

BORIM, J. C. **Modelagem e Controle de um Processo de Endurecimento de Pelotas de Minério de Ferro**. Dissertação de Mestrado – Faculdade de Engenharia Elétrica, Universidade Federal de Minas Gerais, Minas Gerais, 2000.

KS400, version 3.0. Carl Zeiss Vision, 1998.

CENTRO DE GESTÃO E ESTUDOS ESTRATÉGICOS – CGEE. **Siderurgia no Brasil 2010-2025**. Série Documentos Técnicos, n. 9, 2010.

COSTA, R. V. P. **Otimização da Resistência à Compressão de Pelotas de Minério de Ferro para Redução Direta pela Aplicação de Projeto Robusto**. Dissertação de Mestrado – Rede Temática em Engenharia de Materiais, Universidade Federal de Ouro Preto, Minas Gerais, 2008.

DEPARTAMENTO NACIONAL DE PRODUÇÃO MINERAL – DNPM. **Sumário Mineral 2009**. v. 29. 2010.

DUDA, R. O.; HART, P. E.; STORK, D. G. **Patter Classification**. 2<sup>nd</sup> edition. Wiley-Interscience. 2001.

FEITOSA, R. **Notas de Aula**. Disciplina de Visão Computacional. [200-?].

FRIEL, J. J. Measurements. In: ASM International. **Practical Guide to Image Analysis**, p. 101-128. Materials Park – OH, 2000.

GOMES, O. F. M. **Microscopia Co-Localizada: Novas Possibilidades na Caracterização de Minérios**. Tese de Doutorado – Departamento de Ciência de Materiais e Metalurgia, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2007.

GOMES, O. F. M. **Processamento e Análise de Imagens Aplicados à Caracterização Automática de Materiais**. Dissertação de Mestrado – Departamento de Ciência de Materiais e Metalurgia, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2001.

GONZALEZ, R. C.; WOODS, R.E. **Digital Image Processing**. 2<sup>nd</sup> edition. Prentice- Hall. Upper Saddle River – NJ, 2002.

HARALICK, R. M.; SHANMUGAN, K.; DINSTEN, I. **Textural Features for Image Classification**. IEEE Transactions on Systems, Man and Cybernetics, v. SMC-3, n. 6, p. 610-621, 1973.

IGLESIAS, J. C. A. **Uma Metodologia para Caracterização de Sínter de Minério de Ferro: Microscopia Digital e Análise de Imagens**. Dissertação de Mestrado – Departamento de Ciência de Materiais e Metalurgia, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2008.

INSTITUTO BRASILEIRO DE MINERAÇÃO – IBRAM. **Informações e Análises da Economia Mineral Brasileira**. 6<sup>a</sup> edição. 2011.

LUZ, A. B.; LINS, F. A. F. **Introdução ao Tratamento de Minérios**. Comunicação Técnica elaborada para a 4<sup>a</sup> Edição do Livro de Tratamento de Minérios, p. 3-16, 2004.

LUZ, A. B.; et al. **Tratamento de Minérios**. CETEM. Rio de Janeiro – RJ, 2002.

MAATEM, L. V. D.; POSTMA, E. HERIK, J. V. D. **Dimensionality Reduction: A Comparative Review**. Tilburg University, 2009.

MAATEM, L. V. D. **Matlab Toolbox for Dimensionality Reduction**. Disponível em <[http://homepage.tudelft.nl/19j49/Matlab\\_Toolbox\\_for\\_Dimensionality\\_Reduction.html](http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html)>. Acesso em: 24 nov. 2011.

MACHADO, M. L. P.; SOBRINHO, V. P. F. M.; ARRIVABENE, L. F. **Siderurgia para Não Siderurgistas**. Vitória – ES, 2003.

MATLAB, version 7.6.0.324 (R2008a), 2008.

MATOS, A. P. **Influência da Temperatura, Pressão, Produção e Granulometria no Processo de Secagem das Pelotas Cruas**. Dissertação de Mestrado – Rede Temática em Engenharia de Materiais, Universidade Federal de Ouro Preto, Minas Gerais, 2007.

MENDES, J. J. **Influência da Adição de Lama Fina de Aciaria a Oxigênio nas Características Físicas e Microestruturais de Pelotas Queimadas de Minério de Ferro**. Dissertação de Mestrado – Rede Temática em Engenharia de Materiais, Universidade Federal de Ouro Preto, Minas Gerais, 2009.

MEYER, K. **Pelletizing of Iron Ores**. Springer-Verlag. Berlin, Heidelberg, and Verlag Stahleisen mbH, Düsseldorf, 1980.

MOURÃO, M. B.; et al. **Introdução à Siderurgia**. ABM. São Paulo, 2007.

NUNES, J. E. F. **Controle de um Processo de Pelotização: Realimentação por Imagem**. Dissertação de Mestrado – Faculdade de Engenharia Elétrica, Universidade Federal de Minas Gerais, Minas Gerais, 2004.

PACIORNIK, S.; MAURÍCIO, M. H. P. **Digital Imaging**. In: Vander-Voort, G. F. (Ed.). ASM Handbook, Metallography and Microstructures. v. 9, p. 368-402. Materials Park – OH, 2004.

PACIORNIK, S. **Notas de Aula**. Disciplina de Microscopia Quantitativa. Departamento de Engenharia dos Materiais – PUC-Rio. 2010. Disponível em <[www.dema.puc-rio.br/cursos/micquant](http://www.dema.puc-rio.br/cursos/micquant)>. Acesso em: 22 out. 2011.

PACIORNIK, S. **Notas de Aula**. Disciplina de Processamento Digital de Imagens. Departamento de Engenharia dos Materiais – PUC-Rio. 2010. Disponível em: <[www.dema.puc-rio.br/cursos/ipdi](http://www.dema.puc-rio.br/cursos/ipdi)>. Acesso em: 22 out. 2011.

SILVA, F. D. **Redução do Delta de Resistência à Compressão entre Pelotas Produzidas e Embarcadas nas Usinas de Pelotização da Vale – Uma Análise de Causas**. Monografia de Pós-Graduação – Departamento de Engenharia de Minas, Universidade Federal de Ouro Preto, Minas Gerais, 2010.

SINDICATO NACIONAL DA INDÚSTRIA DA EXTRAÇÃO DO FERRO E METAIS BÁSICOS – SINFERBASE. **Relatório novembro – 2009**.

SINDICATO NACIONAL DA INDÚSTRIA DA EXTRAÇÃO DO FERRO E METAIS BÁSICOS – SINFERBASE. **Relatório novembro – 2010**.

SINDICATO NACIONAL DA INDÚSTRIA DA EXTRAÇÃO DO FERRO E METAIS BÁSICOS – SINFERBASE. **Relatório novembro – 2011**.

SRB, J.; RUZICKOVÁ, Z. **Pelletization of Fines**. Tchecoslováquia, 1988.

TAKEHARA, L. **Caracterização Geometalúrgica dos Principais Minérios de Ferro Brasileiros – Fração *Sinter Feed***. Tese de Doutorado – Instituto de Geociências, Universidade Federal do Rio Grande do Sul, Rio Grande do Sul, 2004.

VALE. **Technical Cooperation Agreement Formosa Ha Tinh STEEL Corporation – Vale**. Training Program - Pelletizing Process. Marketing and Research & Development Department – DITM. 2010.

VARGAS, G. B. **Orçamento Empresarial no Setor de Minério de Ferro no Brasil: Um Estudo de Caso na Samarco Mineração**. Dissertação de Mestrado – Faculdade de Administração, Faculdades Integradas de Pedro Leopoldo, Minas Gerais, 2009.

WAGNER, D. T.; ROUCO, H. V.; GOMES, O. F. M.; PACIORNIK, S.; VIEIRA, M. B. **Caracterização de Pelotas de Minério de Ferro por Microscopia Digital e Análise de Imagens**. Tecnologia em Metalurgia e Materiais, São Paulo, v. 5, n. 4, p. 215-218, 2009.

# 10 Anexo

## 10.1. Resultados

### 10.1.1. Seleção de Atributos por Busca Exaustiva da Melhor Taxa de Acerto

- Classificador *Mahalanobis*:

