UNIVERSIDADE DE ÉVORA

Mestrado em Engenharia Informática

Processamento e Classificação de Imagens Biológicas: Aplicação à Histologia

Bruno Miguel Bárbara Nunes

Orientador
Professor Luís Miguel Rato

Janeiro de 2011

Resumo

A Histologia, o estudo de tecidos, é uma das áreas fundamentais da Biologia que permitiu enormes avanços científicos. Sendo uma tarefa exigente, meticulosa e demorada, será importante aproveitar a existência de ferramentas e algoritmos computacionais no seu auxílio, tornando o processo mais rápido e possibilitando a descoberta de informação que poderá não estar visível à partida.

Esta dissertação tem como principal objectivo averiguar se um animal foi ou não sujeito à ingestão de um xenobiótico.

Com esse objectivo em vista, utilizaram-se técnicas de processamento e segmentação de imagem aplicadas a imagens de tecido renal de ratos saudáveis e ratos que ingeriram o xenobiótico. Destas imagens extraíram-se inúmeras características do corpúsculo renal que após serem analisadas através de vários algoritmos de classificação mostraram ser possível saber se o animal ingeriu ou não o xenobiótico, com um reduzido grau de incerteza.

Palavras-Chave: *Processamento de Imagem, Segmentação, Extracção de Características, Classificação, Aprendizagem Automática, Histologia, Gomérulos Renais, Xenobióticos*

Processing and Classification of Biological Images: Application to Histology

ABSTRACT

Histology, the study of tissues, is one of the key areas of Biology that has allowed huge advances in Science. Being a demanding, meticulous and time consuming task, it is important to use the existence of computational tools and algorithms in its aid, making the process faster and enabling the discovery of information that may not be initially visible.

The main goal of this thesis is to ascertain if an animal was subjected or not to the ingestion of a xenobiotic.

With this in mind, were used image processing and segmentation techniques applied on images of kidney tissue from healthy rats and rats that ingested the xenobiotic. From these images were extracted several features of renal glomeruli that after being analyzed by various classification algorithms had shown to be possible to know, with an acceptable degree of certainty, if the animal ingested or not the xenobiotic.

Keywords: *Image Processing, Segmentation, Feature Extraction, Classification, Machine Learning, Histology, Renal Glomerulus, Xenobiotics*

Agradecimentos

Ao atingir este objectivo da minha vida, não posso deixar de agradecer às pessoas que me apoiaram e fizeram parte de todo o meu processo de formação pessoal e académica:

Aos meus pais, Manuel Bárbara Nunes e Maria Celízia Nunes, os sólidos pilares que ergueram e sustentaram a minha educação.

À Filipa, pela cumplicidade, carinho, compreensão, paciência e sobretudo pela sua constante presença e incentivo.

A todos os Professores com que me cruzei e com quem tive o privilégio de aprender algo novo, em especial ao meu orientador, o Professor Luís Miguel Rato, que me acompanhou ao longo de todo este trabalho, estando sempre disponível para ajudar, suprimir dúvidas, tecer comentários e partilhar opiniões.

Ao Professor Fernando Capela e Silva, pela cedência das imagens utilizadas neste trabalho e por todos os esclarecimentos relacionados com a área da Biologia.

Aos meus Amigos, sobretudo aos que, estando longe, nunca deixaram de estar perto.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Termos e Acrónimos

FN Falso Negativo.

FP Falso Positivo.

T_{FP} Taxa de Falsos Positivos.

T_{VN} Taxa de Verdadeiros Negativos.

T_{VP} Taxa de Verdadeiros Positivos.

T_a Taxa de Acerto.

T_e Taxa de Erro.

VN Verdadeiro Negativo.

VP Verdadeiro Positivo.

OneR *One Rule.*

ZeroR *Zero Rule.*

ARFF *Attribute-Relation File Format.*

ASCII *American Standard Code for Information Interchange.*

AUC *Area Under Curve.*

CSV *Comma-Separated Values.*

FBC *Fractal Box Count.*

J48 *J48 Decision Tree.*

MCO *Multicell Outliner.*

NIH *National Institute of Health.*

RGB *Red, Green and Blue.*

ROC *Receiver Operating Characteristic.*

SMO *Sequential Minimal Optimization.*

SVM *Support Vector Machine.*

WEKA *Waikato Environment for Knowledge Analysis.*

XLS *Microsoft Spreadsheet File.*

Capítulo 1

Introdução

A Histologia é a área da Biologia que estuda a formação, estrutura e função dos tecidos biológicos. Tradicionalmente, os histopatologistas examinam imagens ou lâminas de cortes histológicos com o intuito de classificar tecidos, diagnosticar e fazer prognósticos de doenças ou realizar estudos sobre desequilíbrios hormonais e outros tipos de desordens. O processo, normalmente, inicia-se com a recolha de amostras de tecidos de humanos, animais ou plantas através de biópsia. Posteriormente, essas amostras são examinadas através de um microscópio. Finalmente, e tendo em conta a habitual morfologia celular e distribuição dos tecidos, são avaliados desvios ou mudanças nas estruturas das células e/ou distribuição das mesmas ao longo dos tecidos. A parte ingrata, é que muitas vezes as avaliações resultantes destes processos são algo subjectivas e apresentam alguma variabilidade e discrepância.

Torna-se então perceptível que este processo, apesar de se tratar de um processo exigente, meticuloso, demorado e que requer profissionais altamente qualificados, por vezes não produz resultados fiáveis.

Tendo em conta estes factores aliados à evolução dos computadores, problemas Histológicos passaram a ser contemplados pela Bioinformática.

Como resultado dessa aliança, cada vez mais se aproveita a existência de ferramentas e algoritmos computacionais no auxílio de processos histológicos, tornando-os mais rápidos, automáticos e possibilitando a descoberta de detalhes e informações que dificilmente um observador humano descobriria à partida [BB01, SM97, GJ01]. A utilização destas ferramentas em conjunto com a cooperação de histopatologistas permitirá efectuar estudos e avaliações com resultados mais objectivos e fiáveis.

Os avanços teóricos na Matemática e Ciências da Computação, áreas que fundamentam a Bioinformática, sofreram simultanemente uma evolução. O aparecimento de novas sub-áreas nas Ciências da Computação como a Aprendizagem Automática, a Mineração de Dados e o Processamento de Imagem, vieram revolucionar a forma como se avaliam os problemas histológicos [BB01].

Relativamente aos desafios existentes nos problemas histológicos salientam-se:

- a elimininação de ruídos nas imagens alvo de análise;
- a correcta segmentação das áreas de interesse;
- a escolha de características para representar e distinguir células/tecidos;
- a recolha e avaliação de medidas quantitativas de células/tecidos;

Esta dissertação pretende recorrer a meios computacionais para tentar superar os desafios referidos no caso específico da avaliação de tecidos renais recolhidos de ratos saudáveis e ratos sujeitos à ingestão de xenobióticos. Para tal, utilizaram-se técnicas de processamento de imagem que permitiram isolar as regiões de interesse; procedeu-se à definição de um conjunto de características capaz de caracterizar as áreas de interesse e consequentemente à medição das mesmas; e procurou-se testar várias técnicas de mineração de dados e aprendizagem automática com o objectivo de construir um modelo capaz de classificar de forma eficaz novas amostras.

Como principais contribuições desta dissertação pode destacar-se:

- a possibilidade de segmentação semi-automática de glomérulos renais utilizando um novo método;
- a medição automática das características morfológicas de glomérulos renais;
- a aplicação de classificadores a dados histomorfométricos;

1.1 Estado da Arte

O processamento digital de imagem surgiu no final dos anos 60 e desde então cresceu de tal modo que, actualmente, não existem áreas que não utilizem estas técnicas de alguma forma [AR05].

Em áreas como a Geografia, a Indústria, a Astronomia, a Segurança ou Defesa Militar, existem diversos exemplos de aplicação do processamento de imagem como: a localização de regiões ou objectos em imagens de satélite, por exemplo, veículos [ZN01] ou embarcações [SSS08, AT93], estradas [WGZ+99, SU10, XTT05], florestas [KM07], zonas poluídas [SSS96, PKF+00], fontes de recursos naturais ou mesmo fenómenos espaciais [Mee99]; sistemas de reconhecimento facial [WDL08], para combate ao crime e terrorismo ou reconhecimento de pessoas desaparecidas; sistemas de controlo de tráfego [MYHT92, THS96]; sistemas de visão computacional, para controlo de qualidade e inspecção e/ou detecção de falhas no fabrico de diversos componentes industriais [PPVRSPGP07, LF08, MNAOSA03], entre muitas outras aplicações.

Além das áreas referidas, estas técnicas são amplamente utilizadas na Medicina e na Biologia. A aplicação do processamento de imagem em microscopias [MM90, FSMC06, Bru88], radiografias [Tru81, HHH94, ZSW+08, OML+08], tomografias computadorizadas [MY08, QSWR03], ecografias ou ressonâncias magnéticas [LBSM09, CGBL+05, KNAT07] permite, entre muitas coisas, a localização de tumores e outras patologias [UMS89, TJO01], a medição de volume de tecidos [LELL03], o diagnóstico de doenças [WMZDD89, AYAN89, MFM+05, SCSG09] ou o estudo de estruturas anatómicas [AR05, GW01, HKC07].

Existem também referências de trabalhos em que técnicas de processamento e segmentação de imagem são utilizadas para o estudo de problemas histológicos [PKC+04, CTCW07, HBP09, NJA10].

No caso específico de glomérulos renais, são conhecidos alguns trabalhos onde foram aplicadas algumas técnicas de segmentação como: transformada de *wavelet* e algoritmo *watershed* [ZF06a, ZF06b, ZZ07], método de Saltykov [AdAdS+96], curvas de ajuste [ZH08a], programação dinâmica [YMK88], método de Otsu [ZH08b] ou método *split and merge* [KRB08].

Também são conhecidos trabalhos em que técnicas mineração de dados e aprendizagem automática, como redes bayesianas, regras de associação, árvores de decisão e *Support Vector Machines* foram utilizadas para a classificação de imagens médicas e biológicas [WCS94, DOY98, SWC01, OE02, KIC04, FOT+05, ULdC05, CLT+06, lBQ08, RNR+10].

1.2 Estrutura da Tese

Esta dissertação tem seis capítulos e está organizada do seguinte modo:

No **Capítulo 1**, pode-se ficar a conhecer quais os objectivos e motivações desta dissertação. Este capítulo integra ainda o estado da arte e a estrutura organizacional deste documento.

O **Capítulo 2** apresenta os conceitos de natureza biológica que devem ser tidos em conta para o entendimento deste trabalho.

No **Capítulo 3** constam todas as explicações referentes aos procedimentos efetuados ao nível do processamento de imagem.

No **Capítulo 4** relata como foi feita toda a selecção e extracção de características a partir das imagens já processadas.

O **Capítulo 5** refere-se a toda a parte de classificação e aos algoritmos utilizados para o efeito.

O **Capítulo 6** contem as conclusões que se retiraram de todo o trabalho e que melhoramentos ou trabalhos futuros poderão ser desenvolvidos.

Após as conclusões, surge a **Bibliografia** onde podem ser consultadas todas as fontes que serviram de referência para este documento e finalmente os **Anexos** onde consta toda a parametrização dos classificadores.

Capítulo 2

Conceitos de Natureza Biológica

Este capítulo tem como objectivo apresentar uma breve descrição dos conceitos de natureza biológica necessários à compreensão de parte deste trabalho, nomeadamente, as características morfológicas da estrutura biológica da qual foram extraídas as imagens utilizadas futuramente, o Rim.

2.1 Morfologia do Rim

A unidade morfofuncional do rim designa-se por tubo urinífero e é o conjunto de todos os tubos uriníferos que forma o parênquima renal.

O tubo urinífero é formado por dois componentes: o nefrónio e o sistema de canais colectores.

O nefrónio, é constituído por: corpúsculo renal, glomérulo, cápsula de *Bowman*, tubo contornado proximal, ansa de Henle e tubo contornado distal.

O corpúsculo renal, ilustrado na Figura 2.1, é uma formação esférica constituída por um conjunto de ansas capilares, o glomérulo, situado no interior da chamada cápsula de *Bowman*, formada por duas membranas, uma interna, que envolve intimamente os capilares glomerulares e uma externa, separada da interna. Entre as membranas internas e externas existe uma cavidade denominada espaço capsular de *Bowman*, na qual se acumula o filtrado glomerular, ou urina primária.

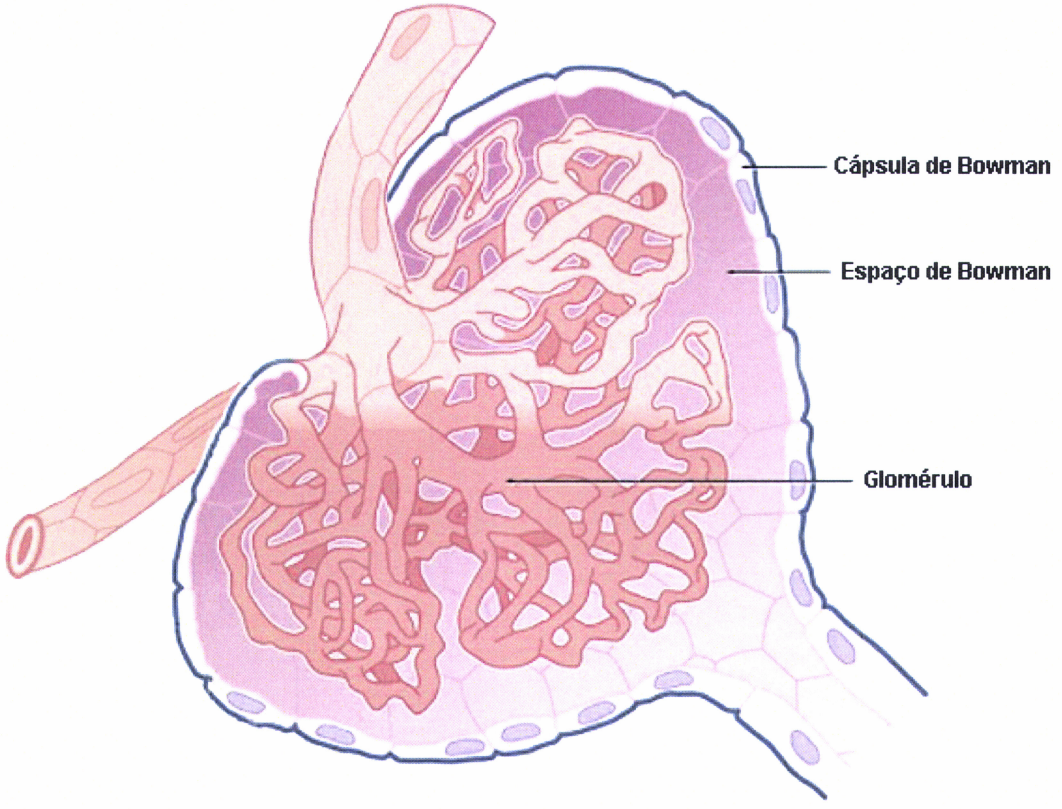


Figura 2.1: Corpúsculo renal

Capítulo 3

Processamento de Imagem

Na Ciência da Computação, o processamento de imagens refere-se ao processamento de imagens digitais através da utilização de computadores [GW01].

Pode-se então definir o processamento de imagens como um conjunto de técnicas computacionais que permitem realizar um vasto conjunto de operações sobre imagens. Dentro desse conjunto de operações destacam-se: o reconhecimento de padrões, a extracção e análise de informação, o restauro, o realce e a percepção de imagens.

Devido à evolução dos computadores e das tecnologias de processamento de sinal, esta área cresceu consideravelmente ao longo da última década [Pra07] e começou a ter um papel significativo em inúmeras aplicações relacionadas com a Ciência, a Indústria ou a Medicina.

3.1 Condições de Recolha das Imagens

Neste trabalho utilizaram-se imagens de corpúsculos renais de rins de rato obtidas no Instituto de Patologia Experimental, Faculdade de Medicina da Universidade de Coimbra que posteriormente foram cedidas ao Departamento de Biologia da Universidade de Évora.

As seguintes subsecções apresentam a descrição técnica dos procedimentos seguidos para a obtenção das imagens.

3.1.1 Materiais e Métodos

Utilizaram-se 21 ratos Wistar, machos, com 7 dias de idade. Durante o estudo, os animais foram alimentados com ração sintética completa e água, distribuídas em sistema *ad libitum*[1].

[1] à vontade

Os animais foram mantidos nas condições ambientais e de manutenção determinadas pela legislação em vigor na União Europeia, relativa à experimentação animal (FELASA, http://www.felasa.eu) e em Portugal (Portaria 1005/92).

Os ratos foram distribuídos aleatoriamente em três gaiolas do tipo IV, correspondentes a cada um dos seguintes grupos:

Grupo	N.° Animais	Classe
I	7	Controlo
II	8	Óleo de Milho
III	6	Thirame

Tabela 3.1: Distribuição dos animais por grupos

Os animais do Grupo I não sofreram qualquer manipulação; aos animais do Grupo III, foi administrado oralmente, duas vezes por semana, o ditiocarbamato Thirame [C6H12N2S4, CAS n.° 137-26-8, SIGMA], na dose de 100 $mg.kg^{-1}$ de peso corporal, sendo o veículo de administração óleo de milho, na quantidade de 0,1 ml; aos animais do Grupo II, foi administrado óleo de milho na mesma quantidade de 0,1 ml, em cada administração.

Em todas as administrações os animais foram pesados individualmente, correspondendo cada peso, em gramas, obtido em balança digital (0,01 g), à média de duas pesagens semanais.

Os animais foram sacrificados aos 35 dias de idade, por excesso de anestésico, individualmente, e na ausência dos outros animais.

Histopatologia

Após o sacrifício, removeram-se os rins, e após terem sido seccionados sagitalmente, foram fixados em formaldeído neutro a 10% , tamponado (pH 7.4), durante 24 horas. De seguida procedeu-se ao seu processamento pelas técnicas histológicas de rotina, em sistema automático: inclusão em parafina e corte em micrótomo rotativo, em secções com 5 μm de espessura. Os cortes foram estendidos em lâminas de vidro de 75 x 25 mm e corados com Hematoxilina e Eosina (H&E), para observação da estrutura geral.

Histomorfometria

As preparações definitivas foram observadas em microscópio fotónico Nikon Eclipse 600, com uma ampliação de 200X, sendo as imagens obtidas através de uma câmara digital Nikon DN100. Por cada animal de cada grupo, foi seleccionada

uma lâmina e em cada uma delas, observaram-se aleatoriamente dez corpúsculos renais, totalizando 210 imagens.

3.2 Ferramentas Utilizadas

Nesta secção são apresentadas as ferramentas informáticas utilizadas no processamento das imagens deste trabalho.

3.2.1 ImageJ

O ImageJ é um programa de processamento e análise de imagens desenvolvido em Java, por Wayne Rasband do *National Institute of Health* (NIH) [Ras09] que é capaz de exibir, editar, analisar, processar, gravar e imprimir imagens de vários formatos.

Este *software* permite calcular áreas, medir ângulos e distâncias em zonas definidas pelo utilizador e engloba funções de processamento de imagem *standard* tais como a manipulação de brilho, contraste, nitidez, suavização, *threshold* e detecção de bordas.

O ImageJ tolera ainda operações em vários factores de ampliação e efectua transformações geométricas como o redimensionamento, a rotação e o espelhamento de imagens. Além disso suporta o processamento de inúmeras imagens simultaneamente, estando limitado apenas pela memória disponível, e através de *stacks* ou "pilhas"de imagens, permite que a partir de uma única janela, várias imagens possam ser processadas em lote.

