class*m_class');
```

Questão 04

Avaliar a invertibilidade da matriz de covariância global e as de cada classe através do seu posto e do seu número de condicionamento. Usar comandos rank e cond.

Solução

Uma matriz possui posto k se e somente se tem k linhas e k colunas linearmente independentes, enquanto cada uma das linhas restantes e colunas é uma combinação linear dos k precedentes. Suponhamos que a matriz seja:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{bmatrix} \tag{23}$$

e suponhamos que o menor:

$$\lambda = \begin{vmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,k} \\ a_{2,1} & a_{2,2} & \dots & a_{2,k} \\ \dots & \dots & \dots & \dots \\ a_{k,1} & a_{k,2} & \dots & a_{k,n} \end{vmatrix} \tag{24}$$

formado pelas primeiras linhas k e as primeiras colunas k situação que sempre é possível obter por meio de substituições adequadas entre linhas e colunas não é nula. Deve notar-se que as primeiras linhas *k* são linearmente independentes. Na verdade, se o contrário for verdadeiro, pelo menos uma das linhas seria uma combinação linear dos restantes e (24) seria nulo contra a hipótese.

Lema 1

Subespaços $[x_1, x_2, ...x_r]$ e $[y_1, y_2, ...y_r]$ são iguais se cada $x_1, x_2, ...x_r$ for uma combinação linear de $y_1, y_2, ...y_r$. Similar $d_1y_1 + ... + d_sy_s$, onde y é uma combinação linear de $x_1, x_2, ...x_r$. Isso mostra a iguadade de dois subespaços.

Tendo conhecimento do conceito de posto, usamos a função *rank* do Matlab para determinar o posto da matriz GLOBAL de covariância, bem como o posto de cada matriz de covariância por classe. Como pode ser visto na Tabela2, nenhumas destas matrizes por classe é de posto completo. Devido a isto essas matrizes não são invertíveis e não podem ser consideradas bases.

Tabela 2. Posto das classes.

Classes	1	2	3	4	5	6	Global
Size (34x)	111	60	71	48	48	20	34
Posto	25	23	26	16	18	19	34

O condicionamento de um sistema linear é um conceito relacionado à forma como os erros se propagam dos dados de entrada para os dados de saída[3]. No contexto de um sistema linear $Ax = y$, temos que a solução x depende dos dados de entrada y. Consideremos, então, o problema:

$$A(x + \delta_x) = y + \delta_y \tag{25}$$

Aqui, δ_x representa uma variação (erro) em x e δ_y representa uma variação em y (erro). Temos:

$$Ax + A\delta_x = y + \delta_y \tag{26}$$

e, portanto,

$$A\delta_x = \delta_y \tag{27}$$

Queremos avaliar a razão entre o erro relativo em x e o erro relativo em y, isto é[4]

$$\frac{\parallel \delta_x \parallel}{\parallel x \parallel} \frac{\parallel \delta_y \parallel}{\parallel y \parallel} \tag{28}$$

$$\begin{aligned} \frac{\parallel \delta_x \parallel / \parallel x \parallel}{\parallel \delta_y \parallel / \parallel y \parallel} &= \frac{\parallel \delta_x \parallel}{\parallel x \parallel} \frac{\parallel y \parallel}{\parallel \delta_y \parallel} \\ &= \frac{\parallel A^{-1}\delta_x \parallel}{\parallel x \parallel} \frac{\parallel Ay \parallel}{\parallel \delta_y \parallel} \\ &\leq \frac{\parallel A^{-1} \parallel \parallel \delta_x \parallel}{\parallel x \parallel} \frac{\parallel A \parallel \parallel y \parallel}{\parallel \delta_y \parallel} \\ &= \parallel A \parallel \parallel A^{-}1 \parallel \end{aligned} \tag{29}$$

Na hora de calcular o número de condicionamento, precisamos ter noção do conceito de norma e que o uso de uma norma diferente, resulta num diferente número de condicionamento. São conhecidas algumas normas e estas são vistas a seguir[3].

Essa equação é a que resulta no cálculo da norma Euclidiana, onde esse valor de μ_{max} é o maior autovalor de $[A]^T[A]$. Essa norma é a norma mínima e fornece um valor mais justo. Essa norma é justamente a *default* do Matlab dentro da função *cond*, que resulta no número de condicionamento de uma matriz[4].

$$\parallel A \parallel_2 = (\mu_{max})^{1/2} \tag{30}$$

O número de condicionamento de cada matriz de covariância das classes dos dados de dermatologia obtidos com o comando *cond*, usando a norma Euclidiana foi:

Tabela 3. Condicionamento das classes utilizando norma euclidiana.

Classes	1	2	3	4	5	6	Global
Cond	$9.297*10^{50}$	$1.553*10^{51}$	$6.155*10^{35}$	∞	$1.799*10^{67}$	$8.869*10^{49}$	$1.778*10^{4}$

Essa norma é chamada norma das colunas, pois, é feita a soma dos valoresabsolutos dos coeficientes para cada coluna e a maior dessas somas é usada como norma. Pode ser definida segunda a equação abaixo [4]:

$$\parallel A \parallel_1 = \max_{1 \leq k \leq n} \sum_{i=1}^{m} |a_{ik}| \tag{31}$$

O número de condicionamento de cada matriz de covariância das classes dos dados de dermatologia obtidos com a norma-1 foi:

Tabela 4. Condicionamento das classes utilizando a norma-1.

Classes	1	2	3	4	5	6
Cond	inf	inf	inf	inf	inf	inf

Semelhante a norma-1 temos a norma infinita ou norma das linhas, sendo que o cálculo feito é nas linhas e não nas colunas [4]:

$$\| A \|_{\infty} = \max_{1 \leq i \leq n} \sum_{j=1}^{m} |a_{ij}| \tag{32}$$

O número de condicionamento de cada matriz de covariância das classes dos dados de dermatologia obtidos com a norma-∞ foi:

Tabela 5. Condicionamento das classes utilizando a norma∞.

Classes	1	2	3	4	5	6
Cond	inf	inf	inf	inf	inf	inf

Uma das normas de matriz mais antigas e mais simples é a norma de Frobenius, às vezes chamada de norma Hilbert-Schmidt. É definido como a raiz quadrada da soma dos quadrados de todas as entradas da matriz [3], ou

$$\| A \|_{F} = (\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^{2})^{1/2} \tag{33}$$

O número de condicionamento de cada matriz de covariância das classes dos dados de dermatologia obtidos com a norma de Frobenius foi:

Tabela 6. Condicionamento das classes utilizando a norma de Frobenius.

Classes	1	2	3	4	5	6
Cond	inf	inf	inf	inf	inf	inf

O *rcond* retorna o número de condicionamento recíproco, retornado como escalar. O número de condicionamento recíproco é uma medida invariante da escala de quão próxima uma matriz dada é para o conjunto de matrizes singulares[5].

- Se *rcond* está perto de 0, a matriz é quase singular e mal condicionada;
- Se *rcond* é próximo de 1, a matriz está bem condicionada.

O número de condicionamento de cada matriz de covariância das classes dos dados de dermatologia obtidos com o comando $rcond$ foi:

Tabela 7. Condicionamento das classes utilizando o rond.

Classes	1	2	3	4	5	6
Rcond	0	0	0	0	0	0

Logo, podemos dizer que a matrizes de covariância de cada classe são mal-condicionadas.

Pequenas variações nos coeficientes das matrizes fazem as soluções ficarem bem distintas, isto é, pequenas variações nos dados de entrada geram grandes variações na solução do sistema. Quando isso acontece, dizemos que o problema é mal-condicionado, que é justamente o que acontece conosco aqui, onde os valores dos números de condicionamento para norma Euclidiana começam da ordem de $cond[A](O^{49})$, enquanto para outras normas o resultado é infinito (∞), além de dar 0 quando usamos o comando recíproco (*rcond*).

Código