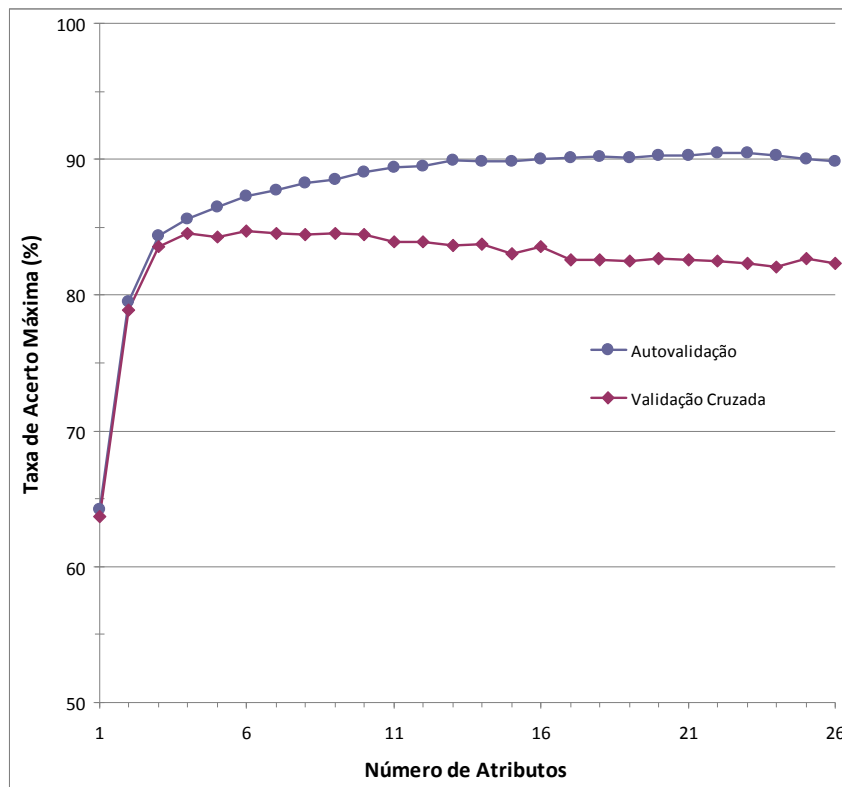


Figura 82 – Autovalidação e validação cruzada com o classificador *Mahalanobis*

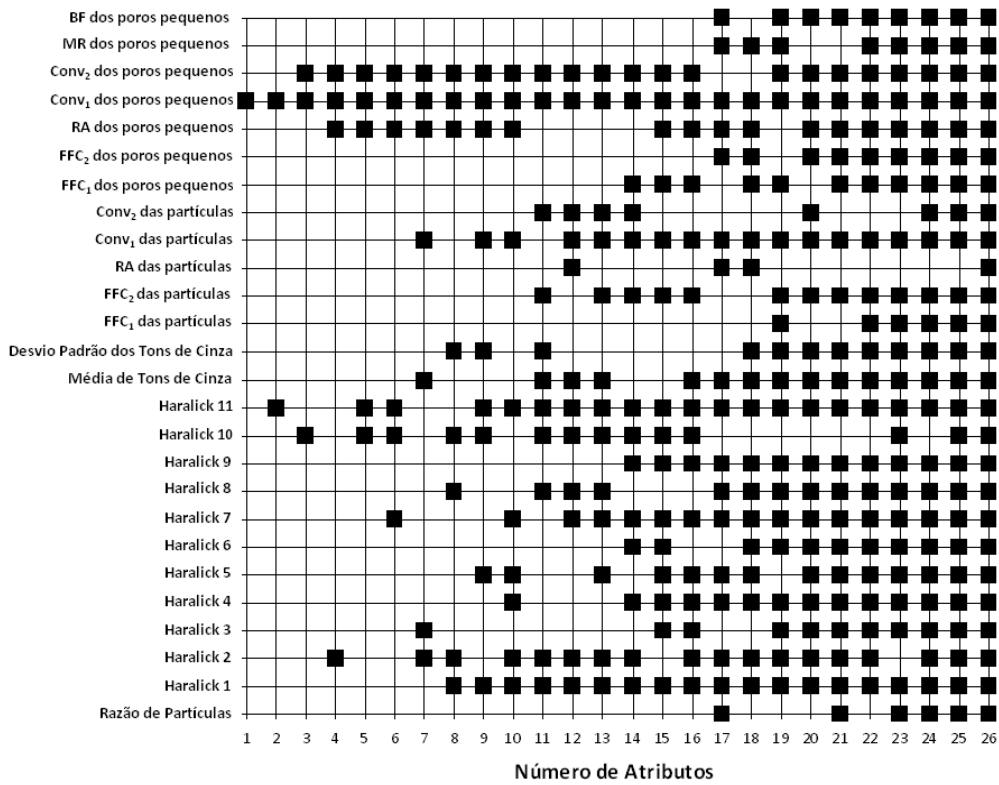


Figura 83 – Atributos selecionados para cada conjunto com a melhor taxa de acerto

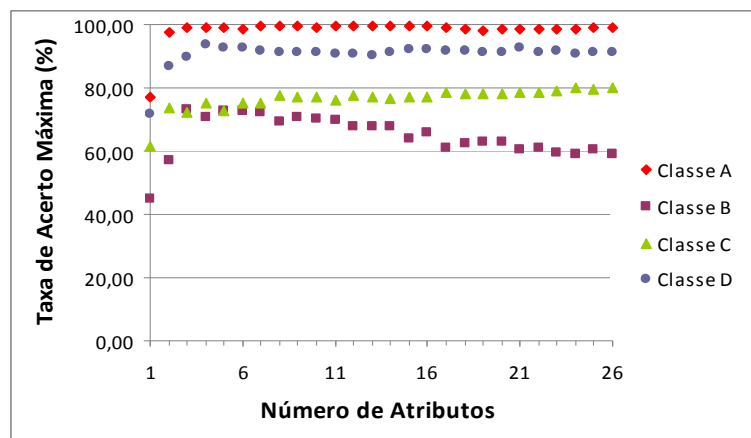


Figura 84 – Taxa de acerto de cada classe na validação cruzada com o classificador Mahalanobis

- Classificador *Linear*:

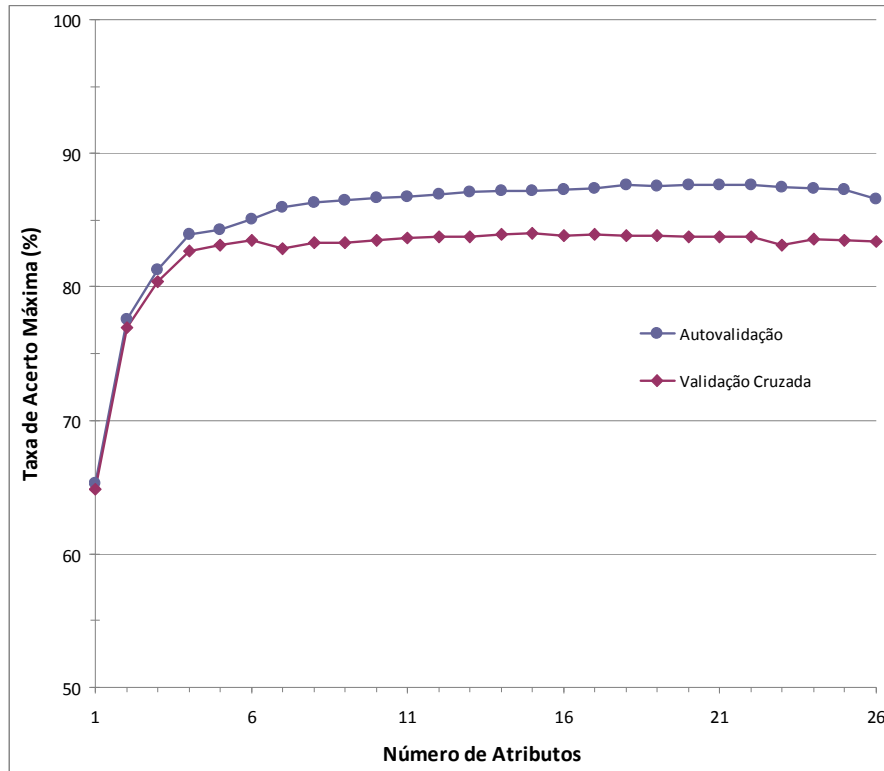


Figura 85 – Autovalidação e validação cruzada com o classificador *Linear*

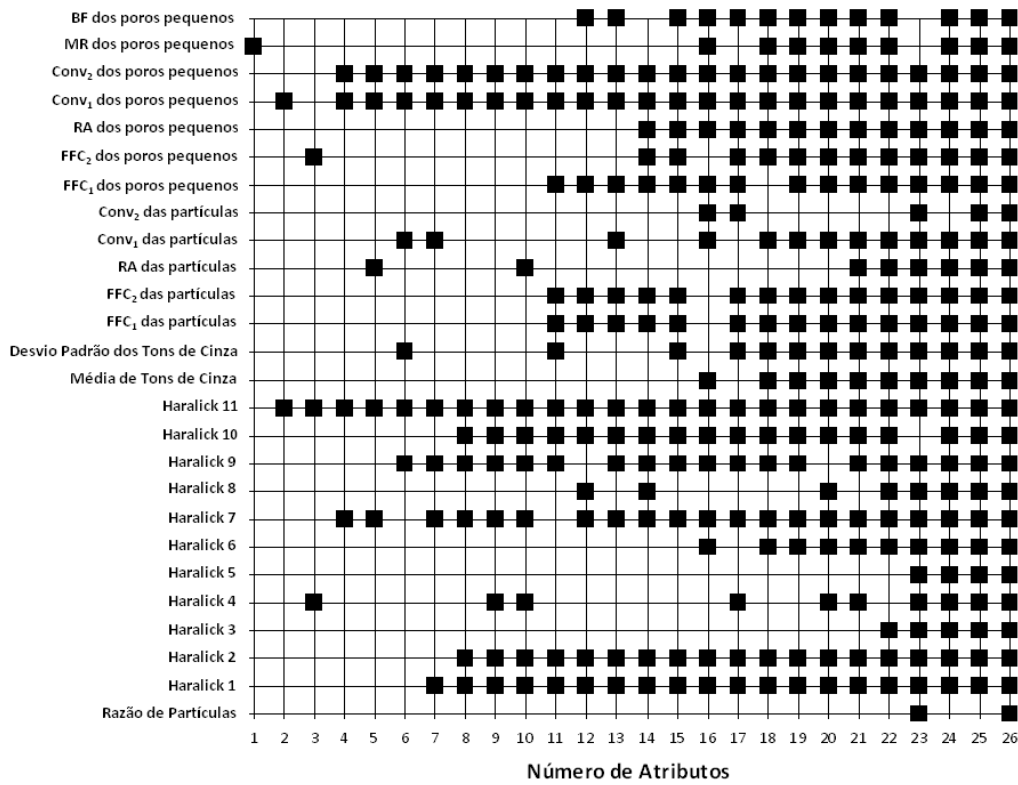


Figura 86 – Atributos selecionados para cada conjunto com a melhor taxa de acerto

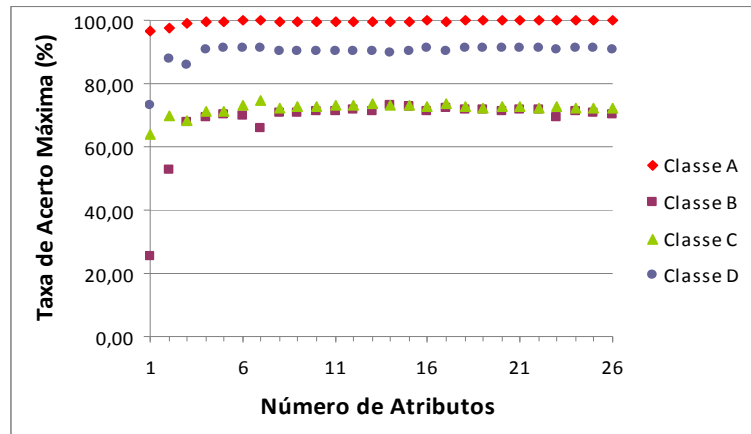


Figura 87 – Taxa de acerto de cada classe na validação cruzada com o classificador *Linear*

### 10.1.2. PCA

- Classificador *Quadrático*:

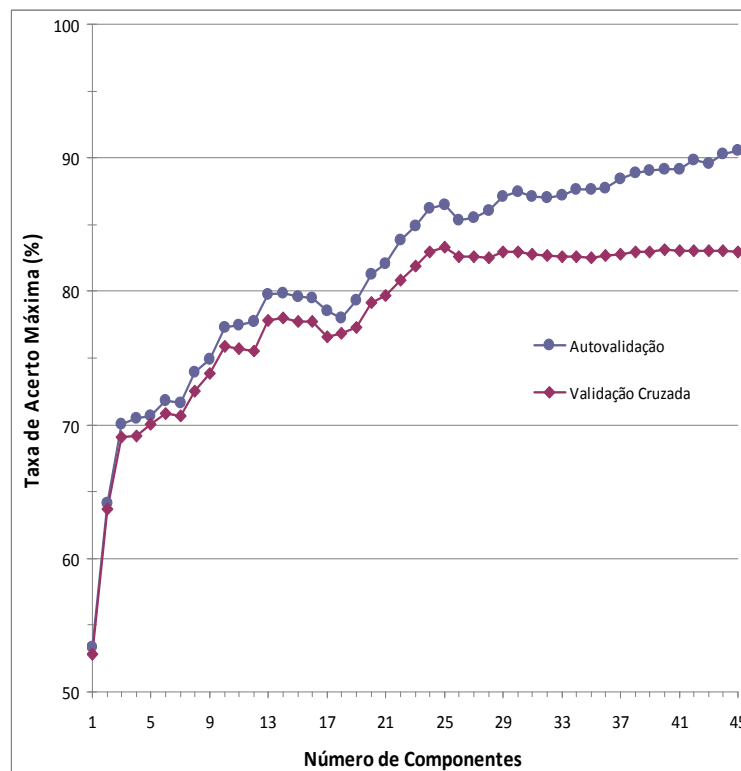


Figura 88 – Autovalidação e validação cruzada com o classificador *Quadrático*



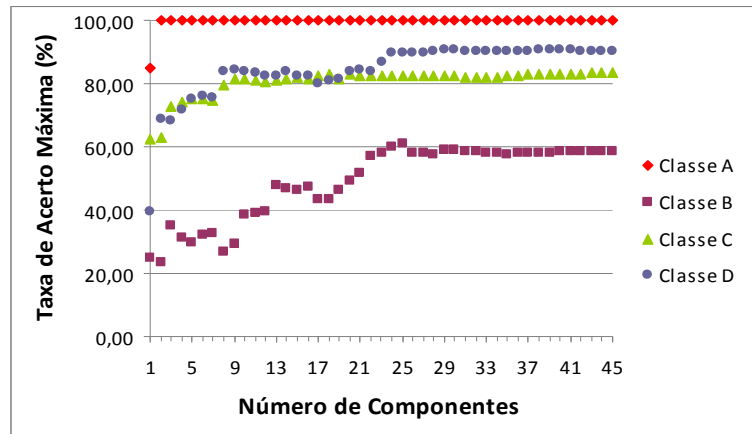


Figura 89 – Taxa de acerto de cada classe na validação cruzada com o classificador *Quadrático*

- Classificador *Linear*:

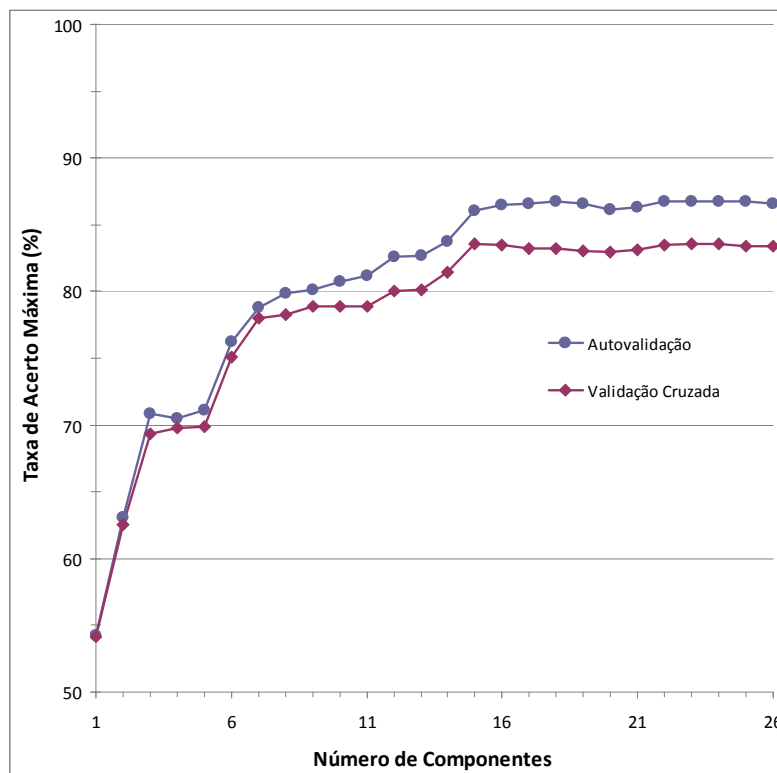


Figura 90 – Autovalidação e validação cruzada com o classificador *Linear*

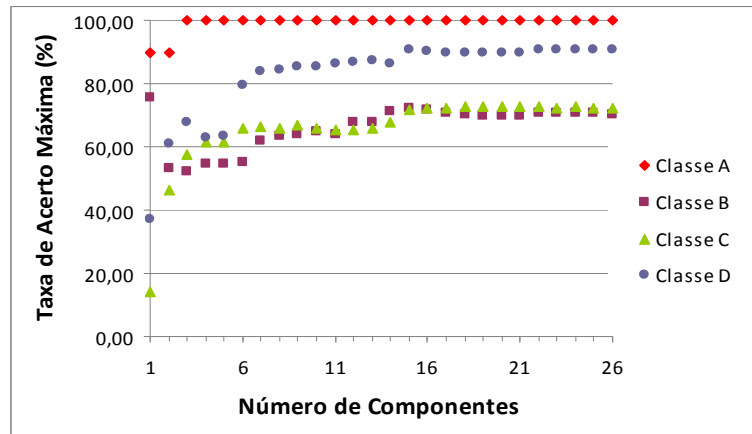


Figura 91 – Taxa de acerto de cada classe na validação cruzada com o classificador *Linear*

- Classificador *Mahalanobis*:

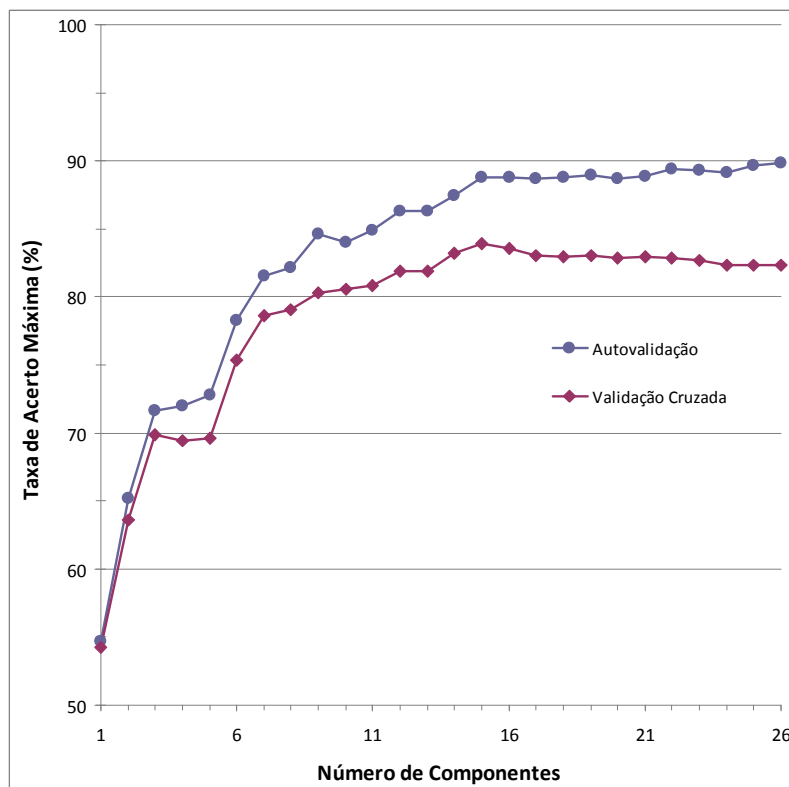


Figura 92 – Autovalidação e validação cruzada com o classificador *Mahalanobis*

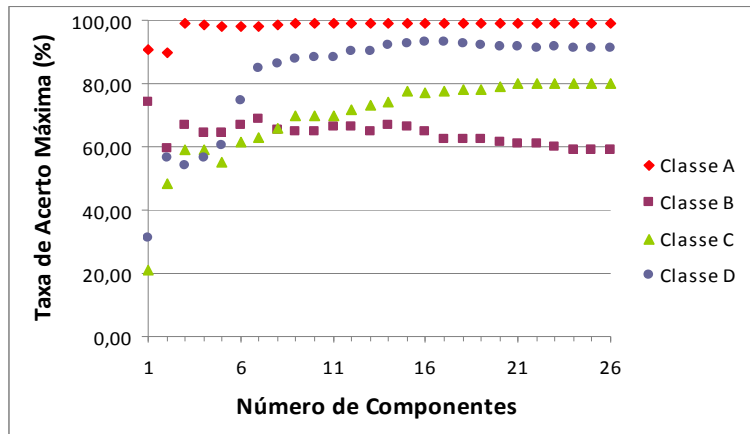


Figura 93 – Taxa de acerto de cada classe na validação cruzada com o classificador *Mahalanobis*

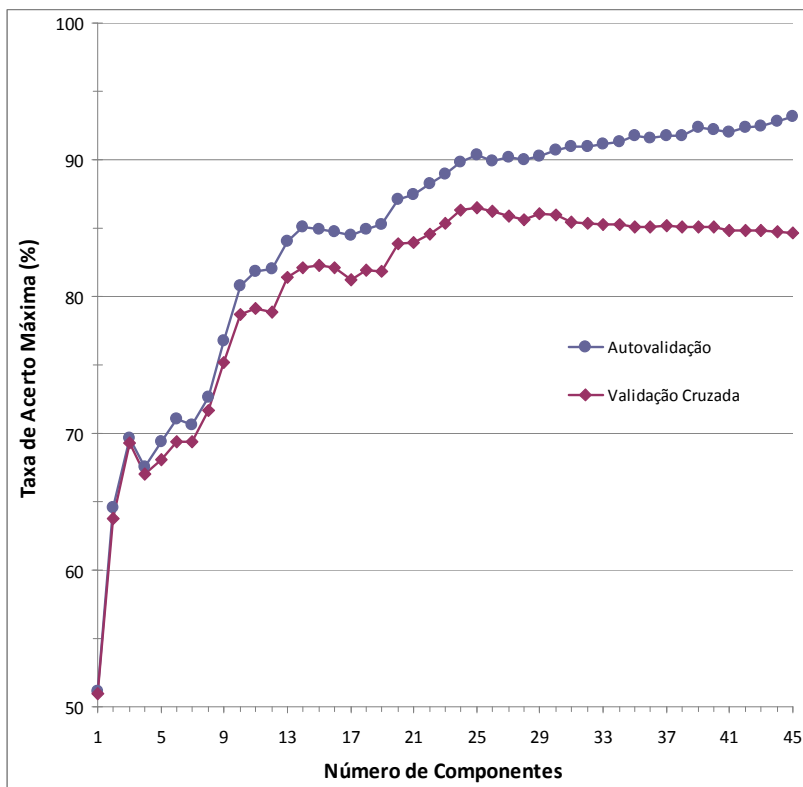


Figura 94 – Autovalidação e validação cruzada com o classificador *Mahalanobis*

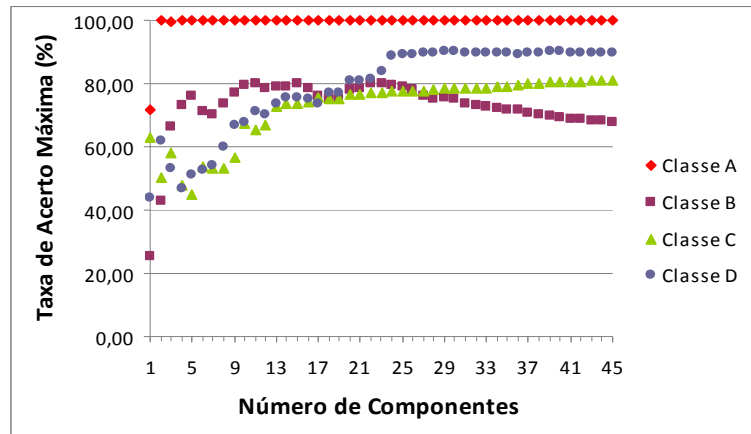


Figura 95 – Taxa de acerto de cada classe na validação cruzada com o classificador *Mahalanobis*