Por ter sido um *software* projectado com uma arquitetura aberta, é possível extender as suas funcionalidades recorrendo a *plugins* Java, que podem ser desenvolvidos pelo utilizador através do editor e compilador Java incorporados no programa, tornando possível a resolução de quase todo o tipo de problemas relacionados com o processamento e análise de imagens.

O facto de possuir todas estas funcionalidades, bastante úteis ao nível do processamento e análise de imagens, contribuiu para que o ImageJ tenha sido a ferramenta escolhida para o desenvolvimento desta fase da dissertação.

3.2.2 MultiCell Outliner

O *Multicell Outliner* (MCO), é um *plugin* para o ImageJ desenvolvido por Koldo Latxiondo, membro do P.A.S. Group (Procesado Avanzado de Señal) da Universidade de Deusto, Bilbao.

Este *plugin* utiliza a ferramenta *Magic Wand* para aplicar uma selecção num determinado ponto $P(x, y)$ da imagem com um *threshold* específico.

A ferramenta *Magic Wand* baseia-se no algoritmo *Flood Fill* [Wik] que faz uma pesquisa em todas as direcções a partir do ponto $P(x, y)$ para determinar se a cor dos pontos adjacentes corresponde à cor do ponto $P(x, y)$. Se a cor for correspondente ou estiver dentro de uma determinada tolerância, o ponto em questão é considerado na selecção. A comparação das cores é feita através do cálculo da diferença absoluta entre os componentes de cor *Red, Green and Blue* (RGB) do ponto e a cor escolhida. No caso de coordenadas RGB com 8 bits, a cor do ponto i corresponde à cor do ponto de referência se:

$$\max(|R_{ref} - R_i|, |G_{ref} - G_i|, |B_{ref} - B_i|) < Th \tag{3.1}$$

Onde:

- $R_{ref}, G_{ref}, B_{ref}$ são os valores das componentes vermelha, verde e azul no ponto de referência
- R_i, G_i, B_i são os valores das componentes vermelha, verde e azul no ponto i
- Th é a tolerância definida

A tolerância determina o quanto uma cor é próxima da cor escolhida. Deste modo, se o seu valor for 0, então só são escolhidos os pontos adjacentes em que a cor corresponde exactamente à cor escolhida, se for 255, são escolhidos todos os pontos da imagem [Fin07].

O MCO possibilita ainda a gravação de regiões de interesse em múltiplas imagens, para que possam ser processadas posteriormente, o que se revelou bastante útil na fase inicial de segmentação das imagens [Lat06]. A interface deste *plugin* pode ser vista na Figura 3.1.

3.3 Segmentação de Imagem

3.3.1 Introdução

Além da diversidade que a Biologia porporciona, a forma como o corpúsculo renal é cortado também afecta o aspecto que a imagem terá em duas dimensões, podendo em algumas situações surgir glomérulos com uma forma circular bem definida e sem interrupções, e noutras glomérulos com um aspecto mais disforme e com quebras.

Para que o conteúdo das imagens utilizadas seja perceptível, veja-se a Figura 3.2, que demonstra os tipos de corte que o corpúsculo renal pode sofrer e as imagens que originam, respectivamente.

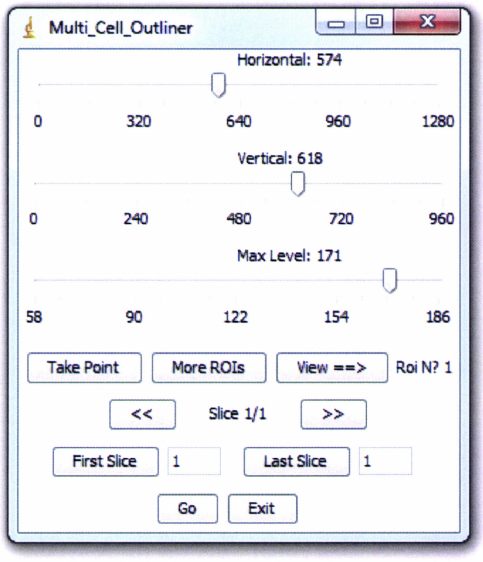


Figura 3.1: Interface do *plugin* MultiCell Outliner

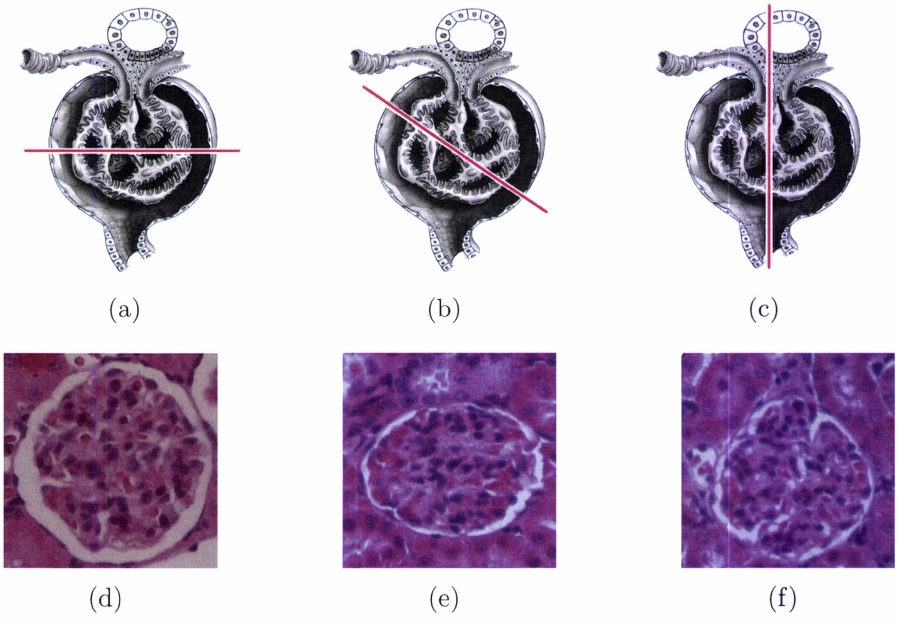


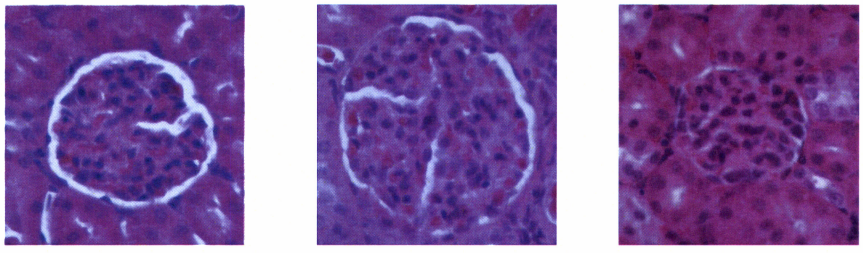
Figura 3.2: Tipos de corte num corpúsculo renal e imagens resultantes

A segmentação de imagem consiste na divisão de uma imagem em regiões ou objectos, segundo um critério [GW01] e tem dois objectivos: decompor a imagem em partes extraíndo somente as regiões que se irão analisar e alterar a forma de

representação da imagem para que a análise futura seja facilitada [SS01].

Para que fosse mais fácil expôr o trabalho realizado neste capítulo, que será de seguida explicado em detalhe, dividiu-se este processo em duas partes: a delimitação e a extracção dos glomérulos.

É de referir que devido à enorme variabilidade e complexidade deste tipo de imagens, como pode ser verificado através da Figura 3.3, não foi possível efectuar este trabalho de uma forma completamente automática, pelo que, para alcançar o resultado desejado foi necessário conjungar a utilização do *plugin* MCO com algumas ferramentas e operações do ImageJ, podendo dizer-se que esta tarefa acabou por ser realizada de forma semi-automática.



(a) baixo (b) intermédio (c) elevado

Figura 3.3: Glomérulos com diferentes graus de complexidade na segmentação

3.3.2 Delimitação dos Glomérulos

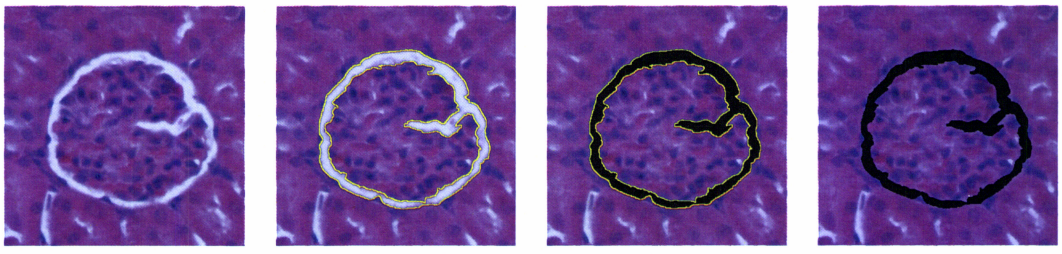
A delimitação dos glomérulos teve como objectivo definir e preencher o contorno de toda a sua estrutura. Embora semelhantes, existiram três tipos de abordagem, consoante os diferentes graus de complexidade das imagens.

Grau de complexidade baixo

Nos casos de baixa complexidade (Fig. 3.3(a)) conseguiu-se delimitar o corpúsculo renal, totalmente e sem dificuldades, utilizando somente o MCO com apenas um ponto da imagem e um valor de *threshold*. Após a selecção, pintou-se o glomérulo com a ferramenta *Fill*, como é visível na Figura 3.4(c). O resultado final da delimitação pode ser visto na Figura 3.4(d).

Grau de complexidade intermédio

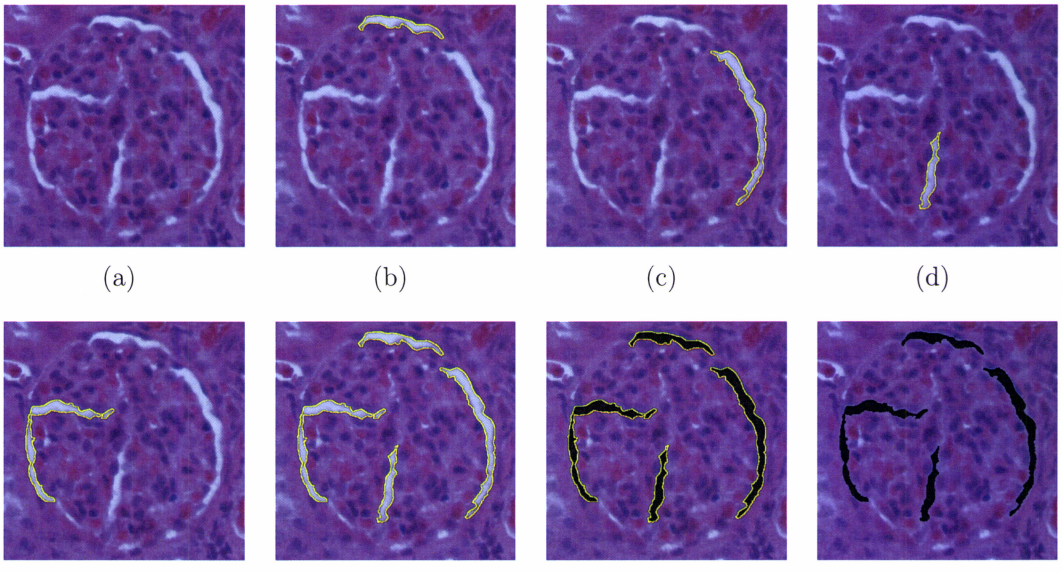
Nos casos de complexidade intermédia (Fig. 3.3(b)), só se conseguiu delimitar o glomérulo parcialmente com a ajuda do MCO e utilizando vários pontos da



(a) (b) (c) (d)

Figura 3.4: Delimitação de um glomérulo com baixo grau de complexidade

imagem com diferentes valores de *threshold*. Após o destaque parcial pintaram-se essas regiões, como se vê na Figura 3.5.



(a) (b) (c) (d)

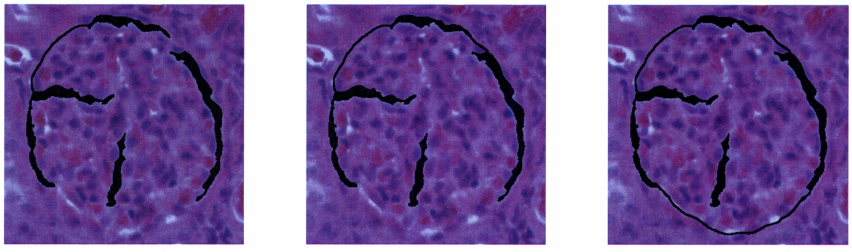
(e) (f) (g) (h)

Figura 3.5: Delimitação parcial de um glomérulo com grau de complexidade intermédio utilizando o MCO

Para completar o processo, foi necessário demarcar o restante contorno do glomérulo manualmente, fazendo uso da ferramenta de pintura *brush* com espessura de 3 *píxel*.

Grau de complexidade elevado

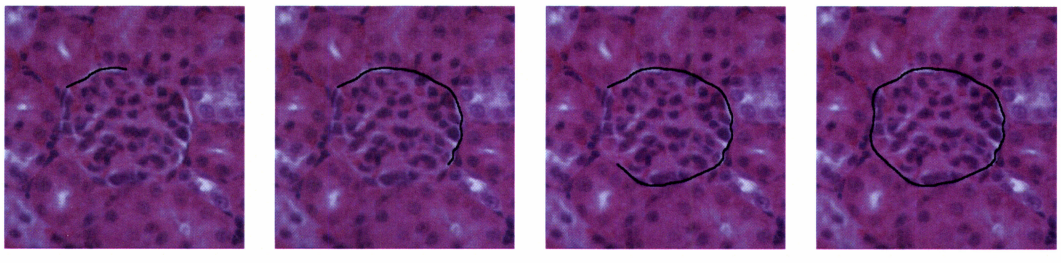
Nas situações de elevado grau de complexidade, mesmo escolhendo vários pontos com diferentes valores de *threshold*, o *plugin* não obteve resultados satisfatórios



(a) (b) (c)

Figura 3.6: Conclusão manual da delimitação parcial de um glomérulo com grau de complexidade intermédio

e o glomérulo teve de ser delimitado à mão na sua totalidade, recorrendo também à ferramenta *brush* com 3 *píxel* de espessura.



(a) (b) (c) (d)

Figura 3.7: Delimitação de um glomérulo com grau de complexidade elevado

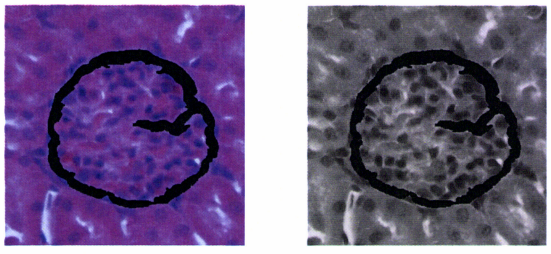
3.3.3 Extracção dos Glomérulos

No processo de extracção dos glomérulos, o objectivo foi isolar as regiões demarcadas em 3.3.2. Para isso utilizou-se apenas uma abordagem para todas as imagens.

Após completar a delimitação dos glomérulos, foi necessário preparar as imagens para que se conseguisse extrair as suas silhuetas. A técnica utilizada, foi converter todas as imagens do formato RGB para o formato *Grayscale* (escala de cinzentos).

Ao efectuar esta conversão (Fig. 3.8), cada *píxel* da imagem deixa de guardar cor e passa apenas a guardar uma intensidade de cinzento, sendo a intensidade mais forte o preto e a mais fraca o branco.

Às imagens resultantes desta conversão pode-se então aplicar uma técnica bastante utilizada em segmentação de imagem, a limiarização ou *thresholding*.



(a) (b)

Figura 3.8: Conversão do formato RGB para o formato *Grayscale*

Esta técnica permite que se transforme uma imagem com o formato *grayscale* numa imagem binária, isto é, a preto e branco [SS01]. Para isso, é definido um valor l, o limiar, e todos os *píxeis* com valores de cinza abaixo de l passam a 0 ficando com a cor branca e valores acima passam a 1 ficando com a cor preta.

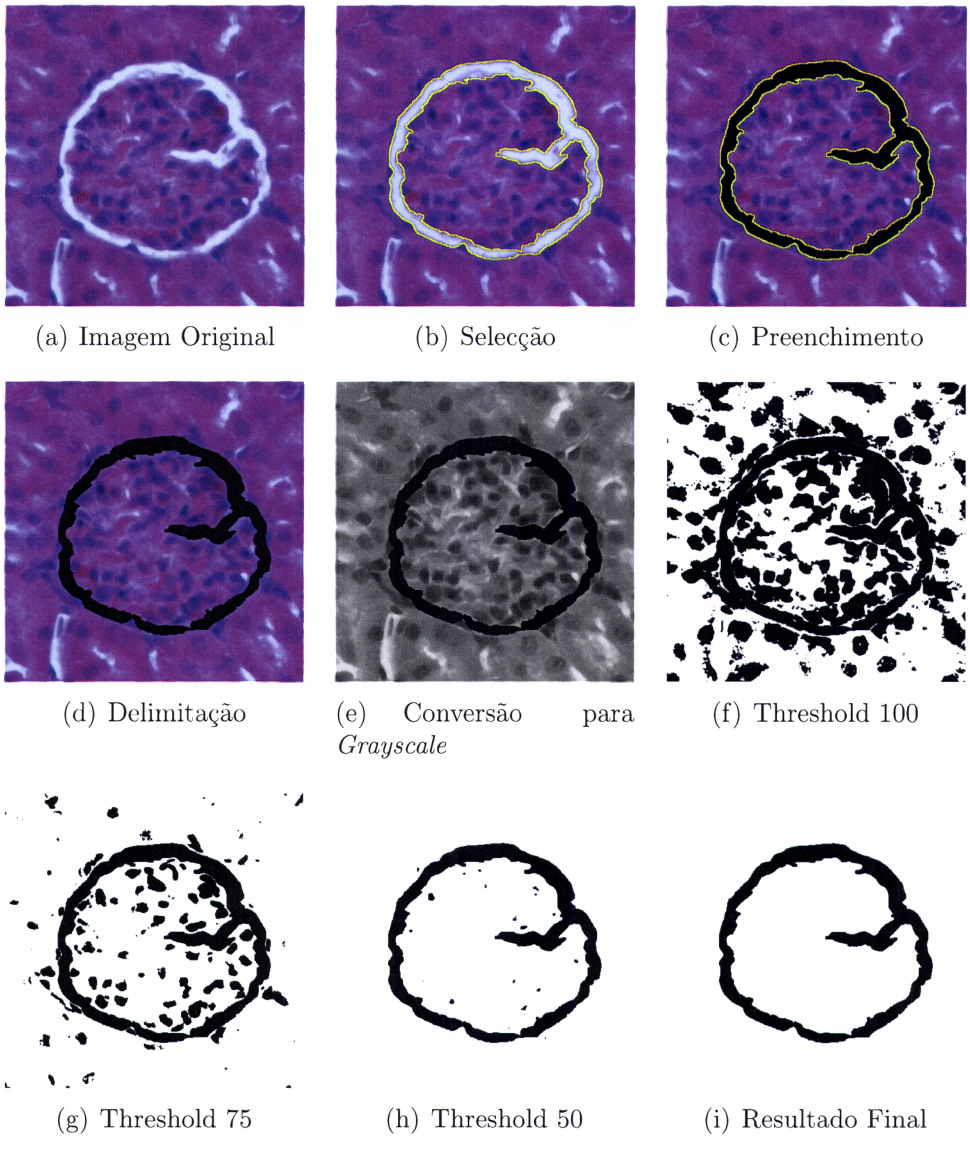
Aplicando a limiarização à imagem da Figura 3.8(b), é possível destacar-se a silhueta do glomérulo das restantes partes da imagem que não constituem interesse para análise, como se verifica na Figura 3.9.