```
%tamanho de cada classe
%necessario para verificar posto
size_each_class=size(separate_class{i});
%posto de cada classe
find_rank{i}=rank(Matrix_cov_class{i});
%numero de condicionamento
%norm-2 default
find_number_condit{i}=cond(Matrix_cov_class{i});
%normas p genericas
%norma de Frobenius
find_number_condit_fro{i}=cond(Matrix_cov_class{i},'fro');
%norma 1
find_number_condit_hum{i}=cond(Matrix_cov_class{i},1);
%norma infinita
find_number_condit_inf{i}=cond(Matrix_cov_class{i},'inf');
find_number_rcondit{i}=rcond(Matrix_cov_class{i});

end
%pega tamanho de cada classe 34xn
size_each_class1=size(separate_class{1});
size_each_class2=size(separate_class{2});
size_each_class3=size(separate_class{3});
size_each_class4=size(separate_class{4});
size_each_class5=size(separate_class{5});
size_each_class6=size(separate_class{6});

%tabela com resultados
name={'TAMANHO - CLASSE'};
T = table(size_each_class1,size_each_class2,size_each_class3,
   size_each_class4,size_each_class5,size_each_class6,...
'RowNames',name)

%=====================================
find_number_rcondit1=rcond(Matrix_cov_class{1});
find_number_rcondit2=rcond(Matrix_cov_class{2});
find_number_rcondit3=rcond(Matrix_cov_class{3});
find_number_rcondit4=rcond(Matrix_cov_class{4});
```

```
find_number_rcondit5=rcond(Matrix_cov_class{5});
find_number_rcondit6=rcond(Matrix_cov_class{6});

name={'RCOND - CLASSE'};
T = table(find_number_rcondit1,find_number_rcondit2,
   find_number_rcondit3,find_number_rcondit4,
   find_number_rcondit5,find_number_rcondit6,...
'RowNames',name)
%=====================================

find_number_condit1=cond(Matrix_cov_class{1});
find_number_condit2=cond(Matrix_cov_class{2});
find_number_condit3=cond(Matrix_cov_class{3});
find_number_condit4=cond(Matrix_cov_class{4});
find_number_condit5=cond(Matrix_cov_class{5});
find_number_condit6=cond(Matrix_cov_class{6});

name={'COND - CLASSE'};
T = table(find_number_condit1,find_number_condit2,
   find_number_condit3,find_number_condit4,find_number_condit5,
   find_number_condit6,...
'RowNames',name)
```

Referências

[1] DERMATOLOGIA, https://archive.ics.uci.edu/ml/datasets/Dermatology, Acesso em 30 de novembro de 2017.

[2] BARRETO, Guilherme A. Introdução à Classificação de Padrões. Slides, Fortaleza, 2017.

[3] FORD, William. Numerical linear algebra with applications: Using MATLAB, Chapter 13, 13.2. Symmetric positive definite matrices. Academic Press, 2014.

[4] CHAPRA, Steven C. Métodos numéricos para ingenieros. McGraw-Hill,, 2007.

[5] Mathworks, rcond, (2017), disponível em https://www.mathworks.com/help/matlab/ref/rcond.html, Acesso em 3 de dezembro de 2017.