### 10.1.3. LDA

- Classificador *Quadrático*:

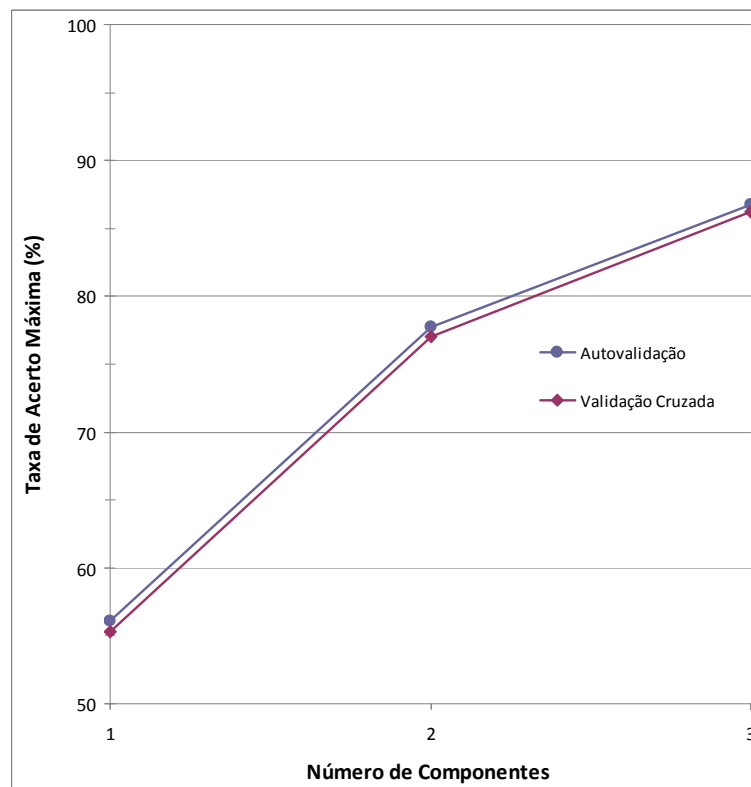


Figura 96 – Autovalidação e validação cruzada com o classificador *Quadrático* (26 parâmetros)

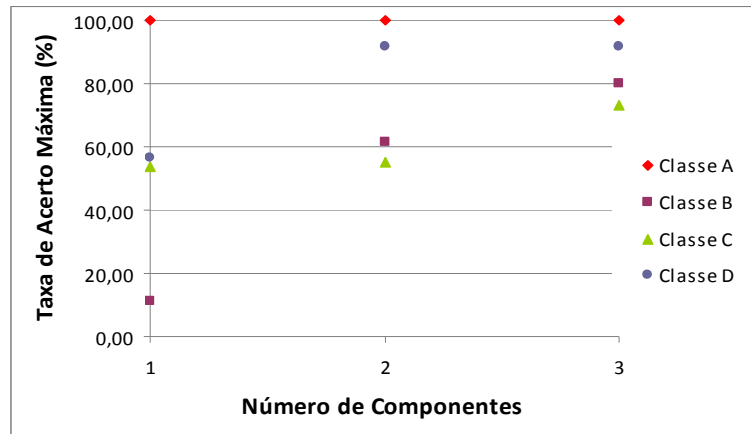


Figura 97 – Taxa de acerto de cada classe na validação cruzada com o classificador *Quadrático* (26 parâmetros)

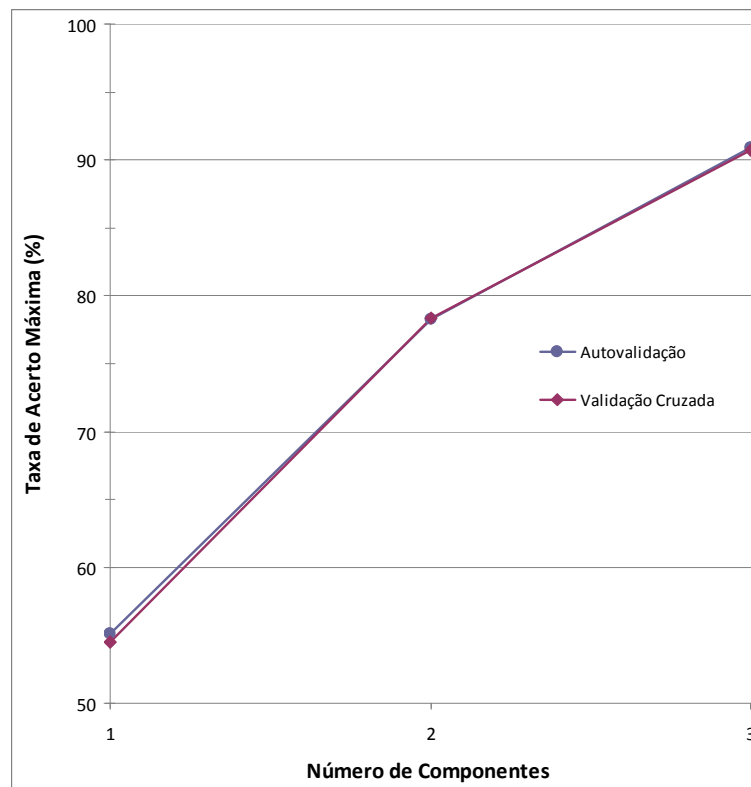


Figura 98 – Autovalidação e validação cruzada com o classificador *Quadrático* (45 parâmetros)

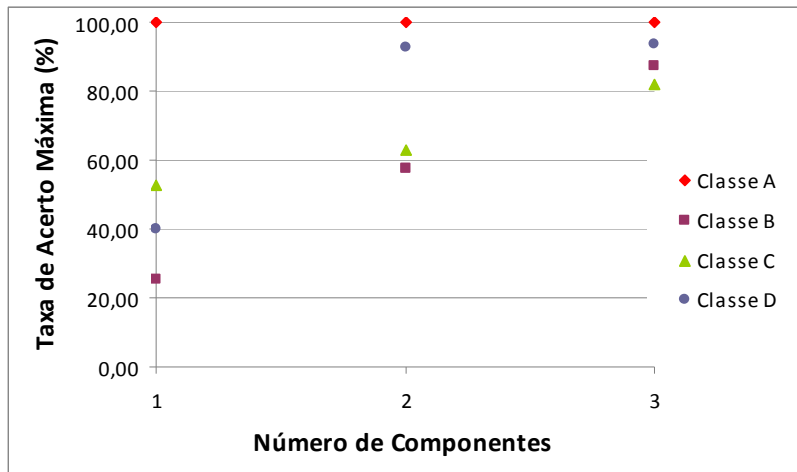


Figura 99 – Taxa de acerto de cada classe na validação cruzada com o classificador *Quadrático* (45 parâmetros)

- Classificador *Mahalanobis*:

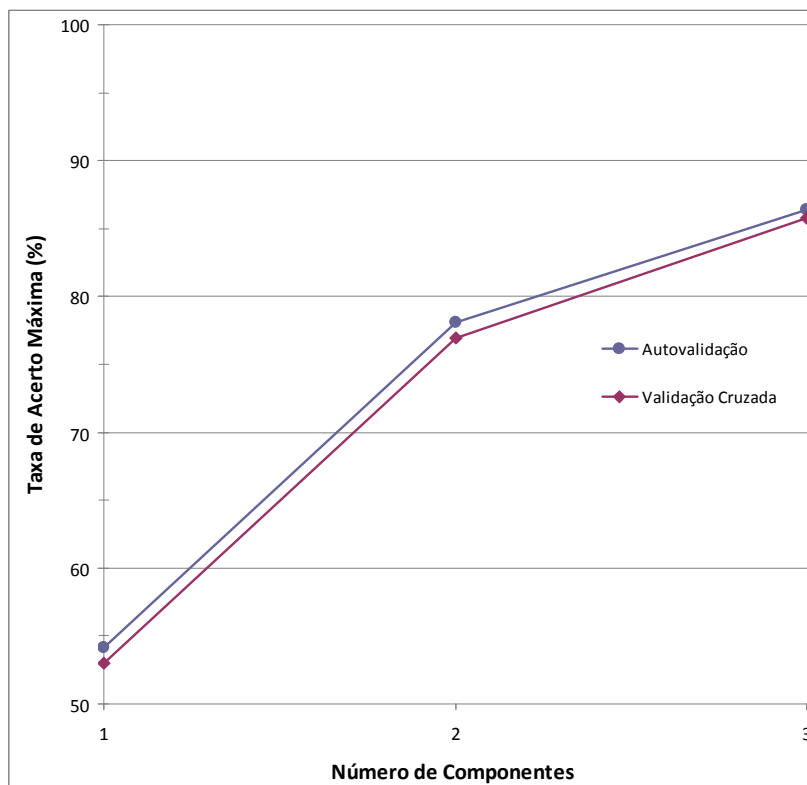


Figura 100 – Autovalidação e validação cruzada com o classificador *Mahalanobis* (26 parâmetros)