(a) (b) (c) (d)

Figura 3.9: Aplicação de diferentes valores *threshold*

Este passo encerra o processo de segmentação de imagem uma vez que se chegou ao objectivo desejado. Como o processo foi explicado de forma faseada e para que se tenha uma melhor percepção do processo no seu todo, veja-se a Figura 3.10 que, utilizando a imagem de um glomérulo de baixa complexidade, mostra todas as etapas de segmentação de um glomérulo, partindo da sua imagem original e chegando à imagem pretendida.

Depois de extraídos todos os glomérulos das imagens originais e uma vez que o tamanho absoluto das imagens resultantes era variável, aplicou-se uma máscara cujo tamanho é função do maior dos corpúsculos. Esta medida permitiu que todas as imagens ficassem com o mesmo tamanho absoluto não alterando o tamanho relativo dos glomérulos, o que facilitou o trabalho realizado no capítulo seguinte.



(a) Imagem Original (b) Selecção (c) Preenchimento

(d) Delimitação (e) Conversão para *Grayscale* (f) Threshold 100

(g) Threshold 75 (h) Threshold 50 (i) Resultado Final

Figura 3.10: Processo completo de segmentação da imagem de um corpúsculo renal

É ainda de destacar que o resultado da segmentação foi um conjunto com 220 imagens de glomérulos, apesar de no conjunto inicial constarem apenas 210. Este acréscimo no número de imagens deve-se ao facto de algumas imagens possuírem mais do que um glomérulo.

Capítulo 4

Extracção de Características

O objectivo da extracção de características, é caracterizar um objecto através de atributos cujos valores sejam semelhantes para objectos pertencentes à mesma categoria e distintos para objectos pertencentes a uma categoria diferente [DHS01].

Visto que um objecto pode ter uma infinidade de características este processo torna-se complicado, pois pode surgir a tentação de considerar uma vasta colecção de atributos acabando por dar a ilusão de que teremos mais informação relevante disponível. O que se passa nessa situação é que, inicialmente, o desempenho até pode ser bom mas à medida que o número atributos considerados aumenta, surge o problema da complexidade computacional e o desempenho acaba por se ir detrioriando [ETPZ09, Man98].

Uma forma de resolver este problema passa por seleccionar um pequeno conjunto de atributos que permita caracterizar e distinguir os objectos da melhor forma possível, daí que seja essencial ter um bom conhecimento no domínio dos objectos.

A extracção de características pode ser uma tarefa complicada sendo bastante importante que seja efectuada correcta e cuidadosamente, já que é um passo crítico que mais tarde terá influência na classificação.

4.1 Descrição de Atributos

Neste trabalho, os objectos alvo da extracção de características foram os glomérulos renais. No início, não havia uma ideia clara relativamente aos atributos que poderiam ser relevantes para descrever e diferenciar estas estruturas. Pensou-se que, além da forma, poderia ser útil considerar intensidades de cor ou brilho dos *píxeis*, porém, chegou-se à conclusão que a inclusão desses atributos iria aumentar a complexidade computacional não justificando o ganho de informação. Além desse aspecto, o facto de a fase de segmentação ter tido como resultado imagens binárias, fez com que apenas fizesse sentido considerar as características morfológicas do glomérulo.

Tendo em conta todos estes factos, o resultado da extracção de características foi um vector composto por 13 atributos que serão descritos abaixo.

Nome – Identificador único utilizado apenas para distinguir as imagens umas das outras.

Diâmetro – Distância máxima entre dois pontos pertencentes ao corpúsculo renal.

Perímetro – Perímetro exterior do corpúsculo, ou seja, o perímetro da cápsula de *Bowman*.

Área Capsular – Representa o espaço capsular de *Bowman*.

Área Interior – Toda área interior do glomérulo, excluíndo a área capsular mas incluíndo os espaços entre os capilares.

Área Total – Todo o espaço interior ao perímetro da cápsula de *Bowman*.

Área de Espaços Interiores – Área ocupada pelos espaços isolados existentes no interior dos capilares.

Área de Vasos – Toda a área interior, excluíndo os espaços interiores.

Área Vazia – Todos os espaços existentes no corpúsculo renal, ou seja, é a soma do espaço capsular com os espaços interiores.

Fractais e Dimensão Fractal

Pode-se dizer que os fractais são formas geométricas não-euclideanas caracterizadas pela auto-semelhança e complexidade infinita [Man83], ou seja, se essa forma geométrica for infinitamente subdividida em partes, cada uma delas será uma cópia reduzida, exacta ou aproximada, da forma original. Essa auto-semelhança pode ser traduzida matematicamente por um coeficiente D, a que se chama dimensão fractal. Esta dimensão, diferente da habitual dimensão topológica (que é um número inteiro), serve para caracterizar o fractal indicando o seu grau de irregularidade. Uma vez que os glomérulos podem ser considerados fractais, consideraram-se ainda os seguintes atributos:

Dimensão Fractal – Grau de irregularidade considerando todo o glomérulo, isto é, tendo em conta o perímetro exterior e interior da cápsula de *Bowman*.

Dimensão Fractal Exterior – Grau de irregularidade do perímetro da cápsula de *Bowman*.

Dimensão Fractal Interior – Grau de irregularidade do perímetro da área interior do corpúsculo.

Classe – Identificador do grupo de ratos a que pertence uma imagem: Controlo, Óleo de Milho ou Thirame.

4.2 Processos de Cálculo e Medição

Após a definição e descrição do conjunto de atributos que foram considerados relevantes para este trabalho, foi necessário proceder ao cálculo e medição dos valores para cada um desses atributos.

Para facilitar a explicação dos procedimentos, os atributos foram separados em dois grupos: no primeiro grupo estão os atributos que foram medidos ou calculados de forma directa e no segundo grupo os atributos que foram calculados a partir de atributos pertencentes ao primeiro grupo.

Primeiro Grupo

Os atributos pertencentes a este primeiro grupo são: o diâmetro, o perímetro, a área interior, a área total, a área de espaços interiores, a dimensão fractal, a dimensão fractal exterior e a dimensão fractal interior.

Os primeiros cinco atributos deste grupo foram calculados utilizando uma ferramenta de medição incorporada no *software* ImageJ. Esta ferramenta permite que, através da selecção de uma região da imagem, o utilizador escolha uma série de grandezas que deseja ver medidas ou calculadas sobre essa região.

Pela necessidade de se ter de medir e calcular vários atributos referentes a diferentes regiões do objecto, efectuaram-se três tipos distintos de selecção na imagem do binária do corpúsculo renal, como se pode visualizar na seguinte figura:

(a) Exterior (b) Interior (c) Espaço Interior

Figura 4.1: Selecções utilizadas nos procedimentos de cálculo e medição dos atributos

Cada uma destas selecções permitiu medir e/ou calcular um ou mais atributos. O perímetro, o diâmetro e a área total do corpúsculo renal, ilustrados nas Figuras 4.2(a), 4.2(b) e 4.2(c), foram calculados utilizando a selecção exterior, visível na Figura 4.1(a).

(a) Diâmetro (b) Perímetro (c) Área total

Figura 4.2: Grandezas medidas com base na selecção exterior de um glomérulo

A área interior e a área de espaços interiores, ilustradas na Figura 4.3(a) e 4.3(b), foram calculadas a partir da selecção interior, presente na Figura 4.1(b) e da selecção de espaços interiores, presente na Figura 4.1(c). Note-se que esta última

selecção só se justifica perante a existência de espaços no interior do corpúsculo renal.

(a) Área interior (b) Espaços interiores

Figura 4.3: Grandezas medidas com base na selecção interior e selecção de espaços interiores de um glomérulo

Utilizando estas três selecções, escolheu-se quais as grandezas a medir a partir de cada uma delas através de um *menu*, presente na Figura 4.4, e procedeu-se ao seu cálculo fazendo uso do comando *Measure*.

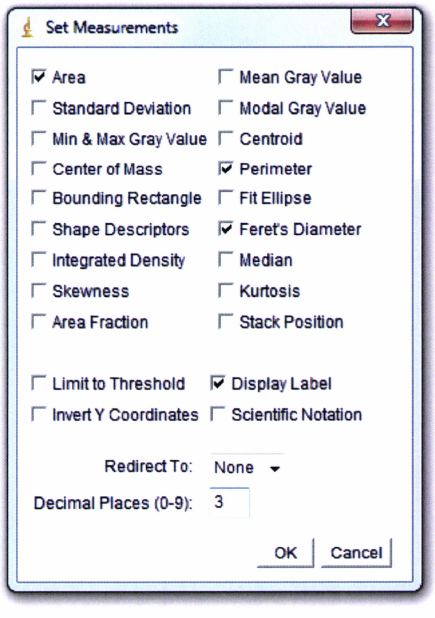


Figura 4.4: Escolha dos valores a medir utilizando a selecção exterior de um corpúsculo renal

Após a execução desse comando, surge uma janela onde constam os resultados

das medições para a selecção em questão, como se pode visualizar na Figura 4.5.

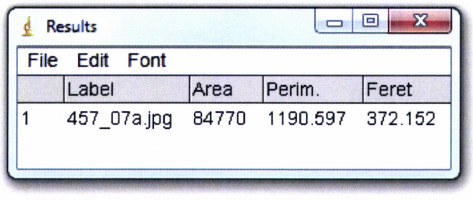


Figura 4.5: Resultado das medições efectuadas utilizando a selecção exterior de um corpúsculo renal

Fractal Box Count

No cálculo das dimensões fractais foi utilizado o algoritmo *Fractal Box Count* (FBC), existente no ImageJ.

Este algoritmo sobrepõe sobre o fractal uma grelha quadriculada, onde cada quadrícula tem um tamanho t. Posteriormente conta o número de quadrículas, N, interceptadas pelo fractal e calcula a dimensão fractal, D, através da seguinte expressão:

$$D = \log(N)/\log(\frac{1}{t}) \tag{4.1}$$

O processo é repetido para vários tamanhos de quadrículas (Fig. 4.6) e é construído um gráfico que relaciona $\log(N)$ com $\log(\frac{1}{t})$, onde é traçada a recta que melhor se ajusta a todos os valores obtidos (Fig. 4.7). A dimensão fractal traduz-se pelo declive dessa recta.

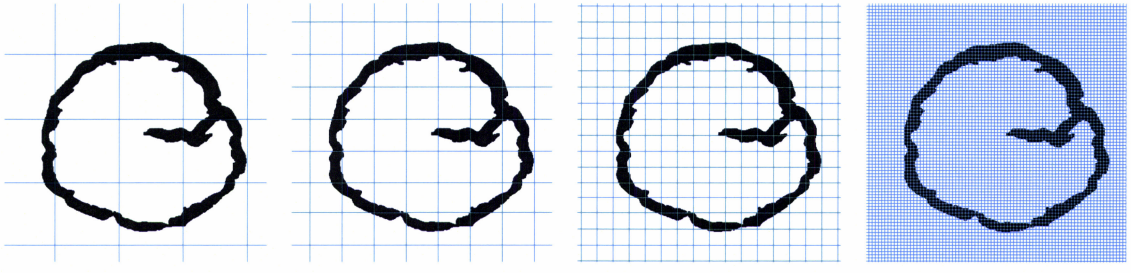


Figura 4.6: Quadrículas de diferentes tamanhos utilizadas no cálculo da dimensão fractal

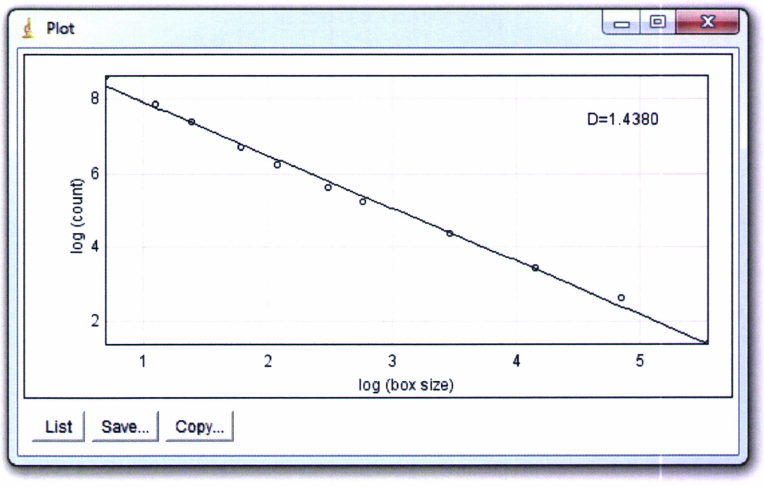


Figura 4.7: Relação entre o número e o tamanho de quadrículas no cálculo da dimensão fractal

Neste trabalho, foi aplicado o FBC com quadrículas de tamanho 2, 3, 4, 6, 8, 12 , 16, 32, 64, 128 e 256.

Apesar do algoritmo de cálculo da dimensão fractal ser o mesmo para todas as dimensões fractais, cada uma delas foi calculada sobre uma zona da imagem.

Para calcular a dimensão fractal, aplicou-se o FBC sobre a imagem do corpúsculo renal sem que esta tenha sofrido quaisquer tranformações.

No caso das dimensões fractais exterior e interior foram desenvolvidas duas pequenas macros: uma com objectivo de aplicar as tranformações necessárias às imagens para destacar o perímetro exterior do corpúsculo renal e outra com o mesmo propósito mas destacando o perímetro interior.

As transformações efectuadas pela primeira macro, presentes na Figura 4.8, consistiram em pintar o interior do corpúsculo para que, ao calcular a dimensão fractal, só seja considerado o perímetro exterior da estrutura.

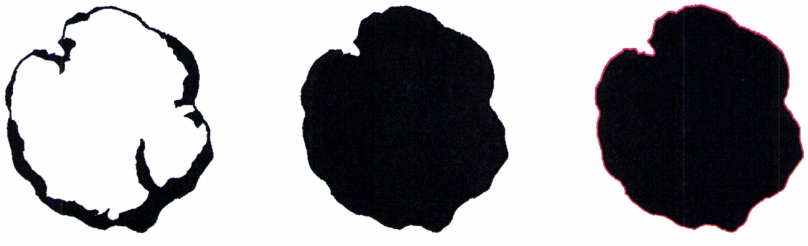


Figura 4.8: Transformações aplicadas ao corpúsculo renal para o cálculo da dimensão fractal exterior

As transformações executadas pela segunda macro, visíveis na Figura 4.9, consistiram em pintar a área exterior ao corpúsculo renal e aplicar uma inversão de cores à imagem para que, ao calcular a dimensão fractal, só o perímetro interior da estrutura seja considerado.

Figura 4.9: Transformações aplicadas ao corpúsculo renal para o cálculo da dimensão fractal interior

Note-se que, caso o corpúsculo renal possua espaços interiores, estes são ignorados no cálculo das dimensões fractais.

Segundo Grupo

Os atributos pertencentes ao segundo grupo são: a área capsular, a área de vasos e a área vazia. Como foi referido anteriormente, estes atributos foram calculados a partir de atributos pertencentes ao primeiro grupo através de formulas matemáticas existentes nas folhas de cálculo *Excel*.

A área capsular, que pode ser vista na Figura 4.10(a), obteve-se subtraíndo a área interior à área total. A área de vasos, ilustrada na Figura 4.10(b), calculou-se subtraíndo os espaços interiores à área interior e a área vazia, visível na Figura 4.10(c), foi calcula somando a área capsular com os espaços interiores.

(a) Área capsular (b) Área de vasos (c) Área vazia

Figura 4.10: Grandezas calculadas a partir de atributos existentes

No que diz respeito às unidades dos atributos: as áreas surgem em px^2, o perímetro e diâmetro em px e as dimensões fractais são adimensionais.

Finalizando este capítulo é conveniente deixar claro que, por motivo de simplificação, as explicações acima estão direccionadas apenas para a imagem de um corpúsculo renal, no entanto, estes procedimentos foram aplicados em lote a todo o conjunto de imagens.

O resultado deste processo foi um ficheiro *Excel* contendo os valores de cada atributo para cada uma das 220 imagens dos corpúsculos renais.

Capítulo 5

Classificação

A classificação é a tarefa que consiste na atribuição de objectos a uma de várias classes pré-definidas [TSK05]. É uma tarefa que pode ser aplicada em problemas das mais diversas áreas, como por exemplo: a detecção de *spam* em mensagens *e-mail* no campo da Informática, ou a categorização de células no campo da Biologia.

Os problemas de classificação recebem como dados de entrada um conjunto de instâncias. Cada instância, também conhecida como amostra, objecto ou exemplo, é caracterizada por um par (X, y), onde X é um conjunto de atributos e y é um atributo especial que designa a classe ou categoria a que a instância pertence.

Como exemplo, vejamos a Tabela 5.1, que exibe um conjunto de dados para classificação do estado de saúde de um paciente com base nos seus sintomas. Cada linha da tabela representa uma instância do conjunto de dados e cada coluna representa um atributo. O atributo *Diagnóstico* é considerado especial, pois designa a classe a que cada instância pertence, ou seja, designa se cada paciente é saudável ou doente.

Paciente	Febre	Enjoo	Manchas	Dor	Diagnóstico
P1	40.0	sim	pequenas	sim	doente
P2	36.5	não	grandes	não	saudável
P3	37.6	sim	pequenas	não	saudável
P4	38.3	não	grandes	sim	doente
P5	36.9	não	pequenas	sim	saudável
P6	37.1	não	grandes	sim	doente
P6	36.2	não	pequenas	não	saudável

Tabela 5.1: Conjunto de dados para diagnóstico de pacientes

Formalmente, a classificação pode ser definida como a construção ou aprendizagem de uma função $f : X \rightarrow Y$ tal que,

$$f : x_i \in X \rightarrow y_j = f(x_i)$$

Onde:

- $X = \{x_1, x_2, \ldots, x_n\}$ é um conjunto de instâncias;
- $Y = \{y_1, y_2, \ldots, y_m\}$ é um conjunto de classes;
- $1 \leq i \leq n$ e $1 \leq j \leq m$;
- f é a função que associa uma classe a uma instância;

A função f é também conhecida como modelo de classificação. Define-se como técnica de classificação ou classificador, a abordagem sistemática para a construção de modelos de classificação a partir de um conjunto de dados [TSK05]. Cada classificador utiliza o seu algoritmo de aprendizagem para identificar o modelo de classificação que melhor se ajusta na relação entre o conjunto de atributos das instâncias e a classe a que está associado.

Em classificação, são várias as técnicas que permitem construir modelos para a classificação de dados, como é o caso: das árvores de decisão, das regras de associação, dos métodos bayesianos, das redes neuronais ou das *support vector machines.*

Os modelos de classificação podem ser: descritivos, se ajudam a explicar como é feita a distinção dos dados por diferentes classes, ou preditivos, se ajudam a descobrir a classe para um objecto em que esta é desconhecida [TSK05].

Neste trabalho focámo-nos no modelo preditivo. A Figura 5.1 mostra a abordagem geral tomada na resolução deste tipo de problemas: em primeiro lugar, é utilizado um conjunto de instâncias onde as classes que lhes estão associadas são conhecidas, designado de conjunto de treino; de seguida, com base no conjunto de treino, é construído o modelo de classificação; e finalmente, o modelo construído é aplicado sobre o conjunto de teste, isto é, um conjunto de novas instâncias das quais se desconhece a classe. O objectivo final é construir um modelo capaz de prever a classe para novos exemplos.

O modelo de classificação gerado por um classificador, deve adaptar-se tanto aos dados já conhecidos como a dados que lhe são desconhecidos, como tal é fundamental que o algoritmo de aprendizagem utilizado na sua construção permita a melhor capacidade de generalização possível [TSK05, WF05].

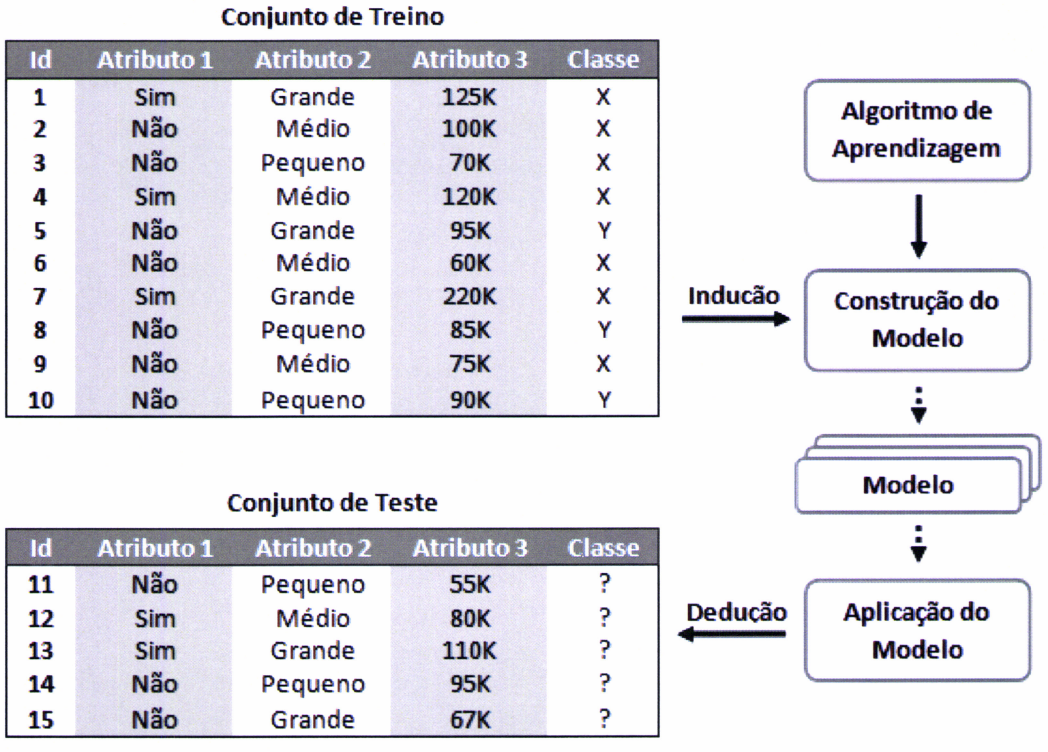


Figura 5.1: Abordagem geral para a construção de um modelo de classificação

5.1 Avaliação de Classificadores

Embora existam várias técnicas de aprendizagem, não há uma que apresente o melhor desempenho em todos os problemas. Assim, é importante compreender os pontos fortes e a limitação dos diferentes classificadores utilizando alguma metodologia de avaliação que permita comparar algoritmos.

A avaliação de classificadores não é trivial. Não só depende do desempenho do modelo de classificação, como também do tipo e da quantidade dos dados a classificar [WF05], razão pela qual o método de amostragem dos dados utilizados na indução do classificador se torna um factor importante.

A ideia dos métodos de amostragem, é a formação de dois conjuntos de dados disjuntos: o conjunto de treino e o conjunto de teste. Uma instância pertencente ao conjunto de treino, utilizado na aprendizagem, não deve pertencer ao conjunto de teste, utilizado na avaliação do desempenho do classificador. A utilização dos mesmos dados no treino e teste do classificador, faz com que o modelo de classificação gerado possua um desempenho optimista face a dados desconhecidos [WF05].

Assim, os métodos de amostragem permitem estimar a capacidade de generalização de um classificador de uma forma mais fiável. Entre os métodos de amostragem, os mais conhecidos são [WF05, HK05, TSK05]:

Holdout

Este método divide o conjunto de dados em dois conjuntos, um para o treino e outro para o teste do classificador, como ilustra a Figura 5.2. O conjunto de treino contém uma porção p dos dados e o conjunto de teste contém o restante, $1 - p$. Resultados empíricos demonstram que $p = \frac{2}{3} \wedge (1 - p) = \frac{1}{3}$, geralmente, geram um modelo de classificação aceitável.

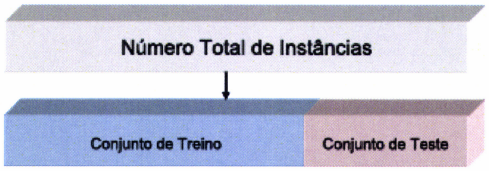


Figura 5.2: Método *Holdout*

Existe ainda uma variação do método *Holdout*, onde o mesmo é aplicado k vezes. Em cada iteração, é seleccionada aleatoriamente uma determinada porção dos dados para treino, ficando a restante porção para teste, como se pode visualizar na Figura 5.3.

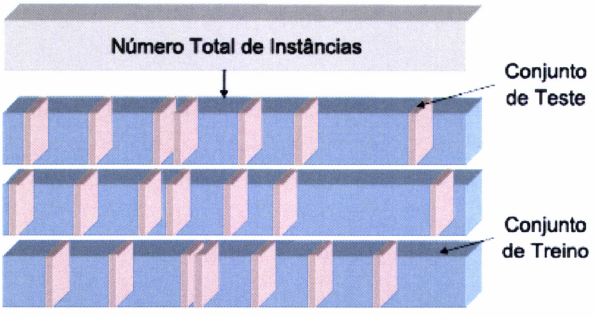


Figura 5.3: Variação do método *Holdout*

Bootstrap

Este método repete o processo de classificação várias vezes utilizando amostragem com reposição para formar o conjunto de treino. Em cada iteração, são

retiradas do conjunto original n amostras aleatórias com reposição, que formam um conjunto de treino com o mesmo número de instâncias que o conjunto original. As instâncias que não fazem parte do conjunto de treino, formam o conjunto de teste. A Figura 5.4 demonstra o processo.

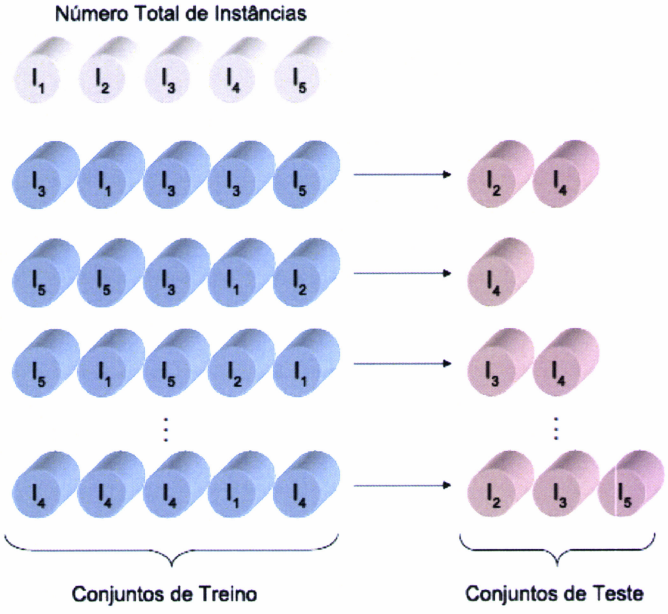


Figura 5.4: Método *Bootstrap*

Cross Validation

Este método divide o conjunto de instâncias em partições disjuntas, denominadas *folds*. O número de *folds* utilizado varia consoante a quantidade de instâncias e a proporção de instâncias pertencentes a cada classe. Para validação cruzada com k *folds*, o conjunto de instâncias é dividido em k *folds* de tamanho igual, sendo que $(k-1)$ *folds* são utilizados para treino e o *fold* restante utilizado para teste. O processo repete-se k vezes, até que todos os *folds* tenham sido utilizados para teste. A Figura 5.5 ilustra o funcionamento deste método.

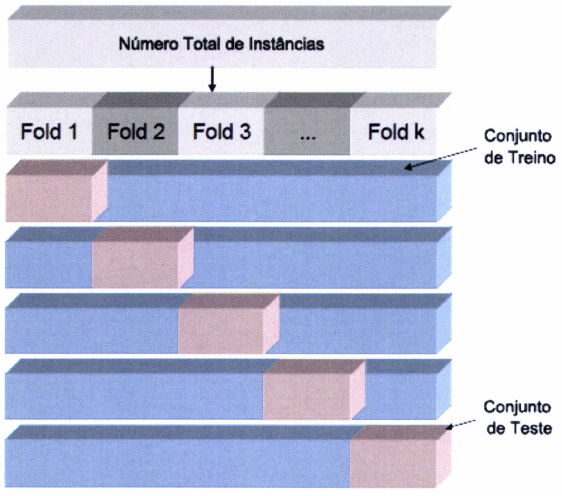


Figura 5.5: Método *Cross Validation*

Existe também um caso específico de validação cruzada denominado *Leave-One-Out*. Este método utiliza validação cruzada com k *folds* em que k é igual ao número total de instâncias. Neste caso, cada conjunto de teste é formado por 1 única instância e o conjunto de treino é formado pelas restantes $(k-1)$ instâncias.

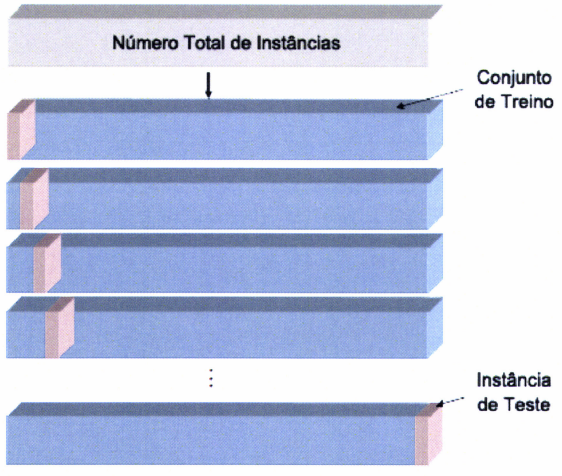


Figura 5.6: Método *Leave-One-out*

Esta abordagem tem como vantagem a utilização do maior número possível de instâncias no conjunto de treino. Em contrapartida, tem um elevado custo computacional, uma vez que se repete k vezes para que todas as instâncias sejam utilizadas uma vez para teste.

Após a escolha do método de amostragem, deve-se definir o método de avaliação do desempenho do classificador. A forma mais frequente para medir o desempenho de um classificador, baseia-se no número de predições correctas e incorrectas efectuadas.

Os resultados da classificação são expostos numa tabela, denominada matriz de confusão. A Tabela 5.2 mostra o formato de uma matriz de confusão para um problema de classificação binária, isto é, onde as instâncias podem ser classificadas como pertencentes às classes positiva ou negativa. As linhas da tabela indicam a classe a que a instância realmente pertence e as colunas indicam a classe segundo a qual a instância foi classificada.

	Classe Predita	
Classe Verdadeira	Positiva	Negativa
Positiva	VP	FN
Negativa	FP	VN

Tabela 5.2: Exemplo de uma matriz de confusão

As siglas (VP, VN, FP, FN) utilizadas na Tabela 5.2 correspondem:

- Verdadeiros Positivos (VP): total de instâncias de teste preditas como positivas e que realmente são positivas;
- Verdadeiros Negativos (VN): total de instâncias de teste preditas como negativas e que realmente são negativas;
- Falsos Positivos (FP): total de instâncias de teste preditas como positivas mas que na verdade são negativas;
- Falsos Negativos (FN): total de instâncias de teste preditas como negativas mas que na verdade são positivas;

Utilizando a matriz de confusão é possível calcular várias medidas que permitem ajudar na avaliação de classificadores. A soma dos valores que constam na diagonal principal da matriz, representa o número de instâncias correctamente classificadas e a soma dos restantes valores, o número de instâncias incorrectamente classificadas. Estes números permitem o cálculo das taxas de acerto e de erro do classificador.

A taxa de acerto, T_a, e a taxa de erro, T_e, indicam a taxa de instâncias correctamente classificadas e a taxa de instâncias incorrectamente classificadas, respectivamente. Estas taxas podem ser calculadas através das seguintes expressões:

$$T_a = \frac{VP + VN}{VP + VN + FP + FN} \tag{5.1}$$

$$T_e = 1 - T_a \tag{5.2}$$

A taxa de verdadeiros positivos, T_{VP}, também conhecida como *recall*, estima a probabilidade de uma instância pertencente à classe positiva ser classificada como positiva. A taxa de falsos positivos, T_{FP}, estima a probabilidade de uma instância pertencente à classe negativa ser classificada como positiva e a taxa de verdadeiros negativos, T_{VN}, estima a probabilidade de uma instância pertencente à classe negativa ser classificada como negativa. O cálculo destas medidas é feito através das seguintes expressões:

$$T_{VP} = \frac{VP}{VP + FN} \tag{5.3}$$

$$T_{FP} = \frac{FP}{FP + VN} \tag{5.4}$$

$$T_{VN} = \frac{VN}{VN + FP} = 1 - T_{FP} \tag{5.5}$$

A precisão, p, estima a probabilidade da predição de uma classe positiva estar correcta e é dada pela expressão:

$$p = \frac{VP}{VP + FP} \tag{5.6}$$

Finalmente, existe ainda a $F_{Measure}$ que relaciona a precisão e o *recall* e é dada pela média harmónica ponderada das duas medidas:

$$F_{Measure} = \frac{2}{\frac{1}{p} + \frac{1}{T_{VP}}} = \frac{2 \times p \times T_{VP}}{p + T_{VP}} \tag{5.7}$$

Além destas medidas, também existem as curvas *ROC* (*Receiver Operator Characteristics*), que têm vindo a ser bastante utilizadas na avaliação de classificadores.

As curvas ROC são gráficos que relacionam a taxa de VP com a taxa de FP. Este tipo de curva permite visualizar o desempenho de um classificador através da área abaixo da curva (*AUC*, *Area Under Curve*). A AUC varia entre 0 e 1 e quanto maior for, melhor é o desempenho do classificador. A Figura 5.7 representa uma curva ROC e a sua AUC.

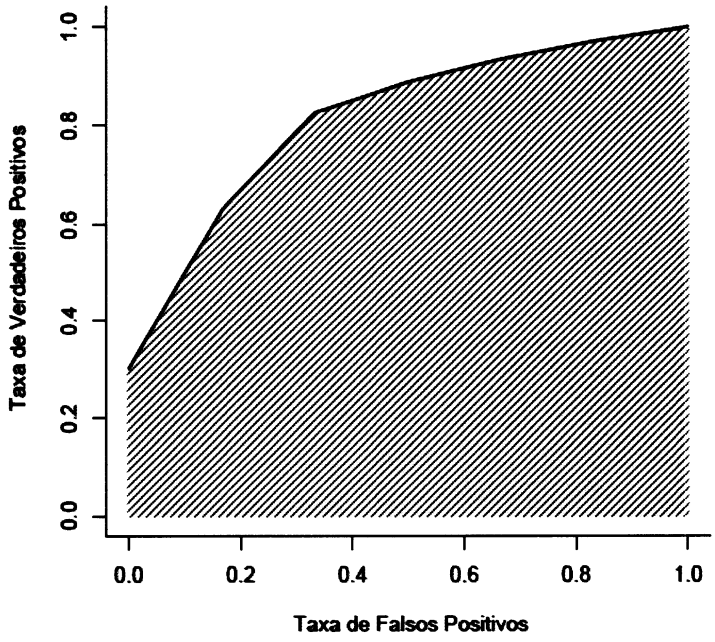


Figura 5.7: Exemplo de uma curva ROC

Uma vez introduzidos os conceitos teóricos necessários, este capítulo prossegue com a exposição das ferramentas utilizadas, a explicação do funcionamento dos algoritmos utilizados e finalmente, a apresentação dos resultados obtidos.

5.2 Ferramentas Utilizadas

5.2.1 WEKA

O *Waikato Environment for Knowledge Analysis* (WEKA), é um programa que possui uma colecção de algoritmos de Aprendizagem Automática e de Mineração de Dados que podem ser directamente aplicados a conjuntos de dados ou invocados em aplicações JAVA desenvolvidas pelo utilizador.

O WEKA incorpora ferramentas para pré-processamento, classificação, *clustering*, associação, selecção e visualização de dados e, por ser um *software* aberto, permite que o utilizador desenvolva os seus próprios algoritmos de aprendizagem [WFT+09].

Ficheiros ARFF

Attribute-Relation File Format (ARFF) é o tipo de formato mais utilizado pelo WEKA.

Este formato consiste num ficheiro de texto *American Standard Code for Information Interchange* (ASCII) que descreve uma lista de instâncias que partilham um conjunto de atributos [WF05]. Está dividido em duas secções: o cabeçalho ou *Header* e a parte de dados ou *Data*.

No cabeçalho é declarado o nome da relação, a lista de atributos e seus respectivos tipos. Os atributos podem assumir quatro tipos: *numeric*, *nominal*, *string* ou *date*.

Os atributos do tipo *numeric*, poderão ser números inteiros ou reais, os do tipo nominal são definidos através de uma lista com os valores que esse atributo pode tomar, os do tipo *string* contêm texto arbitrário e os do tipo *date* definem datas.

A secção de dados surge depois do cabeçalho e contém a declaração de todas as instâncias da relação. Cada instância é representada por uma única linha onde os valores de cada atributo são delimitados por vírgulas e aparecem tendo em conta a ordem com foram declarados no cabeçalho. No caso de existirem atributos sem valor ou *missing values*, estes são representados através do símbolo *?*.

5.3 Algoritmos de Classificação

Na tentativa de encontrar qual o classificador mais capaz de gerar o melhor modelo de classificação para o problema em questão, testaram-se várias hipóteses. Assim optou-se por testar duas regras de classificação, um classificador probabilístico, uma ávore de decisão, uma *Support Vector Machine* e um classificador interactivo. Para que mais tarde se compreenda melhor os resultados obtidos, segue-se um descrição do funcionamento de cada um desses algoritmos.

5.3.1 Zero Rule

Zero Rule, também denominado *ZeroR* ou *0-R*, é o mais primitivo dos classificadores baseados em regras.

O *ZeroR* classifica as instâncias segundo a classe maioritária, isto é, prediz a moda no caso em que a classe é nominal ou a média no caso em que a classe é numérica [WF05]. Embora seja um classificador bastante básico e na prática produza resultados pouco satisfatórios, é comummente utilizado para avaliar o sucesso e a performance de outros classificadores mais complexos.

5.3.2 One Rule

One Rule, também denominado *OneR* ou *1-R*, é um simples classificador baseado em regras.

Este classificador constrói uma regra para cada um dos atributos do conjunto de dados e, em seguida, selecciona a regra que oferece a menor taxa de erro [Hol93]. As regras são determinadas através da classe que aparece com mais frequência para cada valor do atributo e a taxa de erro é o número de instâncias onde a classe de um valor do atributo não corresponde ao valor do atributo na regra. No caso de existirem duas ou mais regras com a mesma taxa de erro, é escolhida aleatoriamente uma delas. O algoritmo do *OneR* é descrito abaixo.

Algoritmo 1 ONE RULE

```
1: for cada atributo A do
2:    for cada valor V desse atributo do
3:       Conta a frequência com que cada classe aparece
4:       Encontra a classe mais frequente, c
5:       Constrói a regra "se A = V então C = c"
6:    end for
7:    Calcula a taxa de erro da regra
8: end for
9: Escolhe o atributo com a menor taxa de erro
```

Este algoritmo é bastante utilizado devido à sua simplicidade e por alcançar bons resultados em problemas reais utilizando apenas um atributo.