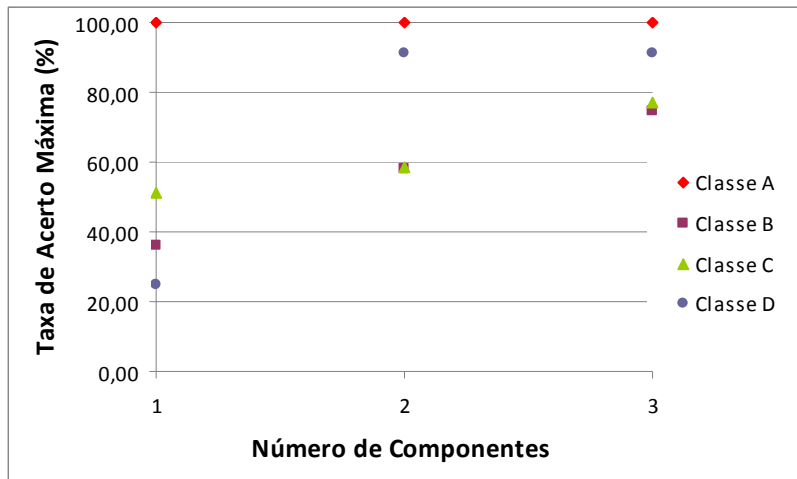


Figura 101 – Taxa de acerto de cada classe na validação cruzada com o classificador *Mahalanobis* (26 parâmetros)

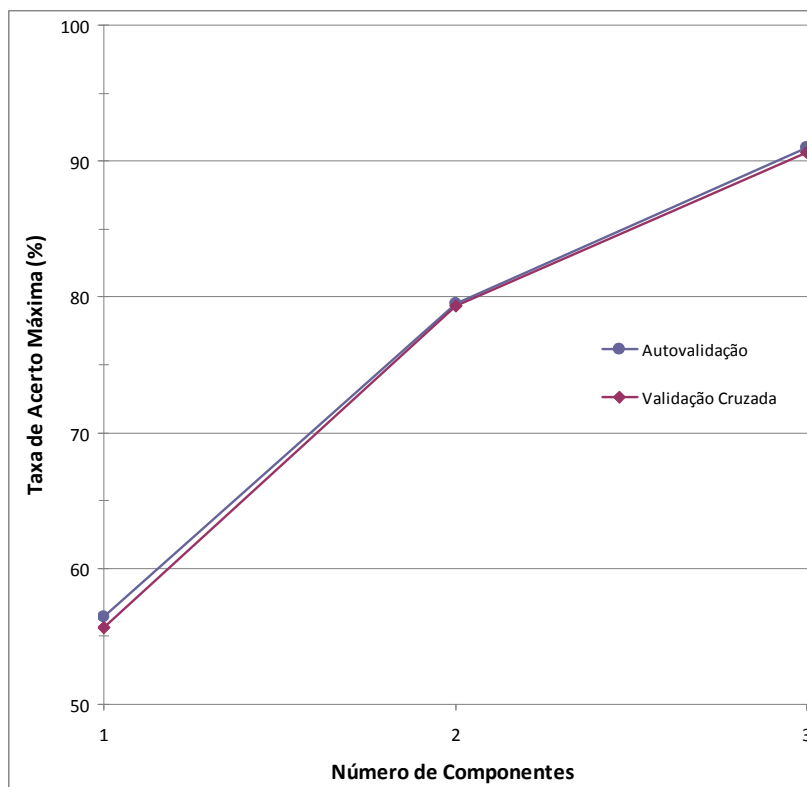


Figura 102 – Autovalidação e validação cruzada com o classificador *Mahalanobis* (45 parâmetros)

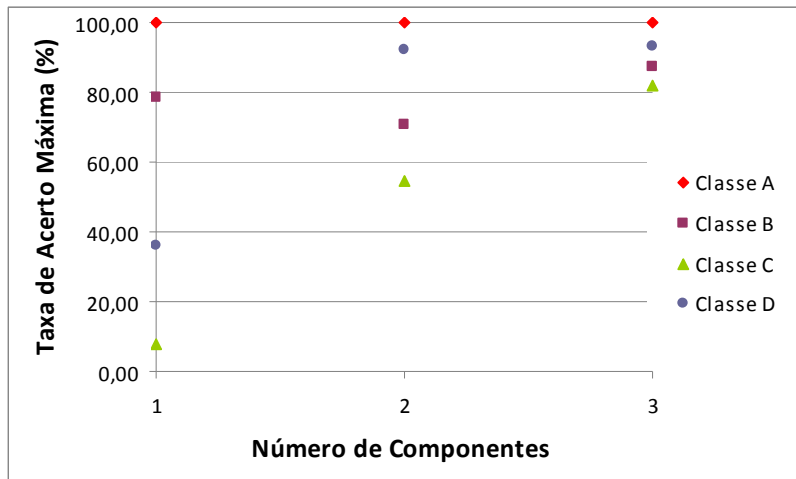


Figura 103 – Taxa de acerto de cada classe na validação cruzada com o classificador *Mahanalobis* (45 parâmetros)

## 10.2. Rotinas – KS400

### 10.2.1. Número de Partículas e Fração de Área (Sem *Watersheds*)

```
imgdelete "*"
Gclear 0
pathbase := "z:/karen/QuadrantesAF08"

# DBdelete "FIELD AF08"

classe = "A"
read classe, "Entre com a classe (A, B, C, D)"

if(classe == "A") : numeroclasse=1
if(classe == "B") : numeroclasse=2
if(classe == "C") : numeroclasse=3
if(classe == "D") : numeroclasse=4

pathimagem = pathbase + "/" + classe

imgsetpath pathimagem
DBsetpath pathbase
MSsetprop "CLASSNAME0", "unknown"
MSsetprop "CLASSNAME1", "A"
MSsetprop "CLASSNAME2", "B"
MSsetprop "CLASSNAME3", "C"
MSsetprop "CLASSNAME4", "D"

MSsetprop "FIELD FEAT", "FLDCOUNT, FLDAREAP"
if(classe == "A")
  DBnew "field", 4
  DBsetcolumn "FIELD", 1, "CLASS", "Int", "<none>"
```



```

DBsetcolumn "FIELD",2,"CLASSNAME","String",<none>"
DBsetcolumn "FIELD",3,"COUNT","Float",<none>"
DBsetcolumn "FIELD",4,"AREAP","Float",<none>"
DBfirstline "field"
endif
img = "*.tif"

while 1
  imgenum img, 1
  if (not _STATUS): break
  imgload img,1
  delin 1,2,3,3
  disaut 2,3
  binscrap 3,4,0,100,0
  RGnew 4,4
  MSmeasall "campo",0,2
  MSgetvalue "FLDCOUNT",count
  MSgetvalue "FLDAREAP",areap

  DBaddline "field"
  DBsetvalue "field","CLASS",numeroclasse
  DBsetvalue "field","CLASSNAME",classe
  DBsetvalue "field","COUNT",count
  DBsetvalue "field","AREAP",areap
  DBnextline "field"
  datalist "field"

endwhile

```

### 10.2.2. Número de Partículas (Com *Watersheds*)

```

imgdelete ".*"
Gclear 0
pathbase := "z:/karen/QuadrantesAF08"

# DBdelete "FIELDWATERSHED"

classe = "A"
read classe, "Entre com a classe (A, B, C, D)"

if(classe == "A") : numeroclasse=1
if(classe == "B") : numeroclasse=2
if(classe == "C") : numeroclasse=3
if(classe == "D") : numeroclasse=4

pathimagem = pathbase + "/" + classe

imgsetpath pathimagem
DBsetpath pathbase

```

```

MSsetprop "CLASSNAME0", "unknown"
MSsetprop "CLASSNAME1", "A"
MSsetprop "CLASSNAME2", "B"
MSsetprop "CLASSNAME3", "C"
MSsetprop "CLASSNAME4", "D"

MSsetprop "FIELDFEAT", "FLDCOUNT, FLDAREAP"
if(classe == "A")
  DBnew "fieldwatershed", 4
  DBsetcolumn "fieldwatershed", 1, "CLASS", "Int", "<none>"
  DBsetcolumn "fieldwatershed", 2, "CLASSNAME", "String", "<none>"
  DBsetcolumn "fieldwatershed", 3, "COUNT", "Float", "<none>"
  DBfirstline "fieldwatershed"
endif

img = "*.tif"

while 1
  imgenum img, 1
  if (not _STATUS): break
  imgload img, 1
  delin 1, 2, 3, 3
  disaut 2, 3
  binnot 3, 4
  binscrap 4, 5, 0, 100, 0
  binnot 5, 6
  grainsbin 6, 7, 2, 7, 1, 12
  RGnew 7, 7
  MSmeasall "campo", 0, 2
  MSgetvalue "FLDCOUNT", count
  MSgetvalue "FLDAREAP", areap

  DBaddline "fieldwatershed"
  DBsetvalue "fieldwatershed", "CLASS", numeroclasse
  DBsetvalue "fieldwatershed", "CLASSNAME", classe
  DBsetvalue "fieldwatershed", "COUNT", count
  DBnextline "fieldwatershed"
  datalist "fieldwatershed"
endwhile

```

### 10.2.3. Área Simples, Perímetro Simples, DCE, FFC<sub>1</sub>, FFC<sub>2</sub>, RA, Conv<sub>1</sub> e Conv<sub>2</sub> – Partículas

```

imgdelete "*.*"
Gclear 0
pathbase := "z:/karen/QuadrantesAF08"

# DBdelete "REGION"
# DBdelete "WATERSHEDREGION"

```

```

classe = "A"
read classe, "Entre com a classe (A, B, C, D)

if(classe == "A") : numeroclas=1
if(classe == "B") : numeroclas=2
if(classe == "C") : numeroclas=3
if(classe == "D") : numeroclas=4

pathimagem = pathbase + "/" + classe

imgsetpath pathimagem
DBsetpath pathbase
MSsetprop "CLASSNAME0", "unknown"
MSsetprop "CLASSNAME1", "A"
MSsetprop "CLASSNAME2", "B"
MSsetprop "CLASSNAME3", "C"
MSsetprop "CLASSNAME4", "D"

MSsetprop "CLASSIFIER", "<none>"
MSsetprop
"REGIONFEAT", "AREA,DCIRCLE,FCIRCLE,FERETRATIO,PERIM,FERET
MAX,"
MSsetprop
"REGIONFEAT", ".,Conv1=AREA/AREAC,Conv2=PERIMC/PERIM,FCIRC2=
4*AREAF/(PI*FERETMAX*FERETMAX)"

if(classe == "A")

    DBnew "watershedregion", 20
    DBsetcolumn "watershedregion", 1, "CLASS", "Int", "<none>"
    DBsetcolumn "watershedregion", 2, "CLASSNAME", "String", "<none>"
    DBsetcolumn "watershedregion", 3, "areamean", "Float", "<none>"
    DBsetcolumn "watershedregion", 4, "dcirclemean", "Float", "<none>"
    DBsetcolumn "watershedregion", 5, "fcirclemean", "Float", "<none>"
    DBsetcolumn "watershedregion", 6, "fcirc2mean", "Float", "<none>"
    DBsetcolumn "watershedregion", 7, "feretmaxmean", "Float", "<none>"
    DBsetcolumn "watershedregion", 8, "feretratiomean", "Float", "<none>"
    DBsetcolumn "watershedregion", 9, "perimmean", "Float", "<none>"
    DBsetcolumn "watershedregion", 10, "conv1mean", "Float", "<none>"
    DBsetcolumn "watershedregion", 11, "conv2mean", "Float", "<none>"
    DBfirstline "watershedregion"
endif

img = "*.tif"

while 1
    imgenum img, 1
    if (not _STATUS): break
    imgload img, 1