5.3.3 Naive Bayes

Naive Bayes é um classificador probabilístico muito popular devido à sua simplicidade e eficiência computacional, o que faz com que seja bastante utilizado em variadas áreas e problemas.

Denomina-se *Naive Bayes* porque aplica o Teorema de Bayes de uma forma ingénua, isto é, assumindo que todos os atributos são igualmente importantes e condicionalmente independentes [WF05].

O Teorema de Bayes relaciona as probabilidades condicionais de dois eventos aleatórios X e Y e é frequentemente utilizado para calcular probabilidades à *posteriori* tendo o conhecimento de um conjunto de evidências.

Teorema de Bayes

Sendo $P(X)$ a probabilidade de X acontecer, $P(Y)$ a probabilidade de Y acontecer e $P(X|Y)$ a probabilidade de X acontecer dado que Y aconteceu, a probabilidade de Y acontecer dado que X aconteceu, $P(Y|X)$, é dada por:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{5.8}$$

Uma vez exposto o Teorema de Bayes, podemos ver de que forma é que o *Naive Bayes* o utiliza na classificação. Suponha-se que:

- $E = \{e_1, e_2, \ldots, e_j\}$ é o conjunto de instâncias e $e \in E$;
- $A = \{a_1, a_2, \ldots, a_n\}$ é o conjunto de atributos de cada instância;
- $C = \{c_1, c_2, \ldots, c_m\}$ é o conjunto de classes mutuamente exclusivas e $c \in C$;

Então:

$$P(c|e_A) = \frac{P(e_A|c)P(c)}{P(e_A)} \tag{5.9}$$

Uma vez que o conjunto de atributos das instâncias é constante e independente das classes, a expressão pode ser simplificada:

$$P(c|e_A) \propto P(e_A|c)P(c) \tag{5.10}$$

Como o algoritmo assume independência entre os atributos, $P(e_A|c)$ é facilmente calculável através da expressão:

$$P(e_A|c) = P(e_{a_1}|c) \times \cdots \times P(e_{a_n}|c) \tag{5.11}$$

Para calcular a classe c mais provável para uma instância $e \in E$, o algoritmo calcula as probabilidades $P(c|e_A)$, $\forall c \in C$. Posteriormente escolhe a probabilidade mais elevada, ou seja, escolhe a hipótese máxima à *posteriori* (MAP) que maximiza a probabilidade $P(c|e_A)$:

$$\begin{aligned} h_{MAP} &= argmax P(c|e_A) \\ &= argmax P(e_A|c)P(c) \end{aligned}$$

5.3.4 J48 Decision Tree

J48 Decision Tree ou apenas J48 é uma implementação da árvore de decisão C4.5 [Qui93]. Este algoritmo constrói um modelo de árvore de decisão que prediz qual a classe para uma nova instância, baseando-se num conjunto de instâncias já classificadas.

A construção da árvore de decisão é feita do topo para a base e em cada nó, é escolhido o atributo que melhor separa os dados, isto é, o atributo que tem o maior ganho de informação.

Para que se perceba o que é o ganho de informação, é necessário definir a entropia.

Seja S um conjunto constituído por instâncias classificadas em c classes, a entropia de S, $H(S)$, é calculada através da expressão:

$$H(S) = \sum_{i=1}^{c} -p_i \log_2 p_i \tag{5.12}$$

onde p_i é a proporção de S pertencente à classe i.

O ganho de informação, $GI(S, A)$ de um atributo A, relativamente ao conjunto de instâncias S é definido por:

$$GI(S, A) = H(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} H(S_v) \tag{5.13}$$

Onde:

- $Valores(A)$ é o conjunto de todos valores possíveis para o atributo A
- S_v é o subconjunto de S para o qual o atributo A tem o valor v (i.e., $S_v = s \in S | A(s) = v$)
- $|Sv|$ é o número de elementos de S_v
- $|S|$ é o número de elementos de S

Após a escolha de um atributo, o conjunto de dados é dividido em subconjuntos, correspondentes aos diferentes valores desse atributo. Este processo repete-se para cada subconjunto até que as instâncias presentes em cada subconjunto pertençam a uma única classe.

Numa árvore de decisão, os nós internos dizem respeito aos diferentes atributos; os ramos, aos possíveis valores que esses atributos podem ter; e os nós terminais (ou folhas), às classes possíveis a que uma instância pode pertencer.

Para classificar uma nova instância basta percorrer um caminho partindo da raiz e chegando até uma das folhas da árvore. Cada percurso desde a raiz até às folhas, corresponde a uma regra de classificação.

As árvores de decisão normalmente aprendem um conjunto de regras bastante precisas, daí que muitas vezes sejam utilizadas para comparar a taxa de precisão de outros algoritmos.

5.3.5 Support Vector Machines

As Máquinas de Vectores de Suporte ou SVM baseiam-se no Princípio de Indução da Minimização do Risco Estrutural que deriva da Teoria da Aprendizagem Estatística [Vap99]. A sua designação deve-se ao facto de utilizarem algumas instâncias como suporte para efectuar a separação entre classes.

As SVM podem ser lineares ou não lineares consoante o conjunto de dados a classificar seja linearmente separável ou não. Um conjunto de dados é linearmente separável quando é possível separar as diferentes classes recorrendo a um hiperplano [Mit97].

Formalmente, podemos dizer que uma SVM linear utiliza um pequeno conjunto de instâncias em limite crítico para formar os vectores de suporte que permitem construir o hiperplano que maximiza a separação entre duas classes [Hay99]. Quanto maior for o espaço que separa as duas classes, melhor será a capacidade de generalização do classificador. A Figura 5.8 representa o conceito.

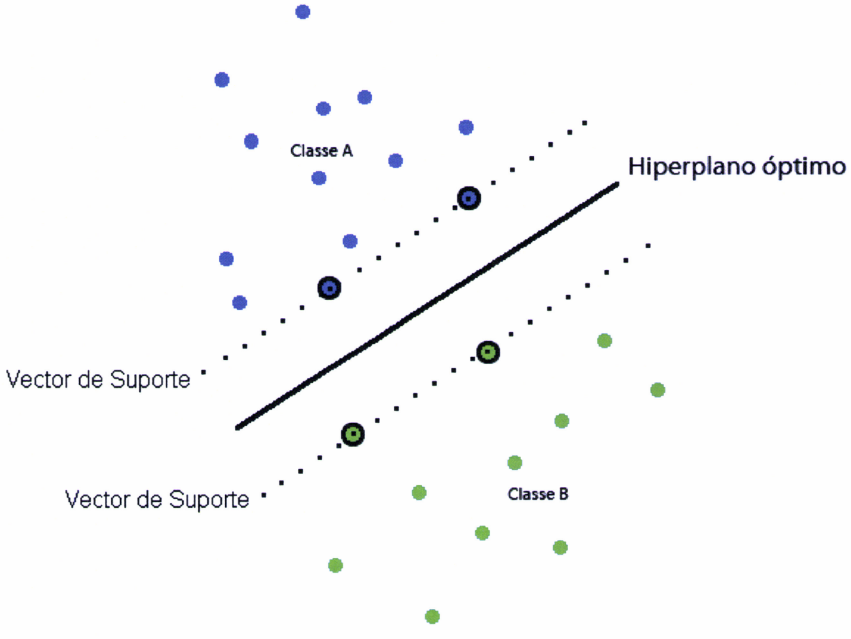


Figura 5.8: Hiperplano de separação de classes construído por uma SVM Linear

Na presença de um conjunto de dados não separável linearmente, as SVMs, de acordo com o Teorema de Cover [HP05] e recorrendo a funções de *kernel*, transformam o conjunto de dados inicial num novo conjunto de dados linearmente separável, chamado espaço de características, onde passa a ser possível encontrar um hiperplano capaz de separar esses dados [Hay99].

Uma função de *kernel* recebe dois pontos do conjunto de dados inicial e calcula o produto escalar desses dados no novo espaço de características [Her01]. As funções de *kernel* mais utilizadas podem ser vistas na seguinte tabela:

NOME	FUNÇÃO
Linear	$K(x_i, x_j) = x_i T_{x_j}$
Polinomial	$K(x_i, x_j) = (\delta(x_i \cdot x_j) + k)^d, d > 0$
Radial	$K(x_i, x_j) = \exp(-\delta\|x_i - x_j\|^2), g > 0$
Sigmoidal	$K(x_i, x_j) = \tanh(kx_i T_{x_j} - d)$

Tabela 5.3: Funções *kernel* mais utilizadas em *Support Vector Machines*

Neste trabalho utilizou-se o algoritmo *Sequential Minimal Optimization* para treino de SVMs com funções de *kernel* polinomiais.

O algoritmo SMO decompõe o problema inicial em vários sub-problemas e resolve-os de forma analítica reduzindo bastante o tempo de processamento [Pla99].

Devido à sua boa capacidade de generalização e robustez em lidar com conjuntos de dados de grande dimensão, as SVM têm aplicação em diversos problemas de regressão, classificação e reconhecimento de padrões em campos como a Bioinformática ou a análise de textos e imagens [Hay99, Bur98].

5.3.6 User Classifier

O *User Classifier*, tem como objectivo tornar o processo de construção do modelo de classificação tão intuitivo quanto possível.

Ao contrário dos classificadores *standard*, que se baseiam num conjunto de dados de entrada e constroem um modelo de classificação, o *User Classifier* permite a classificação dos dados de forma interactiva através de meios visuais [WFH+01].

Neste tipo de classificação, os dados estão representados num gráfico de dispersão em função de dois atributos escolhidos pelo utilizador, como se pode visualizar na Figura 5.9.

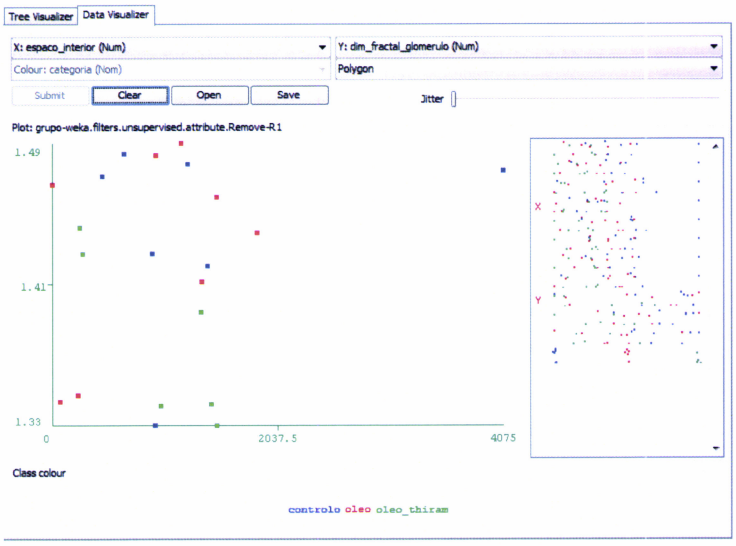


Figura 5.9: Gráfico de dispersão de dados

Para classificar os dados, o utilizador define fronteiras de separação dos dados recorrendo ao desenho de polígonos em torno dos mesmos (Fig. 5.10).

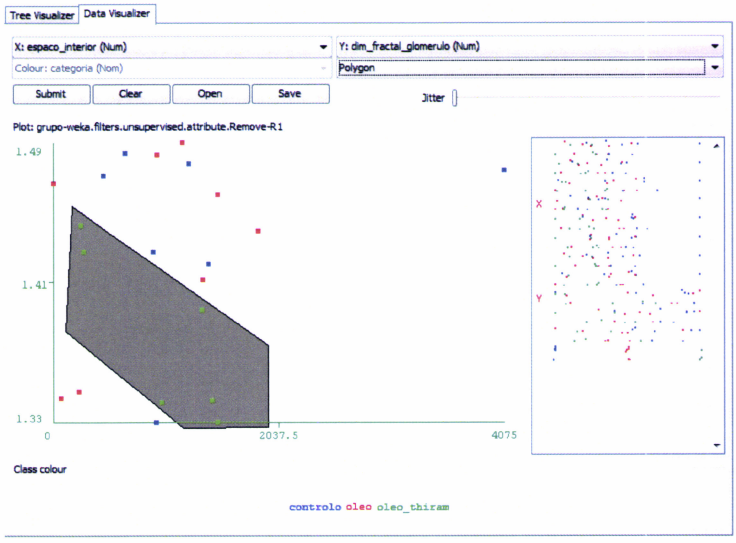


Figura 5.10: Separação de dados recorrendo ao desenho de um polígono

Uma separação é definida pela área de um polígono, ou pela união de áreas no caso de existir mais que um polígono. De cada separação resultam dois nós, um que contém as instâncias presentes no interior do polígono e o outro as restantes instâncias (Fig. 5.11).

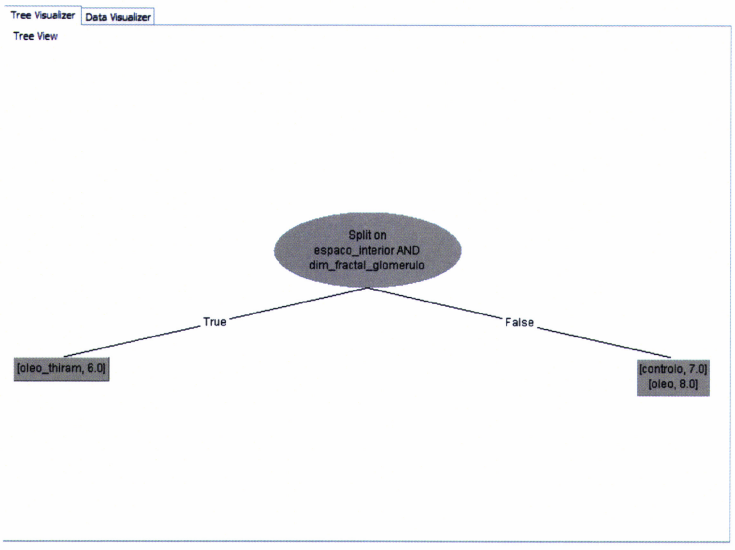


Figura 5.11: Nós resultantes de uma separação de dados

Os nós resultantes das separações são utilizados para a construção de uma árvore de decisão (Fig. 5.12).

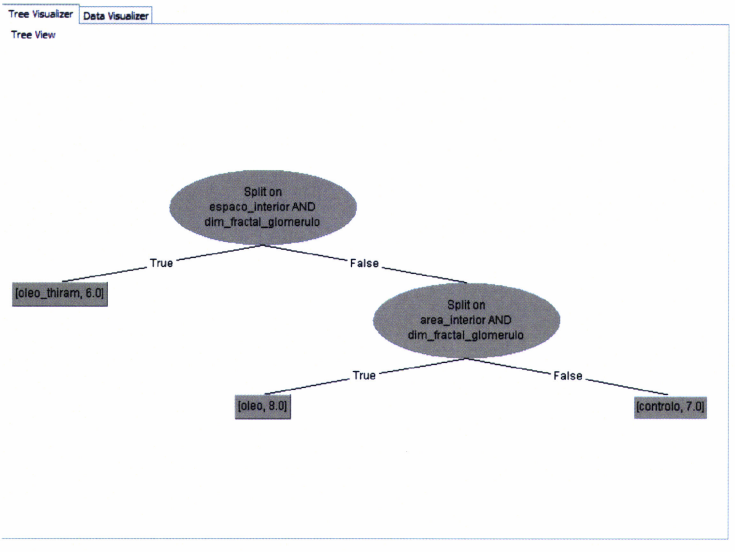


Figura 5.12: Árvore de decisão resultante de duas separações de dados

O processo de separação e consequente criação de nós, é repetido até que o utilizador considere o resultado do classificador satisfatório.

As vantagens deste género de classificadores residem no facto de a classificação

se tornar bastante mais explícita, simples e flexível e possibilitar a utilização por parte de utilizadores com pouco ou nenhum conhecimento acerca do domínio dos dados.

5.4 Pré-Processamento dos Dados

Antes de proceder à utilização do WEKA para construir os modelos de classificação a partir dos diferentes algoritmos, foi necessário proceder a algumas alterações dos dados.

Casas Decimais

Com o intuito de homogenizar os dados, o número de casas decimais foi reduzido às centésimas, ficando todos os valores de todos os atributos numéricos com 3 casas decimais.

Conversão do Formato dos Ficheiros

Como referido no final do capítulo anterior, os dados resultantes da extracção de características após a utilização do *software* ImageJ, foram armazenados em ficheiros *Excel* com o formato *Microsoft Spreadsheet File* (XLS).

O WEKA permite a leitura de inúmeros formatos de ficheiros mas o formato XLS não consta nessa lista. Entre os formatos que permite ler, os mais utilizados são o formato *Comma-Separated Values* (CSV) e o ARFF.

Uma vez que a leitura do formato XLS não é possível, foi necessário efectuar uma conversão do formato dos ficheiros para um que fosse compatível com o programa. Por ser melhor estruturado e permitir guardar mais informação, o formato escolhido foi o ARFF.

Não sendo possível efectuar uma conversão directa do formato XLS para o formato ARFF, a solução encontrada passou por converter os ficheiros do formato XLS para o CSV, de modo a que a leitura pelo WEKA se tornasse possível. Esta operação permitiu a leitura dos ficheiros por parte do WEKA e, após a possibilidade de leitura utilizou-se o próprio WEKA para, finalmente, fazer a conversão para o formato escolhido, o ARFF.

Transformação de Dados

Tentando prever uma redução na incerteza e com o intuito de obter melhores resultados finais, foram efectuadas algumas transformações aos dados. As transformações em questão foram:

- A agregação dos dados referentes ao mesmo animal através do cálculo da média aritmética dos valores dos atributos para cada animal;
- A agregação das classes Controlo e Óleo de Milho, passando somente a existir duas classes, C+O e Thirame;

A agregação de classes é aceitável, pois a classe Óleo de Milho serviu apenas como teste para garantir que se existissem alterações morfológicas no rim, estas eram derivadas à administração do xenobiótico e não devido ao efeito do Óleo de Milho, que por ser um excipiente não deve ter efeito nas células renais.

Assim, foram criados quatro ficheiros com as seguintes características:

Ficheiro	N.° Instâncias	N.° Classes
1	220	3
2	21	3
3	220	2
4	21	2

Tabela 5.4: Ficheiros utilizados nos testes experimentais

5.5 Resultados Experimentais

Nesta secção são apresentados os resultados obtidos com a utilização dos vários classificadores sobre as características extraídas das imagens tratatas nos capítulos anteriores [NRCeS+10].

Dada a escassez de dados, utilizou-se como método de avaliação de desempenho a validação cruzada com 10 *folds* nos Ficheiros 1 e 3 e com 4 *folds* nos Ficheiros 2 e 4. No caso do classificador interactivo, o *User Classifier*, utilizou-se todo o conjunto de dados para treino e posteriormente para teste.

Em primeiro lugar, testaram-se todos os classificadores, à excepção do *User Classifier*, utilizando os quatro ficheiros da Tabela 5.4 com o conjunto normal de atributos. Posteriormente, testaram-se novamente os classificadores utilizando os mesmos ficheiros, mas aplicando uma selecção de atributos sobre o conjunto de atributos inicial. Uma vez que a escolha de atributos é um requisito para a sua utilização, o *User Classifier*, apenas integrou os testes onde existiu selecção de atributos. Neste classificador, os atributos foram seleccionados manualmente recorrendo ao gráfico de dispersão dos dados e considerando os dois atributos que melhor separavam os dados visualmente: a área vazia e a dimensão fractal interior.

Nos restantes classificadores, a selecção de atributos foi feita utilizando o algoritmo *CfsSubsetEval* do WEKA, que avalia o valor de um subconjunto de atributos

considerando a sua capacidade preditiva em conjunto com o grau de redundância entre os mesmos [HS98]. A Tabela 5.32, mostra os atributos considerados na classificação para cada ficheiro:

Ficheiro 1	Ficheiro 2	Ficheiro 3	Ficheiro 4
Área Capsular	Dimensão Fractal Exterior	Área Total	Área Capsular
Perímetro	Dimensão Fractal Interior	Perímetro	Perímetro
Diâmetro	Área Vazia		Área Vazia
Área de Vasos			
Área Vazia			

Tabela 5.5: Atributos considerados pelo algoritmo *CfsSubsetEval*

Em alguns dos classificadores foram afinados determinados parâmetros, de modo a obter melhores resultados. As afinações efectuadas basearam-se na experimentação por tentativa e erro. A parametrização utilizada em cada classificador consta no Anexo A.

Para facilitar a exposição dos resultados, estes estão organizados de acordo com os ficheiros presentes na Tabela 5.4.

5.5.1 Sem Selecção de Atributos

Ficheiro 1

Classificador	T_a	T_e	T_{VP}	T_{FP}	p	$F_{Measure}$	AUC ROC
ZeroR	38,6	61,4	38,6	38,6	14,9	21,5	0,479
OneR	47,3	52,7	47,3	27,6	47,3	47,3	0,598
Naive Bayes	49,5	50,5	49,5	27,4	50,8	49,6	0,668
J48	49,1	50,9	49,1	27,4	50,8	49,2	0,624
SVM	46,4	53,6	46,4	27,4	46,2	45,9	0,609
User Classifier	-	-	-	-	-	-	-

Tabela 5.6: Comparativo de medidas dos vários classificadores no Ficheiro 1 sem selecção de atributos

Através das várias medidas podemos ver que, ainda que com um desempenho abaixo do desejado, os classificadores que obtiveram melhores resultados foram o *Naive Bayes* seguido do J48, com taxas de acerto de 49,5% e 49,1%, respectivamente. Apesar de ambos terem igual precisão, o *Naive Bayes* teve maior taxa de

verdadeiros positivos e a sua superioridade relativamente aos restantes classificadores é comprovada pelo valor da $F_{Measure}$ e da AUC ROC.

As matrizes de confusão resultantes destes dois classificadores podem ser vistas nas Tabelas 5.7 e 5.8.

Classificado →	Controlo	Óleo de Milho	Thirame
Controlo	32	33	7
Óleo de Milho	16	48	21
Thirame	7	27	29

Tabela 5.7: Matriz de Confusão *Naive Bayes* - Ficheiro 1 - Sem selecção de atributos

Classificado →	Controlo	Óleo de Milho	Thirame
Controlo	41	26	5
Óleo de Milho	36	39	210
Thirame	12	23	28

Tabela 5.8: Matriz de Confusão *J48* - Ficheiro 1 - Sem selecção de atributos

No extremo oposto aparece o *ZeroR*, com a maior taxa de falsos positivos e as menores taxa de acerto, precisão e AUC ROC. A sua matriz de confusão pode ser vista na Tabela 5.9.

Classificado →	Controlo	Óleo de Milho	Thirame
Controlo	0	72	0
Óleo de Milho	0	85	0
Thirame	0	63	0

Tabela 5.9: Matriz de Confusão *Zero Rule* - Ficheiro 1 - Sem selecção de atributos

Ficheiro 2

CLASSIFICADOR	T_a	T_e	T_{VP}	T_{FP}	p	$F_{Measure}$	AUC ROC
ZeroR	33,3	66,4	33,3	38,8	21,0	24,6	0,427
OneR	61,9	38,1	61,9	18,8	63,8	62,3	0,716
Naive Bayes	47,6	52,4	47,6	26,5	43,2	43,4	0,663
J48	57,1	42,8	57,1	20,6	57,3	55,8	0,680
SVM	71,4	28,6	71,4	14,0	72,1	71,6	0,739
USER CLASSIFIER	-	-	-	-	-	-	-

Tabela 5.10: Comparativo de medidas dos vários classificadores no Ficheiro 2 sem selecção de atributos

Neste caso, a redução de instâncias veio acentuar o mau desepenho do *ZeroR*. O *Naive Bayes*, embora pouco, também baixou de desempenho.

Os classificadores que se sobressaíram foram SVM e o *OneR*, com taxas de acerto de $71,4\%$ e $61,9\%$. As matrizes de confusão resultantes destes classificadores constam nas Tabelas 5.11 e 5.12.

CLASSIFICADO →	CONTROLO	ÓLEO DE MILHO	THIRAME
CONTROLO	6	0	1
ÓLEO DE MILHO	1	5	2
THIRAME	0	2	4

Tabela 5.11: Matriz de Confusão SVM - Ficheiro 2 - Sem selecção de atributos

CLASSIFICADO →	CONTROLO	ÓLEO DE MILHO	THIRAME
CONTROLO	5	1	1
ÓLEO DE MILHO	1	4	3
THIRAME	0	2	4

Tabela 5.12: Matriz de Confusão *One Rule* - Ficheiro 2 - Sem selecção de atributos

É de referir que, apesar de o classificador *Naive Bayes* ter tido uma pior prestação que o *OneR*, na classe Thirame obteve uma precisão de 100% enquanto que o *OneR* obteve apenas $66,7\%$. A matriz de confusão no *Naive Bayes* está presente na Tabela 5.13.

Classificado →	Controlo	Óleo de Milho	Thirame
Controlo	2	4	1
Óleo de Milho	3	2	3
Thirame	0	0	6

Tabela 5.13: Matriz de Confusão *Naive Bayes* - Ficheiro 2 - Sem selecção de atributos

Ficheiro 3

Classificador	T_a	T_e	T_{VP}	T_{FP}	p	$F_{Measure}$	AUC ROC
ZeroR	71,4	28,6	71,4	71,4	50,9	59,4	0,477
OneR	70,5	29,5	70,5	43,2	70,3	70,4	0,636
Naive Bayes	64,5	35,5	64,5	40,8	68,4	65,8	0,688
J48	70,5	29,5	70,5	44,2	70,0	70,2	0,647
SVM	69,1	30,9	69,1	39,0	70,9	69,8	0,698
User Classifier	-	-	-	-	-	-	-

Tabela 5.14: Comparativo de medidas dos vários classificadores no Ficheiro 3 sem selecção de atributos

A agregação de classes melhorou, em geral, a taxa de acerto de todos os classificadores.

Neste caso, o *ZeroR* teve a maior taxa de acerto, no entanto foi o classificador com o pior desempenho, como se pode comprovar através da sua elevada taxa de falsos positivos e baixa $F_{Measure}$. Esta situação deve-se ao facto de, após a agregação de instâncias, o número de instâncias das duas classes agregadas ser bastante superior ao número de instâncias da classe restante, como se pode visualizar na matriz de confusão presente na Tabela 5.15.

Classificado →	C+O	Thirame
C+O	157	0
Thirame	63	0

Tabela 5.15: Matriz de Confusão *Zero Rule* - Ficheiro 3 - Sem selecção de atributos

Estes resultados obrigam a que, nestas condições, não se possa tirar conclusões olhando apenas para a taxa de acerto, devendo considerar todos os outros valores. Assim, ainda que tendo taxas de acerto mais baixas, os classificadores com

melhor desempenho foram o SVM e o *Naive Bayes*, com as menores taxas médias de falsos positivos e as maiores AUC ROC. As matrizes de confusão destes dois classificadores são exibidas nas Tabelas 5.16 e 5.17.

CLASSIFICADO →	C+O	THIRAME
C+O	117	40
THIRAME	28	35

Tabela 5.16: Matriz de Confusão *SVM* - Ficheiro 3 - Sem selecção de atributos

CLASSIFICADO →	C+O	THIRAME
C+O	107	50
THIRAME	28	35

Tabela 5.17: Matriz de Confusão *Naive Bayes* - Ficheiro 3 - Sem selecção de atributos

Ficheiro 4

CLASSIFICADOR	T_a	T_e	T_{VP}	T_{FP}	p	$F_{Measure}$	AUC ROC
ZeroR	71,4	28,6	71,4	71,4	51,0	59,5	0,389
OneR	66,7	33,3	66,7	43,3	68,4	67,4	0,617
Naive Bayes	85,7	14,3	85,7	0,057	90,5	86,3	0,822
J48	71,4	28,6	71,4	21,4	79,2	72,8	0,750
SVM	76,2	23,8	76,2	29,5	77,6	76,7	0,811
USER CLASSIFIER	-	-	-	-	-	-	-

Tabela 5.18: Comparativo de medidas dos vários classificadores no Ficheiro 4 sem selecção de atributos

A simultânea redução de instâncias e agregação de classes, trouxe melhores resultados de classificação relativamente aos casos anteriores.

O *Naive Bayes* foi o classificador que obteve melhor desempenho, com uma taxa de acerto de $85,7\%$ e uma reduzida taxa média de falsos positivos de $0,057\%$. A sua $F_{Measure}$ foi de $86,3\%$, indicando uma excelente relação entre a precisão e a taxa média de verdadeiros positivos. A AUC ROC confirma o seu poder classificativo. É de notar ainda que, mais uma vez este classificador conseguiu uma precisão de 100% na classe Thirame como se pode comprovar observando a sua

matriz de classificação, presente na Tabela 5.19.

CLASSIFICADO →	C+O	THIRAME
C+O	12	3
THIRAME	0	6

Tabela 5.19: Matriz de Confusão *Naive Bayes* - Ficheiro 4 - Sem selecção de atributos

Os classificadores SVM e J48 tiveram também um desempenho aceitável, ainda que com taxas de acerto de aproximadamente 10% abaixo do *Naive Bayes*, no caso do SVM e 14% no caso J48. Ambos tiveram valores para a $F_{Measure}$ entre os 72% e os 77%, indicando uma boa relação entre a precisão e a taxa média de verdadeiros positivos. Ambos se aproximaram do *Naive Bayes* com valores AUC ROC de $0,811$ e $0,750$, respectivamente. As suas matrizes de confusão encontram-se expostas nas Tabelas 5.20 e 5.21.

CLASSIFICADO →	C+O	THIRAME
C+O	12	3
THIRAME	2	4

Tabela 5.20: Matriz de Confusão *SVM* - Ficheiro 4 - Sem selecção de atributos

CLASSIFICADO →	C+O	THIRAME
C+O	10	5
THIRAME	1	5

Tabela 5.21: Matriz de Confusão *J48* - Ficheiro 4 - Sem selecção de atributos

Mais uma vez, o *ZeroR* obteve os piores resultados, comprovados pela sua elevada taxa média de falsos positivos, baixa $F_{Measure}$ e reduzida AUC ROC.

5.5.2 Com Selecção de Atributos

Ficheiro 1

CLASSIFICADOR	T_a	T_e	T_{VP}	T_{FP}	p	$F_{Measure}$	AUC ROC
ZeroR	38,6	61,4	38,6	38,6	14,9	21,5	0,479
OneR	50,9	49,1	50,9	25,3	50,9	50,8	0,628
Naive Bayes	46,4	53,6	46,4	28,1	47,0	46,3	0,658
J48	48,2	51,8	48,2	26,9	48,6	48,3	0,617
SVM	50,9	49,1	50,9	26,4	52,0	50,5	0,681
USER CLASSIFIER	55,9	44,1	55,9	23,7	56,4	55,9	0,684

Tabela 5.22: Comparativo de medidas dos vários classificadores no Ficheiro 1 com selecção de atributos

Nos testes realizados com selecção de atributos sobre o Ficheiro 1, presentes na Tabela 5.22, o *User Classifier* obteve a melhor prestação, com uma taxa de acerto de 55, 9%, uma precisão de 56, 4% e a mais baixa taxa de falsos positivos, 23, 7%. Em sintonia com estes valores, a sua AUC ROC foi a mais alta, 0, 684.

Comparando estes resultados com os da Tabela 5.6, onde não existiu selecção de atributos, podemos verificar que, embora muito ligeira, houve uma melhoria de desempenho na maioria dos classificadores.

O pior desempenho continuou a pertencer a *ZeroR*, uma vez que os seus valores não sofreram alterações.

Ficheiro 2

CLASSIFICADOR	T_a	T_e	T_{VP}	T_{FP}	p	$F_{Measure}$	AUC ROC
ZeroR	33,3	66,7	33,3	38,8	21,0	24,6	0,427
OneR	52,4	48,6	52,4	25,1	53,3	52,2	0,636
Naive Bayes	61,9	38,1	61,9	18,2	62,5	60,4	0,697
J48	57,1	42,9	57,1	20,6	57,3	55,8	0,609
SVM	71,4	28,6	71,4	12,9	74,9	71,1	0,725
USER CLASSIFIER	100	0	100	0	100	100	1

Tabela 5.23: Comparativo de medidas dos vários classificadores no Ficheiro 2 com selecção de atributos

Analisando os resultados da Tabela 5.23, podemos verificar sem surpresa que o pior desempenho foi do classificador *ZeroR*. À semelhança dos testes realizados sem selecção de atributos, a redução de instâncias também causou uma redução

no desempenho.

O *User Classifier* obteve novamente o melhor desempenho, superando os restantes classificadores com excelentes resultados, uma vez que atingiu uma taxa de acerto de 100% e a sua AUC ROC foi máxima. A sua matriz de confusão consta na Tabela 5.24.

Classificado →	Controlo	Óleo de Milho	Thirame
Controlo	7	0	0
Óleo de Milho	0	8	0
Thirame	0	0	6

Tabela 5.24: Matriz de Confusão *User Classifier* - Ficheiro 2 - Com selecção de atributos

Comparando estes resultados com os da Tabela 5.10, podemos perceber que não existiram grandes alterações no desempenho, à excepção do *OneR* que teve uma redução da taxa de acerto em 9% e do *Naive Bayes* que melhorou significativamente, subindo a taxa de acerto de $47,6\%$ para $61,9\%$. É de notar que o *User Classifier* superou o resultado de todos os outros classificadores efectuando ou não selecção de atributos.

Ficheiro 3

Classificador	T_a	T_e	T_{VP}	T_{FP}	p	$F_{Measure}$	AUC ROC
ZeroR	71,4	28,6	71,4	71,4	50,9	59,4	0,477
OneR	72,7	27,3	72,7	46,1	71,1	71,6	0,633
Naive Bayes	73,6	26,4	73,6	43,8	72,3	72,7	0,710
J48	73,2	26,8	73,2	45,0	71,7	72,2	0,708
SVM	78,2	21,8	78,2	45,8	77,3	75,7	0,749
User Classifier	74,1	25,9	74,1	35,1	74,7	74,4	0,695

Tabela 5.25: Comparativo de medidas dos vários classificadores no Ficheiro 3 com selecção de atributos

A Tabela 5.25 mostra que, comparativamente aos resultados expostos nas Tabelas 5.22 e 5.14, o desempenho dos classificadores melhorou, à excepção do *ZeroR*. Mais uma vez, à semelhança do que tinha ocorrido nos resultados de classificação sem selecção de atributos, a agregação de classes veio melhorar a taxa de acerto dos classificadores.

Neste teste, o SVM obteve os melhores resultados, porém, os classificadores *User Classifier*, *Naive Bayes* e J48 não ficaram muito atrás. É ainda de notar que o *User Classifier* teve melhor precisão na classe Thirame comparando com os restantes onde o número de falsos positivos foi sempre superior ao número de verdadeiros negativos. Podemos visualizar as matrizes de classificação dos 4 classificadores:

CLASSIFICADO →	C+O	THIRAME
C+O	148	9
THIRAME	39	24

Tabela 5.26: Matriz de Confusão *SVM* - Ficheiro 3 - Com selecção de atributos

CLASSIFICADO →	C+O	THIRAME
C+O	126	31
THIRAME	26	37

Tabela 5.27: Matriz de Confusão *User Classifier* - Ficheiro 3 - Com selecção de atributos

CLASSIFICADO →	C+O	THIRAME
C+O	134	23
THIRAME	35	28

Tabela 5.28: Matriz de Confusão *Naive Bayes* - Ficheiro 3 - Com selecção de atributos

CLASSIFICADO →	C+O	THIRAME
C+O	134	23
THIRAME	36	27

Tabela 5.29: Matriz de Confusão *J48* - Ficheiro 3 - Com selecção de atributos

Ficheiro 4

CLASSIFICADOR	T_a	T_e	T_{VP}	T_{FP}	p	$F_{Measure}$	AUC ROC
ZeroR	71,4	28,6	71,4	71,4	50,9	59,4	0,477
OneR	66,7	33,3	66,7	33,3	72,2	68,1	0,667
Naive Bayes	85,7	14,3	85,7	0,057	90,5	86,3	0,811
J48	66,7	33,3	66,7	43,3	68,4	67,4	0,628
SVM	81,0	19,0	81,0	17,6	83,8	81,6	0,811
USER CLASSIFIER	100	0	100	0	100	100	1

Tabela 5.30: Comparativo de medidas dos vários classificadores no Ficheiro 4 com selecção de atributos

Os resultados apresentados na Tabela 5.30 mostram que, à semelhança do ocorrido nos testes sem selecção de atributos, a simultânea redução de instâncias e agregação de classes melhoraram os resultados dos classificadores em geral.

O *User Classifier* foi o classificador que obteve o melhor desempenho com taxa de acerto, taxa média de verdadeiros positivos, precisão de 100% e AUC ROC máxima. Seguiram-se o *Naive Bayes* e o SVM. Em comparação com os resultados da Tabela 5.18, os valores do *Naive Bayes* permaneceram iguais e os do SVM melhoraram ligeiramente.

As matrizes de classificação do *User Classifier*, *Naive Bayes* e SVM podem ser vistas nas seguintes tabelas:

CLASSIFICADO →	C+O	THIRAME
C+O	15	0
THIRAME	0	6

Tabela 5.31: Matriz de Confusão *User Classifier* - Ficheiro 4 - Com selecção de atributos

CLASSIFICADO →	C+O	THIRAME
C+O	12	3
THIRAME	0	6

Tabela 5.32: Matriz de Confusão *Naive Bayes* - Ficheiro 4 - Com selecção de atributos

CLASSIFICADO →	C+O	THIRAME
C+O	12	3
THIRAME	1	5

Tabela 5.33: Matriz de Confusão SVM - Ficheiro 4 - Com selecção de atributos

5.5.3 Conclusões

Tendo em conta os resultados dos testes efectuados, podemos considerar que foram satisfatórios.

As transformações efectuadas aos dados, referidas na secção 5.4 deste capítulo, surtiram efeito, pois foi notório o aumento de desempenho nos casos em que existiu redução de instâncias e agregação de classes.

Os testes mostram que, neste caso, uma elevada taxa de acerto do classificador não significa um bom desempenho e é necessário considerar outros valores como: a precisão, a taxa de verdadeiros positivos, a taxa de falsos positivos e a AUC ROC.

A selecção de atributos deve ser considerada, pois não acentuou o custo computacional e permitiu obter melhorias de desempenho, ainda que ligeiras. Os atributos mais relevantes, segundo o algoritmo de selecção utilizado, foram: o diâmetro, o perímetro, a área capsular, a área de vasos, a área vazia, a área total, a dimensão fractal exterior e a dimensão fractal interior.

Considerando estes factos podemos concluir que, entre todos os classificadores testados, as melhores opções são o *Naive Bayes* e o SVM. O *User Classifier* também obteve bom desempenho, sendo superior ao *Naive Bayes* e ao SVM. No entanto, deve ter-se em conta que neste caso o conjunto de treino foi utilizado também como conjunto de teste, o que não é aconselhado.