```

```

delin 1,2,3,3
disaut 2,3
binnot 3,4
binscrap 4,5, 0,100,0
binnot 5,6
grainsbin 6,7,2,7,1,12
RGnew 7,7
MSmeasall "region",1,1

```

```

datastatvalue "region","AREA","MEAN",areamean
datastatvalue "region","DCIRCLE","MEAN",dcirclemean
datastatvalue "region","FCIRCLE","MEAN",fcirclemean
datastatvalue "region","FCIRC2","MEAN",fcirc2mean
datastatvalue "region","FERETMAX","MEAN",feretmaxmean
datastatvalue "region","FERETRATIO","MEAN",feretratiomean
datastatvalue "region","PERIM","MEAN",perimmean
datastatvalue "region","Conv1","MEAN",conv1mean
datastatvalue "region","Conv2","MEAN",conv2mean

```

```

DBaddline "watershedregion"
DBsetvalue "watershedregion","CLASS",numeroclasse
DBsetvalue "watershedregion","CLASSNAME",classe
DBsetvalue "watershedregion","areamean", areamean
DBsetvalue "watershedregion","dcirclemean", dcirclemean
DBsetvalue "watershedregion","fcirclemean", fcirclemean
DBsetvalue "watershedregion","fcirc2mean", fcirc2mean
DBsetvalue "watershedregion","feretmaxmean", feretmaxmean
DBsetvalue "watershedregion","feretratiomean", feretratiomean
DBsetvalue "watershedregion","perimmean", perimmean
DBsetvalue "watershedregion","conv1mean", conv1mean
DBsetvalue "watershedregion","conv2mean", conv2mean

```

```

DBnextline "watershedregion"
datalist "watershedregion"
Gclear 0
imgdelete "*.*"

```

```

DBdelete "REGION"

```

```

endwhile

```

#### 10.2.4. Textura (Campo Claro)

```

imgdelete "*.*"
Gclear 0

```

```

pathbase := "z:/karen/QuadrantesAF08"

```

```

# DBdelete "TEXTURA"

```

```

classe = "A"
read classe, "Entre com a classe (A, B, C, D)"
pathimagem = pathbase + "/" + classe

imgsetpath pathimagem
DBsetpath pathbase

    MSsetprop "REGIONFEAT", "HARAM1, HARAM2, HARAM3, HARAM4,
HARAM5, HARAM6, HARAM7"
    MSsetprop "REGIONFEAT", "„HARAM8, HARAM9, HARAM10,
HARAM11"
    MSsetprop "REGIONFEAT", "„MEAND, STDD "

img = "*.tif"

    imgenum img, 1
    imgload img, 1
    imgpal2grey 1, 1
    imgcopy 1, 2
    imgclear 2, 255

while 1
    imgenum img, 1
    if (not _STATUS): break
    imgload img, 1
    imgpal2grey 1, 1
    RGnew 2, 1
    MSmeasall "textura", 1, 1
    datalist "textura"
endwhile

```

### 10.2.5. Textura (Campo Claro + Polarização)

```

imgdelete ".*.*"
Gclear 0

# DBdelete "HARALICKPOL"

imgload "A1_BF.TIF", 1
imgcopy 1, 6
imgclear 6, 255

classe = "A"
read classe, "Entre com a classe (A, B, C, D)"

pathbase := "z:/karen/Polarização"
DBsetpath pathbase

```

```
pathimagem = pathbase
imgsetpath pathimagem
```

```
MSsetprop "CLASSIFIER", "<none>"
```

```
MSsetprop "CLASSNAME0", "unknown"
```

```
MSsetprop "CLASSNAME1", "A"
```

```
MSsetprop "CLASSNAME2", "B"
```

```
MSsetprop "CLASSNAME3", "C"
```

```
MSsetprop "CLASSNAME4", "D"
```

```
MSsetprop "REGIONFEAT", "HARAM1, HARAM2, HARAM3, HARAM4,
HARAM5, HARAM6, HARAM7"
```

```
MSsetprop "REGIONFEAT", "„HARAM8, HARAM9, HARAM10,
HARAM11"
```

```
MSsetprop "REGIONFEAT", "„MEAND, STDD "
```

```
img = "*.tif"
```

```
while 1
```

```
#SemPol
```

```
    imgenum img, 1
```

```
    if (not _STATUS): break
```

```
    imgload img,2
```

```
    imgpal2grey 2,2
```

```
    disaut 2,3,0,1
```

```
#ComPol
```

```
    imgenum img, 1
```

```
    imgload img,4
```

```
    imgpal2grey 4,4
```

```
#União das Imagens
```

```
    binand 3,4,5
```

```
    RGnew 6,5
```

```
    MSmeasall "haralickpol",1,1
```

```
    datalist "haralickpol"
```

```
endwhile
```

### 10.2.6. Fração de Área – Poros Pequenos

```
imgdelete "*.*)"
```

```
Gclear 0
```

```
pathbase := "z:/karen/QuadrantesAF08"
```

```
# DBdelete "POROSPEQUENOSFIELD"
```

```

classe = "A"
read classe, "Entre com a classe (A, B, C, D)

if(classe == "A") : numeroclasse=1
if(classe == "B") : numeroclasse=2
if(classe == "C") : numeroclasse=3
if(classe == "D") : numeroclasse=4

pathimagem = pathbase + "/" + classe

imgsetpath pathimagem
DBsetpath pathbase
MSsetprop "CLASSNAME0", "unknown"
MSsetprop "CLASSNAME1", "A"
MSsetprop "CLASSNAME2", "B"
MSsetprop "CLASSNAME3", "C"
MSsetprop "CLASSNAME4", "D"

MSsetprop "FIELDFEAT", "FLDAREAP"
if(classe == "A")
  DBnew "porospequenosfield", 3
  DBsetcolumn "POROSPEQUENOSFIELD", 1, "CLASS", "Int", "<none>"
  DBsetcolumn
"POROSPEQUENOSFIELD", 2, "CLASSNAME", "String", "<none>"
  DBsetcolumn "POROSPEQUENOSFIELD", 3, "AREAP", "Float", "<none>"
  DBfirstline "porospequenosfield"
endif
img = "*.tif"

while 1
  imgenum img, 1
  if (not _STATUS): break
  imgload img, 1
  delin 1, 2, 3, 3
  disaut 2, 3
  binnot 3, 4
  binscrap 4, 5, 0, 3000, 1
  RGnew 5, 5
  MSmeasall "campo", 0, 2
  MSgetvalue "FLDAREAP", areap

  DBaddline "porospequenosfield"
  DBsetvalue "porospequenosfield", "CLASS", numeroclasse
  DBsetvalue "porospequenosfield", "CLASSNAME", classe
  DBsetvalue "porospequenosfield", "AREAP", areap
  DBnextline "porospequenosfield"
  datalist "porospequenosfield"

endwhile

```

### 10.2.7. Área Simples, Perímetro Simples, DCE, $FFC_1$ , $FFC_2$ , RA, $Conv_1$ e $Conv_2$ – Poros Pequenos

```

imgdelete "*"
Gclear 0
pathbase := "z:/karen/QuadrantesAF08"

# DBdelete "REGION"
# DBdelete "POROSPEQUENOSREGION"

classe = "A"
read classe, "Entre com a classe (A, B, C, D)

if(classe == "A") : numeroclasse=1
if(classe == "B") : numeroclasse=2
if(classe == "C") : numeroclasse=3
if(classe == "D") : numeroclasse=4

pathimagem = pathbase + "/" + classe

imgsetpath pathimagem
DBsetpath pathbase
MSsetprop "CLASSNAME0", "unknown"
MSsetprop "CLASSNAME1", "A"
MSsetprop "CLASSNAME2", "B"
MSsetprop "CLASSNAME3", "C"
MSsetprop "CLASSNAME4", "D"

MSsetprop "CLASSIFIER", "<none>"
MSsetprop "REGIONFEAT", "CLASS, CLASSNAME"
MSsetprop
"REGIONFEAT", "AREA,DCIRCLE,FCIRCLE,FERETRATIO,PERIM,FERET
MAX,"
MSsetprop
"REGIONFEAT", ".,Conv1=AREA/AREAC,Conv2=PERIMC/PERIM,FCIRC2=
4*AREAF/(PI*FERETMAX*FERETMAX)"

if(classe == "A")

DBnew "porospequenosregion",20
DBsetcolumn "porospequenosregion",1,"CLASS","Int",<none>"
DBsetcolumn "porospequenosregion",2,"CLASSNAME","String",<none>"
DBsetcolumn "porospequenosregion",3,"areamean","Float",<none>"
DBsetcolumn "porospequenosregion",4,"dcirclemean","Float",<none>"
DBsetcolumn "porospequenosregion",5,"fcirclemean","Float",<none>"
DBsetcolumn "porospequenosregion",6,"fcirc2mean","Float",<none>"
DBsetcolumn "porospequenosregion",7,"feretmaxmean","Float",<none>"
DBsetcolumn "porospequenosregion",8,"feretratiomean","Float",<none>"
DBsetcolumn "porospequenosregion",9,"perimmean","Float",<none>"
DBsetcolumn "porospequenosregion",10,"conv1mean","Float",<none>"

```



```

DBsetcolumn "porospequenosregion",11,"conv2mean","Float","<none>"
DBsetcolumn "porospequenosregion",12,"areastddev","Float","<none>"
DBsetcolumn "porospequenosregion",13,"dcirclestddev","Float","<none>"
DBsetcolumn "porospequenosregion",14,"fcirclestddev","Float","<none>"
DBsetcolumn "porospequenosregion",15,"fcirc2stddev","Float","<none>"
DBsetcolumn "porospequenosregion",16,"feretmaxstddev","Float","<none>"
DBsetcolumn "porospequenosregion",17,"feretratiostdd","Float","<none>"
DBsetcolumn "porospequenosregion",18,"perimstddev","Float","<none>"
DBsetcolumn "porospequenosregion",19,"conv1stddev","Float","<none>"
DBsetcolumn "porospequenosregion",20,"conv2stddev","Float","<none>"

```

```

DBfirstline "porospequenosregion"
endif

```

```

img = "*.tif"

```

```

while 1
  imgenum img, 1
  if (not _STATUS): break
  imgload img, 1
  delin 1,2,3,3
  disaut 2,3
  binnot 3,4
  binscrap 4,5,0,3000,1
  RGnew 5,5
  MSmeasall "region",1,1