Finalmente, é de referir que a incerteza presente nos resultados de alguns classificadores poderia diminuir com a utilização de matrizes de custo associadas à matriz de confusão, de forma a valorizar verdadeiros positivos e verdadeiros negativos e a penalizar falsos positivos e falsos negativos.

Capítulo 6

Conclusões e Trabalho Futuro

Neste capítulo são feitas as considerações finais relativas a todo o trabalho desenvolvido nesta dissertação e apresentadas ideias para trabalhos futuros tendo os resultados alcançados neste trabalho como ponto de partida.

6.1 Conclusões

A dimensão do problema tratado nesta dissertação, a quantidade de métodos existentes e as abordagens possíveis para resolvê-lo, fazem com que se torne difícil encontrar uma solução definitiva.

Os objectivos traçados inicialmente foram cumpridos e os resultados obtidos mostraram-se positivos, no entanto, pode-se considerar que o trabalho realizado conduziu apenas a uma solução parcial para o problema. A complexidade inerente ao tipo de imagens utilizadas neste trabalho esteve na origem das dificuldades que surgiram no processo de segmentação das imagens, fazendo com que, em vez de ser efectuado de forma completamentamente automática, o processo acabasse por ser efectuado de forma semi-automática, isto é, assistido pelo utilizador.

A limitação no número de imagens e consequentemente no tamanho do conjunto de dados e o ligeiro desequilíbrio na distribuição das instâncias pelas classes existentes impuseram limitações nos resultados obtidos. A existência de um conjunto de dados maior e mais equilibrado, teria permitido a criação de conjuntos de dados de treino e teste que, provavelmente, dariam origem a um modelo de classificação mais preciso do que os que foram criados.

Para terminar, podemos comprovar que a segmentação de imagem e aplicação de métodos de classificação a dados histomorfométricos se revelou útil na abordagem deste problema, uma vez que os testes efectuados permitiram determinar se um animal esteve sujeito à ingestão de xenobióticos, com um grau de certeza de $85,7\%$ no caso do *Naive Bayes* e 81% no caso da SVM.

6.2 Trabalhos Futuros

Existem nesta dissertação aspectos que poderão ser melhorados/explorados e servir de base para trabalhos futuros.

No processo de segmentação poder-se-á:

- desenvolver um algoritmo que ligue as selecções parciais através dos seus pontos mais próximos, originando selecções completas sem que o utilizador tenha intervir no processo;
- combinar o processo utilizado neste trabalho com outros métodos existentes, por exemplo, modelos de contorno activo (*snakes*) ou crescimento de regiões;

Relativamente às imagens e ao conjunto de dados poder-se-á:

- proceder à obtenção de mais imagens de forma a aumentar o conjunto de dados;
- considerar outros atributos morfológicos, como por exemplo, a circularidade ou número de espaços interiores;

Ao nível do processo de classificação:

- aprofundar a pesquisa realizada, utilizando novos métodos de aprendizagem, como por exemplo, Redes Neuronais;
- permitir que o classificador se auto-melhore com a entrada de novos exemplos;

Globalmente:

- desenvolver uma aplicação capaz de automatizar todo o processo histomorfométrico, isto é, receber uma imagem de um corpúsculo renal, segmentá-la correctamente, medir os atributos e finalmente fazer a classificação;

Este trabalho poderá ser aplicado com utilidade em:

- controlo ambiental;
- apoio ao diagnóstico de ingestão de xenobióticos;

Bibliografia

[AdAdS+96] A.de.A. Araujo, M.C. de Andrade, A.M.M. dos Santos, F.S. Lameiras, and E.A. Bambirra. Digital characterization of the renal glomeruli by the saltykov method. In *Cybernetic Vision, 1996. Proceedings., Second Workshop on*, pages 45 –50, Dec. 1996.

[AR05] Tinku Acharya and Ajoy K. Ray. *Image Processing - Principles and Applications.* Wiley-Interscience, 2005.

[AT93] C. Alippi and V. Torri. Real-time detection of ships in radar images. In *Neural Networks, 1993. IJCNN '93-Nagoya. Proceedings of 1993 International Joint Conference on*, volume 2, pages 1235 – 1238 Vol. 2, Oct. 1993.

[AYAN89] A.E. Adams, H.C. Yung, C.R. Allen, and M.L. Ng. The processing of laryngoscopic images as a diagnostic aid. In *Image Processing and its Applications, 1989., Third International Conference on*, pages 314 –318, July 1989.

[BB01] Pierre Baldi and Soren Brunak. *Bioinformatics: The Machine Learning Approach.* Cambridge: MIT Press, 2nd edition, 2001.

[Bru88] G. Brugal. Pattern recognition, image processing, related data analysis and expert systems integrated in medical microscopy. In *Pattern Recognition, 1988., 9th International Conference on*, pages 286 –293 Vol. 1, Nov. 1988.

[Bur98] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[CGBL+05] J. Carballido-Gamio, J.S. Bauer, Keh-Yang Lee, S. Krause, and S. Majumdar. Combined image processing techniques for characterization of mri cartilage of the knee. In *Engineering in Medi-*

cine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the, pages 3043 –3046, Jan. 2005.

[CLT+06] Y. Cao, D. Liu, W. Tavanapong, J. Wong, J. Oh, and P.C. de Groen. Automatic classification of images with appendiceal orifice in colonoscopy videos. In *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pages 2349 –2352, Aug. 2006.

[CTCW07] B. Canada, G. Thomas, K. Cheng, and J.Z. Wang. Automated segmentation and classification of zebrafish histology images for high-throughput phenotyping. In *Life Science Systems and Applications Workshop, 2007. LISA 2007. IEEE/NIH*, pages 245 –248, Nov. 2007.

[DHS01] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley, New York, NY, USA, 2nd edition, 2001.

[DOY98] Z. Dokur, T. Olmez, and E. Yazgan. Classification of mr and ct images using genetic algorithms. In *Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual International Conference of the IEEE*, volume 3, pages 1418 –1421 Vol. 3, Oct. 1998.

[ETPZ09] Pablo A. Estevez, Michel Tesmer, Claudio A. Perez, and Jacek M. Zurada. Normalized mutual information feature selection. *Trans. Neur. Netw.*, 20(2):189–201, 2009.

[Fin07] Andy Finnell. How to implement a magic wand tool. `http://losingfight.com/blog/2007/08/28/`, 2007. Online; accesso em 19-Outubro-2010.

[FOT+05] J.C. Felipe, J.B. Olioti, A.J.M. Traina, M.X. Ribeiro, E.P.M. Sousa, and Jr. Traina, C. A low-cost approach for effective shape-based retrieval and classification of medical images. In *Multimedia, Seventh IEEE International Symposium on*, page 6 pp., Dec. 2005.

[FSMC06] J.-J. Fernandez, C.O.S. Sorzano, R. Marabini, and J.-M. Carazo. Image processing and 3-d reconstruction in electron microscopy. *Signal Processing Magazine, IEEE*, 23(3):84 –94, May 2006.

[GJ01] Cynthia Gibas and Per Jambeck. *Developing Bioinformatics Computer Skills.* O'Reilly Media, Inc., 2001.

[GW01] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[Hay99] Simon Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice Hall, Upper Saddle River, NJ, 1999. 2nd Edition.

[HBP09] A. Hafiane, F. Bunyak, and K. Palaniappan. Evaluation of level set-based histology image segmentation using geometric region criteria. In *Biomedical Imaging: From Nano to Macro, 2009. ISBI '09. IEEE International Symposium on*, pages 1 –4, June 2009.

[Her01] Ralf Herbrich. *Learning Kernel Classifiers: Theory and Algorithms.* MIT Press, Cambridge, MA, USA, 2001.

[HHH94] R.F. Hanke, U. Hassler, and K. Heil. Fast automatic x-ray image processing by means of a new multistage filter for background modelling. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 1, pages 392 –396 Vol. 1, Nov. 1994.

[HK05] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[HKC07] Jia-Yann Huang, Pan-Fu Kao, and Yung-Sheng Chen. A set of image processing algorithms for computer-aided diagnosis in nuclear medicine whole body bone scan images. *Nuclear Science, IEEE Transactions on*, 54(3):514 –522, June 2007.

[Hol93] Robert C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11(1):63–90, 1993.

[HP05] Zhen Hou and Johne M. Parker. Texture defect detection using support vector machines with adaptive gabor wavelet features. In *WACV-MOTION '05: Proceedings of the Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05) - Vol. 1*, pages 275–280, Washington, DC, USA, 2005. IEEE Computer Society.

[HS98] Mark A. Hall and Lloyd A. Smith. *Practical Feature Subset Selection for Machine Learning.* Hamilton, New Zealand, 1998.

[KIC04] S.M. Khan, R. Islam, and M.U. Chowdhury. Medical image classfication using an efficient data mining technique. In *Machine Learning and Applications, 2004. Proceedings. 2004 International Conference on*, pages 397 – 402, Dec. 2004.

[KM07] R. Komura and K.-i. Muramoto. Classification of forest stand considering shapes and sizes of tree crown calculated from high spatial resolution satellite image. In *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, pages 4356 –4359, July 2007.

[KNAT07] H. Kudo, M. Nomura, T. Asada, and T. Takeda. Image processing method for analyzing cerebral blood-flow using spect and mri. In *Nuclear Science Symposium Conference Record, 2007. NSS '07. IEEE*, volume 5, pages 4015 –4021, Oct. 2007.

[KRB08] Ilya Kamenetsky, Rangaraj M. Rangayyan, and Hallgrimur Benediktsson. Segmentation and analysis of the glomerular basement membrane using the split and merge method. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 3064 –3067, Aug. 2008.

[Lat06] Koldo Latxiondo. Multicell outliner. `http://rsbweb.nih.gov/ij/plugins/multi-cell-outliner.html`, 2006. Online; accesso em 16-Setembro-2010.

[lBQ08] Xing li Bai and Xu Qian. Medical image classification based on fuzzy support vector machines. In *Intelligent Computation Technology and Automation (ICICTA), 2008 International Conference on*, volume 2, pages 145 –149, Oct. 2008.

[LBSM09] O. Liberda, K. Bartusek, Z. Smekal, and J. Mikulka. Data processing in studying the temporomandibular joint, using mr imaging and sonographic techniques. In *Digital Signal Processing, 2009 16th International Conference on*, pages 1 –6, July 2009.

[LELL03] Xiang Li, D. Eremina, Lihong Li, and Zhengrong Liang. Partial volume segmentation of medical images. In *Nuclear Science Symposium Conference Record, 2003 IEEE*, volume 5, pages 3176 – 3180 Vol. 5, Oct. 2003.

[LF08] F.R. Leta and F.F. Feliciano. Computational system to detect defects in mounted and bare pcb based on connectivity and image

correlation. In *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on*, pages 331 –334, June 2008.

[Man83] Benoit B. Mandelbrot. *The Fractal Geometry of Nature.* W. H. Freedman and Co., New York, 1983.

[Man98] Dragos-Anton Manolescu. Feature extraction - a pattern for information retrieval. 1998.

[Mee99] K.J. Meech. Astronomical image processing - applications to ultra-faint imaging of small, moving, solar system bodies: comets and near-earth-objects. In *Intelligent Processing and Manufacturing of Materials, 1999. IPMM '99. Proceedings of the Second International Conference on*, volume 1, page 445 Vol. 1, July 1999.

[MFM+05] A. Madabhushi, M.D. Feldman, D.N. Metaxas, J. Tomaszeweski, and D. Chute. Automated detection of prostatic adenocarcinoma from high-resolution ex vivo mri. *Medical Imaging, IEEE Transactions on*, 24(12):1611 –1625, Dec. 2005.

[Mit97] Tom M. Mitchell. *Machine Learning.* McGraw-Hill, New York, 1997.

[MM90] David S. Mantus and George H. Morrison. Chemical imaging using ion microscopy and digital image processing. *Journal of Vacuum Science Technology A: Vacuum, Surfaces, and Films*, 8(3):2209 –2212, May 1990.

[MNAOSA03] N. Mir-Nasiri, H.H.L. Al-Obaidy, M.J.E. Salami, and S. Amin. An effective vision technique for microchip lead inspection. In *Industrial Technology, 2003 IEEE International Conference on*, volume 1, pages 135 – 139 Vol. 1, Dec. 2003.

[MY08] T. Maruyama and H. Yamamoto. Imaging techniques for skull radiography using ct images. In *Imaging Systems and Techniques, 2008. IST 2008. IEEE International Workshop on*, pages 277 –282, Sep. 2008.

[MYHT92] T. Morita, S. Yamaoka, M. Hayashida, and M. Toyama. Image processing vehicle detector for urban traffic control systems. In *Vehicle Navigation and Information Systems, 1992. VNIS., The 3rd International Conference on*, pages 98 –103, Sep. 1992.

[NJA10] Kien Nguyen, Anil K. Jain, and Ronald L. Allen. Automated gland segmentation and classification for gleason grading of prostate tissue images. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1497 –1500, Aug. 2010.

[NRCeS+10] Bruno Nunes, Luis Rato, F. Capela e Silva, A. Rafael, and A. S. Cabrita. Processing and classification of biological images: Application to histology. In *6th International Conference on Technology and Medical Sciences*, 2010.

[OE02] A. Olukunle and S. Ehikioya. A fast algorithm for mining association rules in medical image data. In *Electrical and Computer Engineering, 2002. IEEE CCECE 2002. Canadian Conference on*, volume 2, pages 1181 – 1187 Vol. 2, 2002.

[OML+08] S. Oprea, C. Marinescu, I. Lita, M. Jurianu, D.A. Visan, and I.B. Cioc. Image processing techniques used for dental x-ray image analysis. In *Electronics Technology, 2008. ISSE '08. 31st International Spring Seminar on*, pages 125 –129, May 2008.

[PKC+04] S. Petushi, C. Katsinis, C. Coward, F. Garcia, and A. Tozeren. Automated identification of microstructures on histology slides. In *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*, pages 424 – 427 Vol. 1, Apr. 2004.

[PKF+00] A. Prochazka, M. Kolinova, J. Fiala, P. Hampl, and K. Hlavaty. Satellite image processing and air pollution detection. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, volume 6, pages 2282 –2285 Vol. 4, 2000.

[Pla99] John C. Platt. Fast training of support vector machines using sequential minimal optimization. pages 185–208, 1999.

[PPVRSPGP07] B. Paniagua Paniagua, M.A. Vega Rodriguez, J.M. Sanchez Perez, and J.A. Gomez Pulido. Image processing and neuro-fuzzy computing for cork quality classification. In *Industrial Informatics, 2007 5th IEEE International Conference on*, volume 2, pages 657 –661, June 2007.

[Pra07] William K. Pratt. *Digital Image Processing: PIKS Scientific Inside.* Wiley-Interscience, 2007.

[QSWR03] X. Qi, Jr. Sivak, M.V., D.L. Wilson, and A.M. Rollins. Processing of endoscopic optical coherence tomography images for quantitative diagnosis of dysplasia. In *Lasers and Electro-Optics, 2003. CLEO '03. Conference on*, page 3 pp., June 2003.

[Qui93] J. Ross Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA, 1993.

[Ras09] Wayne S. Rasband. Imagej, 1997-2009.

[RNR+10] Rahmadwati, G. Naghdy, M. Ross, C. Todd, and E. Norachmawati. Classification cervical cancer using histology images. In *Computer Engineering and Applications (ICCEA), 2010 Second International Conference on*, volume 1, pages 515 –519, Mar. 2010.

[SCSG09] O. Sertel, U.V. Catalyurek, H. Shimada, and M.N. Guican. Computer-aided prognosis of neuroblastoma: Detection of mitosis and karyorrhexis cells in digitized histological images. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 1433 –1436, Dep. 2009.

[SM97] J. Setubal and J. Meidanis. *Introduction to computational molecular biology.* PWS Publishing Company, 1997.

[SS01] George Stockman and Linda G. Shapiro. *Computer Vision.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

[SSS96] A.H. Schistad Solberg and R. Solberg. A large-scale evaluation of features for automatic detection of oil spills in ers sar images. In *Geoscience and Remote Sensing Symposium, 1996. IGARSS '96. 'Remote Sensing for a Sustainable Future.', International*, volume 3, pages 1484 –1486 Vol. 3, May 1996.

[SSS08] G.K. Santhalia, S. Singh, and S.K. Singh. Safer navigation of ships by image processing. In *Modeling Simulation, 2008. AICMS 08. Second Asia International Conference on*, pages 660 –665, May 2008.

[SU10] Beril Sirmacek and Cem Unsalan. Road network extraction using edge detection and spatial voting. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3113 –3116, Aug. 2010.

[SWC01] Jenn-Lung Su, Guo-Zhen Wu, and I-Pin Chao. The approach of data mining methods for medical database. In *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, volume 4, pages 3824 – 3826 Vol. 4, 2001.

[THS96] M. Toyama, T. Horiuchi, and Y. Shiina. Development of a portable traffic flow measurement system using image processing. In *Vehicle Navigation and Information Systems Conference, 1996. VNIS '96*, volume 7, pages 62 – 68, Oct. 1996.

[TJO01] T. Tanaka, T. Joke, and T. Oka. Cell nucleus segmentation of skin tumor using image processing. In *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, volume 3, pages 2716 – 2719 Vol. 3, 2001.

[Tru81] H.J. Trussell. Processing of x-ray images. *Proceedings of the IEEE*, 69(5):615 – 627, May 1981.

[TSK05] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (1st Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[ULdC05] D.M. Ushizima, A.C. Lorena, and A.C.P.L.F. de Carvalho. Support vector machines applied to white blood cell recognition. In *Hybrid Intelligent Systems, 2005. HIS '05. Fifth International Conference on*, page 6 pp., Nov. 2005.

[UMS89] S.E. Umbaugh, R.H. Moss, and W.V. Stoecker. Automatic color segmentation of images with application to detection of variegated coloring in skin tumors. *Engineering in Medicine and Biology Magazine, IEEE*, 8(4):43 –50, Dec. 1989.

[Vap99] Vladimir Vapnik. An overview of statistical learning theory. 10(5):988–999, 1999.

[WCS94] C.X. Wang, P. Chen, and W.E. Snyder. Left ventricle quantification in gated cardiac nuclear medicine images using bayes classification. In *Engineering in Medicine and Biology Society, 1994. Engineering Advances: New Opportunities for Biomedical Engineers. Proceedings of the 16th Annual International Conference of the IEEE*, pages 520 –521 Vol. 1, Nov. 1994.

[WDL08] Xiang Wu, Chunni Dai, and Jingao Liu. A novel approach for face recognition based on stereo image processing algorithm. In *Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on*, pages 1245 –1249, July 2008.

[WF05] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, CA, 2nd edition, 2005.

[WFH+01] Malcolm Ware, Eibe Frank, Geoffrey Holmes, Mark Hall, and Ian H. Witten. Interactive machine learning: letting users build classifiers. *Int. J. Hum.-Comput. Stud.*, 55(3):281–292, 2001.

[WFT+09] Ian H. Witten, Eibe Frank, Len Trigg, Mark Hall, Geoffrey Holmes, and Bernhard Pfahringer. The weka data mining software: An update, 2009.

[WGZ+99] Fengyuan Wang, Zonghe Guo, Daolin Zhang, Xiaoyan Fu, Xibin Zheng, and Zeyan Liu. Detection of road guiding lines with computer image processing. In *Vehicle Electronics Conference, 1999. (IVEC '99) Proceedings of the IEEE International*, pages 275 –277 Vol. 1, 1999.

[Wik] Wikipedia. Flood fill. `http://en.wikipedia.org/wiki/Flood_fill`. Online; accesso em 19-Outubro-2010.