```

```

datastatvalue "region","AREA","MEAN",areamean
datastatvalue "region","DCIRCLE","MEAN",dcirclemean
datastatvalue "region","FCIRCLE","MEAN",fcirclemean
datastatvalue "region","FCIRC2","MEAN",fcirc2mean
datastatvalue "region","FERETMAX","MEAN",feretmaxmean
datastatvalue "region","FERETRATIO","MEAN",feretratiomean
datastatvalue "region","PERIM","MEAN",perimmean
datastatvalue "region","Conv1","MEAN",conv1mean
datastatvalue "region","Conv2","MEAN",conv2mean

```

```

DBaddline "porospequenosregion"
DBsetvalue "porospequenosregion","CLASS",numeroclasse
DBsetvalue "porospequenosregion","CLASSNAME",classe
DBsetvalue "porospequenosregion","areamean", areamean
DBsetvalue "porospequenosregion","dcirclemean", dcirclemean
DBsetvalue "porospequenosregion","fcirclemean", fcirclemean
DBsetvalue "porospequenosregion","fcirc2mean", fcirc2mean
DBsetvalue "porospequenosregion","feretmaxmean", feretmaxmean
DBsetvalue "porospequenosregion","feretratiomean", feretratiomean
DBsetvalue "porospequenosregion","perimmean", perimmean
DBsetvalue "porospequenosregion","conv1mean", conv1mean
DBsetvalue "porospequenosregion","conv2mean", conv2mean

```

```
DBdelete "REGION"
endwhile
```

### 10.2.8. MR e BF – Poros Pequenos

```
imgdelete ".*.*"
Gclear 0
pathbase := "z:/karen/QuadrantesAF08"

# DBdelete "REGION"
# DBdelete "POROSPEQUENOSMRBF"

classe = "A"
read classe, "Entre com a classe (A, B, C, D)"

if(classe == "A") : numeroclasse=1
if(classe == "B") : numeroclasse=2
if(classe == "C") : numeroclasse=3
if(classe == "D") : numeroclasse=4

pathimagem = pathbase + "/" + classe

imgsetpath pathimagem
DBsetpath pathbase
MSsetprop "CLASSNAME0", "unknown"
MSsetprop "CLASSNAME1", "A"
MSsetprop "CLASSNAME2", "B"
MSsetprop "CLASSNAME3", "C"
MSsetprop "CLASSNAME4", "D"

MSsetprop "CLASSIFIER", "<none>"
MSsetprop "REGIONFEAT", "CLASS, CLASSNAME"
MSsetprop "REGIONFEAT", ".,MR=(2*MAXD/FERETMAX)"
MSsetprop "REGIONFEAT", ".,BF=(2*MAXD/FERETMIN)"

if(classe == "A")

    DBnew "porospequenosmrbf",4
    DBsetcolumn "porospequenosmrbf",1,"CLASS","Int",<none>"
    DBsetcolumn "porospequenosmrbf",2,"CLASSNAME","String",<none>"
    DBsetcolumn "porospequenosmrbf",3,"mrmean","Float",<none>"
    DBsetcolumn "porospequenosmrbf",4,"bfmean","Float",<none>"

    DBfirstline "porospequenosmrbf"
endif

img = ".*.tif"

while 1
```

```

imgenum img, 1
if (not _STATUS): break
imgload img, 1
delim 1,2,3,3
disaut 2,3
binnot 3,4
binscrap 4,5,0,3000,1
binand 5,2,6
distmapeuclid 6,7
RGnew 6,7
MSmeasall "region",1,1

```

```

datastatvalue "region","MR","MEAN",mrmean
datastatvalue "region","BF","MEAN",bfmean

```

```

DBaddline "porospequenosmrbf"
DBsetvalue "porospequenosmrbf","CLASS",numeroclasse
DBsetvalue "porospequenosmrbf","CLASSNAME",classe
DBsetvalue "porospequenosmrbf","mrmean", mrmean
DBsetvalue "porospequenosmrbf","bfmean", bfmean
datalist "porospequenosmrbf"

```

```

DBdelete "REGION"

```

```

endwhile

```

### 10.2.9. Interceptos – Partículas

```

imgdelete "*" "*"
Gclear 0
pathbase := "z:/karen/QuadrantesAF08"

```

```

# DBdelete "INTERCEPTOSAF08"

```

```

classe = "A"
read classe, "Entre com a classe (A, B, C, D)"

```

```

pathimagem = pathbase + "/" + classe

```

```

imgsetpath pathimagem
DBsetpath pathbase
MSsetprop "CLASSNAME0","unknown"
MSsetprop "CLASSNAME1", "A"
MSsetprop "CLASSNAME2", "B"
MSsetprop "CLASSNAME3", "C"
MSsetprop "CLASSNAME4", "D"

```

```

MSsetprop
"FIELDFEAT", "FLDINTERCEPT[#]=SUM(INTERCEPT),FLDSUMCHORD[U

```

```
NIT]=SUM(SUMCHORD),FLDMEANCHORD[UNIT]=SUM(SUMCHORD)/SUM(INTERCEPT/2)"
MSsetprop "NSPACE",1
```

```
img = "*.tif"
```

```
while 1
  imgenum img, 1
  if (not _STATUS): break
  imgload img,1
  delin 1,2,3,3
  disaut 2,3
  binscrap 3,4,0,100,0
  RGnew 4,4
  MSmeasall "interceptos",1,2
  datalist "interceptos"
endwhile
```

### 10.2.10. Interceptos – Poros

```
imgdelete "*.*"
Gclear 0
pathbase := "z:/karen/QuadrantesAF08"
```

```
# DBdelete "INTERCEPTOSPOROS"
```

```
classe = "A"
read classe, "Entre com a classe (A, B, C, D)"
```

```
pathimagem = pathbase + "/" + classe
```

```
imgsetpath pathimagem
DBsetpath pathbase
MSsetprop "CLASSNAME0", "unknown"
MSsetprop "CLASSNAME1", "A"
MSsetprop "CLASSNAME2", "B"
MSsetprop "CLASSNAME3", "C"
MSsetprop "CLASSNAME4", "D"
```

```
MSsetprop
"FIELDFEAT", "FLDINTERCEPT[#]=SUM(INTERCEPT),FLDSUMCHORD[UNIT]=SUM(SUMCHORD),FLDMEANCHORD[UNIT]=SUM(SUMCHORD)/SUM(INTERCEPT/2)"
MSsetprop "NSPACE",1
```

```
img = "*.tif"
```

```
while 1
  imgenum img, 1
```

```

if (not _STATUS): break
imgload img,1
delin 1,2,3,3
disaut 2,3
binnot 3,4
RGnew 4,4
MSmeasall "interceptosporos",1,2
datalist "interceptosporos"
endwhile

```

### 10.2.11. Interceptos – Poros Pequenos

```

imgdelete ".*.*"
Gclear 0
pathbase := "z:/karen/QuadrantesAF08"

# DBdelete "INTERCEPTOSPOROSPEQUENOS"

classe = "A"
read classe, "Entre com a classe (A, B, C, D)"

pathimagem = pathbase + "/" + classe

imgsetpath pathimagem
DBsetpath pathbase
MSsetprop "CLASSNAME0", "unknown"
MSsetprop "CLASSNAME1", "A"
MSsetprop "CLASSNAME2", "B"
MSsetprop "CLASSNAME3", "C"
MSsetprop "CLASSNAME4", "D"

MSsetprop
"FIELDFEAT", "FLDINTERCEPT[#]=SUM(INTERCEPT),FLDSUMCHORD[UNIT]=SUM(SUMCHORD),FLDMEANCHORD[UNIT]=SUM(SUMCHORD)/SUM(INTERCEPT/2)"
MSsetprop "NSPACE", 1

img = ".*.tif"

while 1
  imgenum img, 1
  if (not _STATUS): break
  imgload img,1
  delin 1,2,3,3
  disaut 2,3
  binnot 3,4
  binscrap 4,5,0,3000,1
  RGnew 5,5
  MSmeasall "interceptosporospequenos",1,2

```

```

    datalist "interceptosporospequenos"
endwhile

```

### 10.3. Rotinas – MATLAB

#### 10.3.1. Autovalidação

```

function [TaxaA, TaxaB, TaxaC, TaxaD,
TaxaGlobal]=autovalidacao(class,group,n);

%Autovalidação

%Sintaxe:

% Entrada:
% class : matriz gerada pelo classify
% group : matriz com a classificação correta
% n : número de linhas

% Saída:
% TaxaA : Taxa de acerto da classe A
% TaxaB : Taxa de acerto da classe B
% TaxaC : Taxa de acerto da classe C
% TaxaD : Taxa de acerto da classe D
% TaxaGlobal : Média aritmética das taxas de acerto das classes

acertos=[];
for i=1:n
    if class(i,1)==group(i,1)
        acertos(i,1)=[group(i,1)];
    else
        acertos(i,1)=[0];
    end
end

Classe1=0;
Classe2=0;
Classe3=0;
Classe1Total=0;
Classe2Total=0;
Classe3Total=0;

for i=1:n

    if group(i,1)==1
        ClasseATotal=ClasseATotal+1;
    end
    if group (i,1)==2

```

```

        ClasseBTotal=ClasseBTotal+1;
    end
    if group (i,1)==3
        ClasseCTotal=ClasseCTotal+1;
    end
    if group (i,1)==4
        ClasseDTotal=ClasseDTotal+1;
    end
end

```

```

for i=1:n

```

```

    if acertos(i,1)==1
        ClasseA=ClasseA+1;
    end

```

```

    if acertos(i,1)==2
        ClasseB=ClasseB+1;
    end

```

```

    if acertos(i,1)==3
        ClasseC=ClasseC+1;
    end

```

```

    if acertos(i,1)==4
        ClasseD=ClasseD+1;
    end

```

```

end

```

```

TaxaA=((ClasseA/ClasseATotal)*100);
TaxaB=((ClasseB/ClasseBTotal)*100);
TaxaC=((ClasseC/ClasseCTotal)*100);
TaxaD=((ClasseD/ClasseDTotal)*100);
TaxaGlobal = ((TaxaA + TaxaB + TaxaC + TaxaD)/4);

```

```

clear ClasseA ClasseB ClasseC ClasseD ClasseATotal ClasseBTotal
ClasseCTotal ClasseDTotal

```

### 10.3.2. Matriz de Confusão

```

function [Matriz_de_Confusao]=matconfusao(class,group,n);

```

```

% Matriz de Confusão

```

```

% Sintaxe:

```

```

% Entrada:

```

```

% class : matriz gerada pelo classify
% group : matriz com a classificação correta
% n : número de linhas

% Saída:

% Matriz_de_Confusao : Tabela com a quantidade das classificações em cada
% classe, onde:

% ClasseA_A : Quantidade de Classes A classificada como Classe A
% ClasseA_B : Quantidade de Classes A classificada como Classe B
% ClasseA_C : Quantidade de Classes A classificada como Classe C
% ClasseA_D : Quantidade de Classes A classificada como Classe D
% ClasseB_B : Quantidade de Classes B classificada como Classe B
% ClasseB_A : Quantidade de Classes B classificada como Classe A
% ClasseB_C : Quantidade de Classes B classificada como Classe C
% ClasseB_D : Quantidade de Classes B classificada como Classe D
% ClasseC_C : Quantidade de Classes C classificada como Classe C
% ClasseC_A : Quantidade de Classes C classificada como Classe A
% ClasseC_B : Quantidade de Classes C classificada como Classe B
% ClasseC_D : Quantidade de Classes C classificada como Classe D
% ClasseD_D : Quantidade de Classes D classificada como Classe D
% ClasseD_A : Quantidade de Classes D classificada como Classe A
% ClasseD_B : Quantidade de Classes D classificada como Classe B
% ClasseD_C : Quantidade de Classes D classificada como Classe C

ClasseA_A = 0;
ClasseA_B = 0;
ClasseA_C = 0;
ClasseA_D = 0;
ClasseB_B = 0;
ClasseB_A = 0;
ClasseB_C = 0;
ClasseB_D = 0;
ClasseC_C = 0;
ClasseC_A = 0;
ClasseC_B = 0;
ClasseC_D = 0;
ClasseD_D = 0;
ClasseD_A = 0;
ClasseD_B = 0;
ClasseD_C = 0;

for i=1:n

    if group (i,1)== 1
        if class (i,1) == 1
            ClasseA_A = ClasseA_A + 1;
        end
        if class (i,1) == 2

```



```

        ClasseA_B = ClasseA_B + 1;
    end
    if class (i,1) == 3
        ClasseA_C = ClasseA_C + 1;
    end
    if class (i,1) == 4
        ClasseA_D = ClasseA_D + 1;
    end
end
end
end

```

```

for i=1:n

```

```

    if group (i,1)== 2
        if class (i,1) == 2
            ClasseB_B = ClasseB_B + 1;
        end
        if class (i,1) == 1
            ClasseB_A = ClasseB_A + 1;
        end
        if class (i,1) == 3
            ClasseB_C = ClasseB_C + 1;
        end
        if class (i,1) == 4
            ClasseB_D = ClasseB_D + 1;
        end
    end
end

```

```

end

```

```

for i=1:n

```

```

    if group (i,1)== 3
        if class (i,1) == 3
            ClasseC_C = ClasseC_C + 1;
        end
        if class (i,1) == 1
            ClasseC_A = ClasseC_A + 1;
        end
        if class (i,1) == 2
            ClasseC_B = ClasseC_B + 1;
        end
        if class (i,1) == 4
            ClasseC_D = ClasseC_D + 1;
        end
    end
end

```

```

end

```

```

for i=1:n

```

```

if group (i,1)== 4
    if class (i,1) == 4
        ClasseD_D = ClasseD_D + 1;
    end
    if class (i,1) == 1
        ClasseD_A = ClasseD_A + 1;
    end
    if class (i,1) == 2
        ClasseD_B = ClasseD_B + 1;
    end
    if class (i,1) == 3
        ClasseD_C = ClasseD_C + 1;
    end
end
end
end

```

```

Matriz_de_Confusao=[ClasseA_A, ClasseA_B, ClasseA_C, ClasseA_D;
ClasseB_A, ClasseB_B, ClasseB_C, ClasseB_D; ClasseC_A, ClasseC_B,
ClasseC_C, ClasseC_D; ClasseD_A, ClasseD_B, ClasseD_C, ClasseD_D]

```

### 10.3.3. Validação Cruzada

```

function [ValGlobal,Val,cl]=holdoutestimate(padroes,group,type)
%
% Validação cruzada 'holdout estimate'
%
% Sintaxe:
%   ValGlobal = holdoutestimate(padroes,group,type)
%   [ValGlobal,Val,cl] = holdoutestimate(padroes,group,type)
%
% Entrada:
%   padroes : vetor coluna com os padrões
%   group   : vetor coluna com as classes dos padrões
%   type    : tipo de classificação ('linear' ou 'quadratic')
%
% Output:
%   ValGlobal : taxa de acerto média global
%   Val       : taxa de acerto média por classe
%   cl       : número de validações realizadas
%
% (c) Otávio Gomes
%   ogomes@gmail.com

n = max(group(:));
for i=1:n
    c(i) = sum(group == i);
end

gmax = fix(min(c(:))/2);

```

```

cl = 0;

for g=1:gmax
  if g>=3
    inimax = g-1;
  else
    inimax = 0;
  end
  for ini=0:inimax
    ti = 1;
    pi = 1;
    vi = 1;
    for i=1:n
      ng = fix((c(i)-ini)/g);
      if mod(ng,2)>0
        ng = ng - 1;
      end
      if ini > 0
        pf = pi + ini - 1;
        vf = vi + ini - 1;
        cv(vi:vf,:) = padroes(pi:pf,:);
        grpv(vi:vf,:) = group(pi:pf,:);
        pi = pf + 1;
        vi = vf + 1;
      end

      for k=1:2:(ng-1)
        tf = ti + g - 1;
        pf = pi + g - 1;
        ct(ti:tf,:) = padroes(pi:pf,:);
        grpt(ti:tf,:) = group(pi:pf,:);
        pi = pf + 1;
        pf = pi + g - 1;
        vf = vi + g - 1;
        cv(vi:vf,:) = padroes(pi:pf,:);
        grpv(vi:vf,:) = group(pi:pf,:);
        ti = tf + 1;
        pi = pf + 1;
        vi = vf + 1;
      end

      sobra = (c(i)-ini) - (g*ng);
      if sobra > 0
        if sobra <= ini
          pf = pi + sobra - 1;
          tf = ti + sobra - 1;
          ct(ti:tf,:) = padroes(pi:pf,:);
          grpt(ti:tf,:) = group(pi:pf,:);
        else

```

```

    sobrat = ini + fix((sobra-ini)/2);
    sobrav = sobra - sobrat;
    tf = ti + sobrat - 1;
    pf = pi + sobrat - 1;
    ct(ti:tf,:) = padroes(pi:pf,:);
    grpt(ti:tf,:) = group(pi:pf,:);
    pi = pf + 1;
    pf = pi + sobrav - 1;
    vf = vi + sobrav - 1;
    cv(vi:vf,:) = padroes(pi:pf,:);
    grpv(vi:vf,:) = group(pi:pf,:);
    ti = tf + 1;
end
pi = pf + 1;
vi = vf + 1;
end
end
cl = cl + 1;
class = classify(cv,ct,grpt,type);
erros = (class-grpv)~=0;
errospc = erros.*grpv;
for i=1:n
    acerto(cl,i)=1-(sum(errospc==i)/sum(grpv==i));
end
acerto(cl,n+1)=1-(sum(erros(:))/length(class));

cl = cl + 1;
class = classify(ct,cv,grpv,type);
erros = (class-grpt)~=0;
errospc = erros.*grpt;
for i=1:n
    acerto(cl,i)=1-(sum(errospc==i)/sum(grpt==i));
end
acerto(cl,n+1)=1-(sum(erros(:))/length(class));
clear ct grpt cv grpv class;
end
end

ValGlobal=mean(acerto(:,n+1));
Val=mean(acerto);
Val=Val(:,1:n);

```

#### 10.3.4. Seleção de Atributos por Busca Exaustiva da Melhor Taxa de Acerto

```

function [TAmaz,iDmax] = selctn(sample,group,type,n)
%
% Seleção do Melhor Conjunto de Treinamento com N Atributos por Busca
Exaustiva

```

```

%
% Sintaxe:
% TMax = selctn(sample,group,type,n)
% [TMax,iDmax] = selctn(sample,group,type,n)
%
% Entrada:
% sample : matriz com os padrões nas linhas, cada coluna é um atributo
% group : vetor coluna com as classes dos padrões
% type : tipo de classificação ('linear', 'quadratic', ou 'mahalanobis')
% n : número de atributos do conjunto de treinamento
%
% Output:
% TMax : taxa de acerto máxima
% iDmax : índice (número decimal) do conjunto de atributos com taxa de
% acerto máxima
%
% (c) Otávio Gomes
% ogomes@gmail.com
%
% Versão: 25/04/2012

```

```

natrib = size(sample,2); % número de atributos total

```

```

if nargin < 4, n = natrib; end

```

```

if n < 1, n = 1; end

```

```

if n > natrib, n = natrib; end

```

```

ncombtot = (2^natrib)-1;

```

```

%-----

```

```

A=[1:ncombtot]';

```

```

B=A;

```

```

X=zeros(size(B));

```

```

S=X;

```

```

for i=1:natrib-1

```

```

    X=mod(B,2);

```

```

    S=S+X;

```

```

    B=B-X;

```

```

    B=B./2;

```

```

end

```

```

S=S+B;

```

```

clear X B;

```

```

T=S==n;

```

```

U=A.*T;

```

```

V=nonzeros(U);

```

```

clear T A U;

```

```

%-----

```

```

ncomb = size(V,1);

```

```

TAmax = 0;

for i=1:ncomb

    dec = V(i);
    strbin = dec2bin(dec,natrib);

    ss = zeros(size(sample,1),n);

    k=0;
    for j=1:natrib
        r = str2double(strbin(j));
        if r == 1
            k=k+1;
            ss(:,k)=sample(:,j);
        end
    end

    [~,err]=classify(ss,ss,group,type);
    taxadeacerto=100*(1-err);
    if taxadeacerto > TAmax
        TAmax = taxadeacerto;
        iDmax = dec;
    end;
end;

```

### 10.3.5. Whitening

```

% INPUT
% X: rows are the instances, columns are the features
% epsilon: small number to compensate for nearly 0 eigenvalue [DEFAULT =
% 0.0001]
%
% OUTPUT
% Xwh: whitened data, rows are instances, columns are features
% mu: mean of each feature of the original data
% invMat: the inverse data whitening matrix
% whMat: the whitening matrix

```

```
function [Xwh, mu, invMat, whMat] = whiten(X,epsilon)
```

```

if ~exist('epsilon','var')
    epsilon = 0.0001;
end

```

```

mu = mean(X);
X = bsxfun(@minus, X, mu);
A = X'*X;
[V,D,notused] = svd(A);

```

```
whMat = sqrt(size(X,1)-1)*V*sqrtm(inv(D + eye(size(D))*epsilon))*V';  
Xwh = X*whMat;  
invMat = pinv(whMat);
```

```
end
```