[WMZDD89] A. Wail Mussa, M.H. Zeini, G.N. Dutton, and T.S. Durrani. Image processing of patient's test charts in diagnostic ophthalmology. In *Image Processing and its Applications, 1989., Third International Conference on*, pages 360 –366, July 1989.

[XTT05] Yongguan Xiao, Tiow-Seng Tan, and Seng-Chuan Tay. Utilizing edge to extract roads in high-resolution satellite imagery. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 1, pages I – 637–40, Sep. 2005.

[YMK88] H. Yamada, C. Merritt, and T. Kasvand. Recognition of kidney glomerulus by dynamic programming matching method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(5):731 –737, Sep. 1988.

[ZF06a] Jun Zhang and Jiulun Fan. Glomerulus extraction based on genetic algorithm and watershed transform. In *Intelligent Robots*

and Systems, 2006 IEEE/RSJ International Conference on, pages 4863 –4866, Oct. 2006.

[ZF06b] Jun Zhang and Jiulun Fan. Medical image segmentation based on wavelet transformation and watershed algorithm. In *Information Acquisition, 2006 IEEE International Conference on*, pages 484 –488, Aug. 2006.

[ZH08a] Jun Zhang and Jinglu Hu. Glomerulus extraction by optimizing the fitting curve. In *Computational Intelligence and Design, 2008. ISCID '08. International Symposium on*, volume 2, pages 169 –172, Oct. 2008.

[ZH08b] Jun Zhang and Jinglu Hu. Image segmentation based on 2d otsu method with histogram analysis. In *Computer Science and Software Engineering, 2008 International Conference on*, volume 6, pages 105 –108, Dec. 2008.

[ZN01] Tao Zhao and Ram Nevatia. Car detection in low resolution aerial image. *Computer Vision, IEEE International Conference on*, 1:710, 2001.

[ZSW+08] Hong Zhu, Weizhen Sun, Minhua Wu, Guixia Guan, and Yong Guan. Pre-processing of x-ray medical image based on improved temporal recursive self-adaptive filter. In *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for*, pages 758 –763, Nov. 2008.

[ZZ07] Jun Zhang and Qieshi Zhang. Color image segmentation based on wavelet transformation and sofm neural network. In *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, pages 1778 –1781, Dec. 2007.

Anexos

Anexo A

Parametrização de Classificadores

Neste anexo são apresentados os parâmetros utilizados pelos diferentes classificadores nos testes realizados na Secção 5.5.

Os classificadores *ZeroR* e *User Classifier* não possuem parâmetros de afinação, como tal, não constam neste anexo.

A.1 One Rule

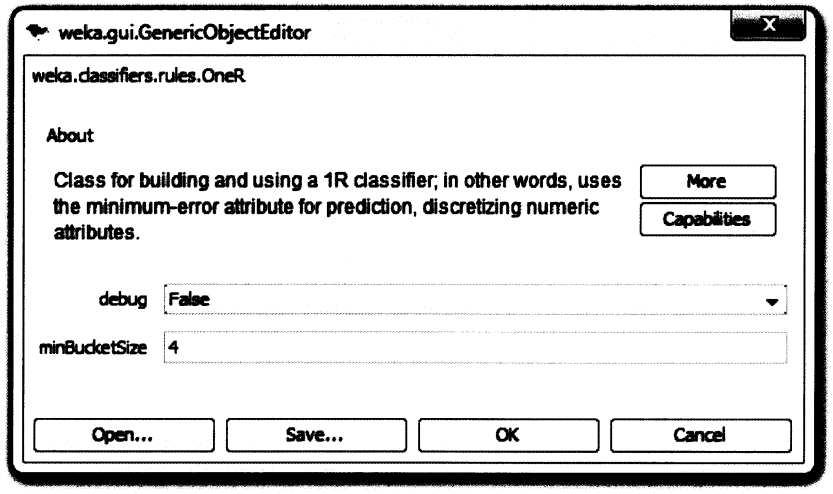


Figura A.1: Parâmetros do *One Rule* na classificação do Ficheiro 1 sem selecção de atributos

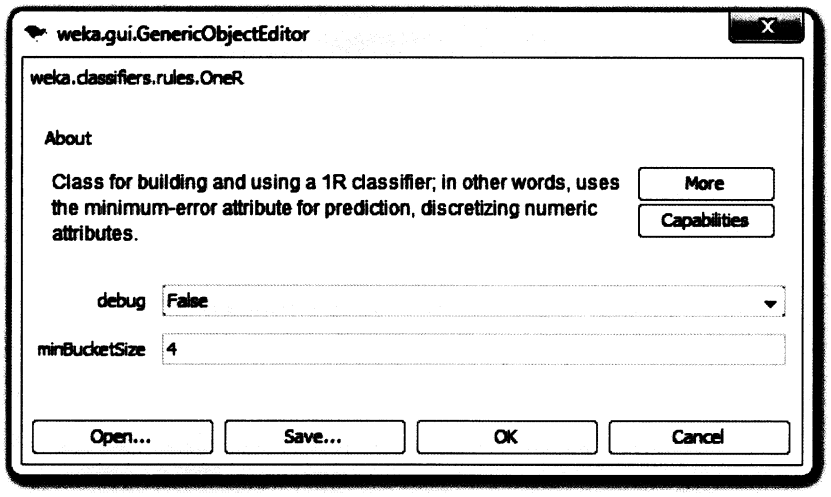


Figura A.2: Parâmetros do *One Rule* na classificação do Ficheiro 2 sem selecção de atributos

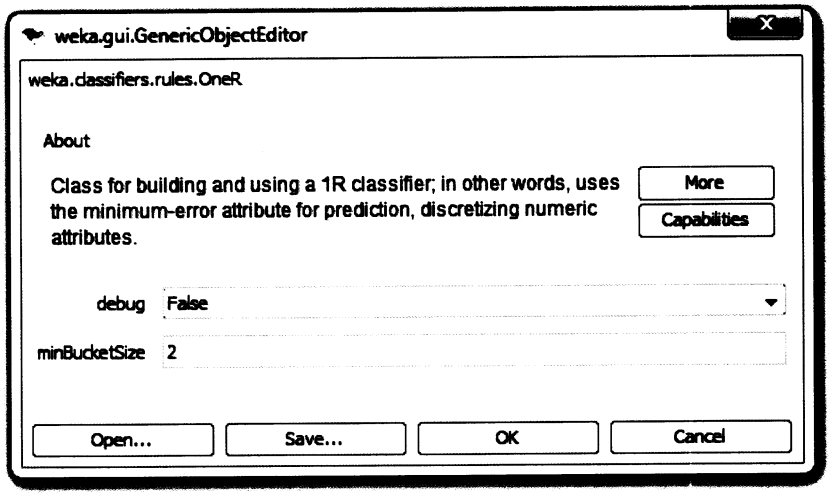


Figura A.3: Parâmetros do *One Rule* na classificação do Ficheiro 3 sem selecção de atributos

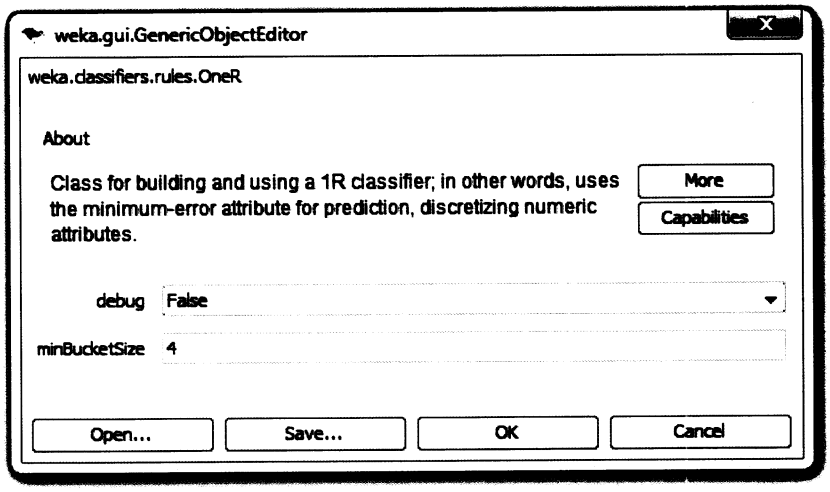


Figura A.4: Parâmetros do *One Rule* na classificação do Ficheiro 4 sem selecção de atributos

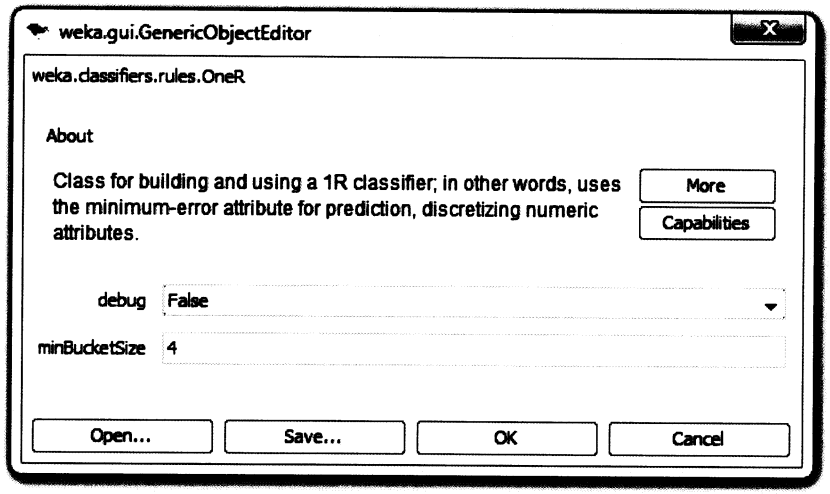


Figura A.5: Parâmetros do *One Rule* na classificação do Ficheiro 1 com selecção de atributos

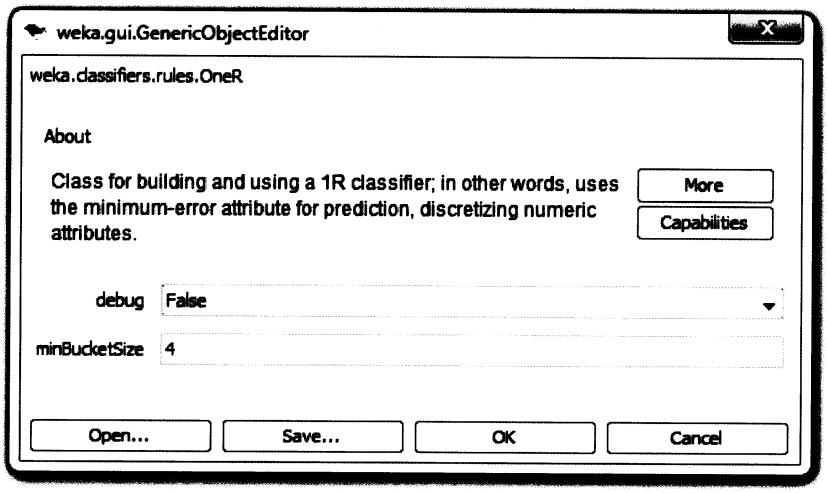


Figura A.6: Parâmetros do *One Rule* na classificação do Ficheiro 2 com selecção de atributos

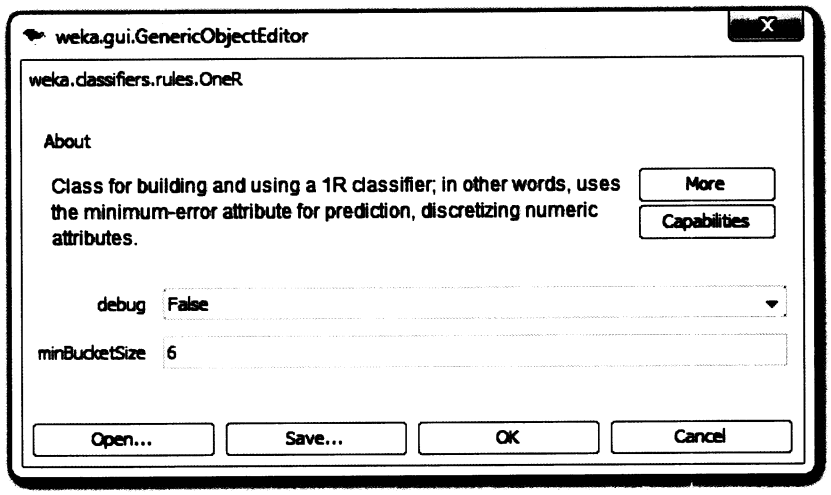


Figura A.7: Parâmetros do *One Rule* na classificação do Ficheiro 3 com selecção de atributos

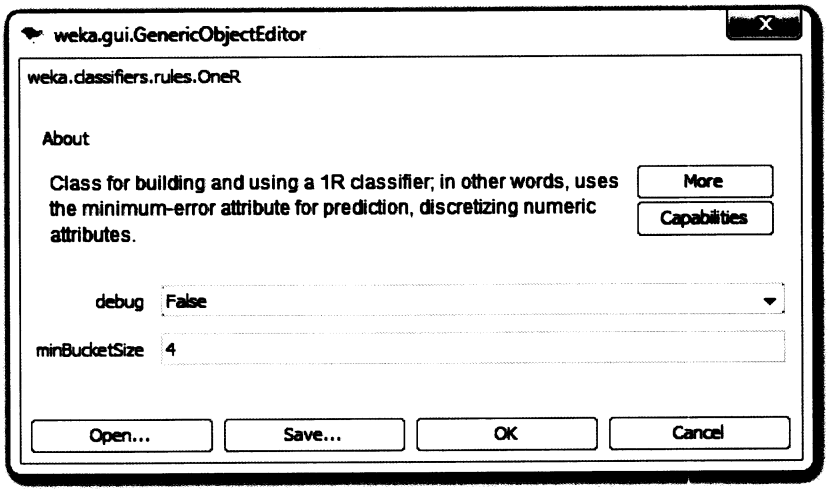


Figura A.8: Parâmetros do *One Rule* na classificação do Ficheiro 4 com selecção de atributos

A.2 Naive Bayes

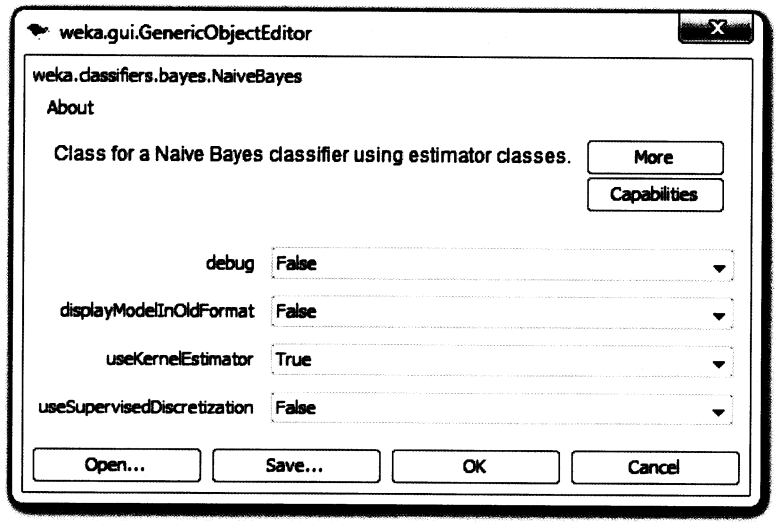


Figura A.9: Parâmetros do *Naive Bayes* na classificação do Ficheiro 1 sem selecção de atributos

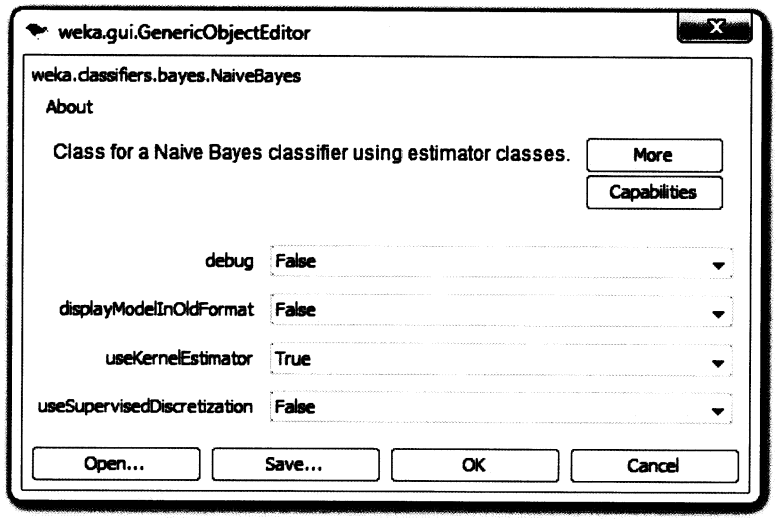


Figura A.10: Parâmetros do *Naive Bayes* na classificação do Ficheiro 2 sem selecção de atributos

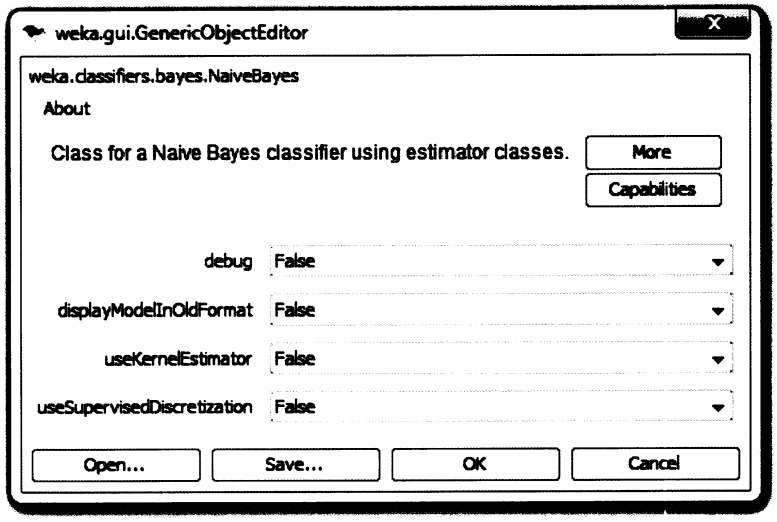


Figura A.11: Parâmetros do *Naive Bayes* na classificação do Ficheiro 3 sem selecção de atributos

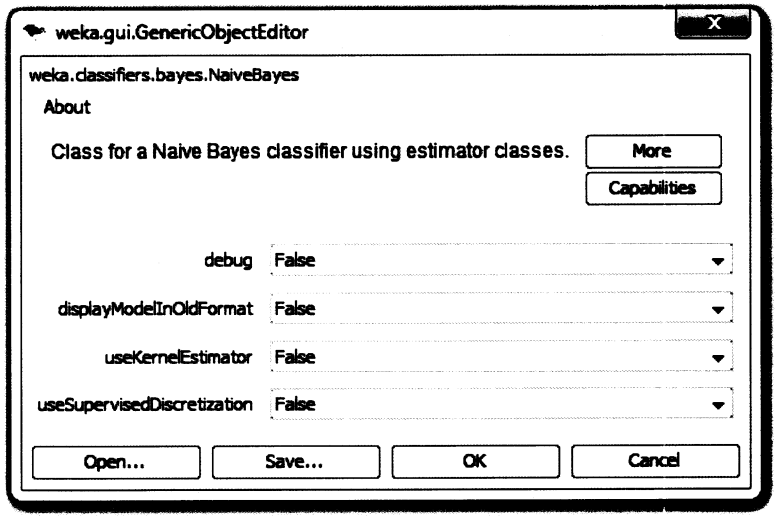


Figura A.12: Parâmetros do *Naive Bayes* na classificação do Ficheiro 4 sem selecção de atributos

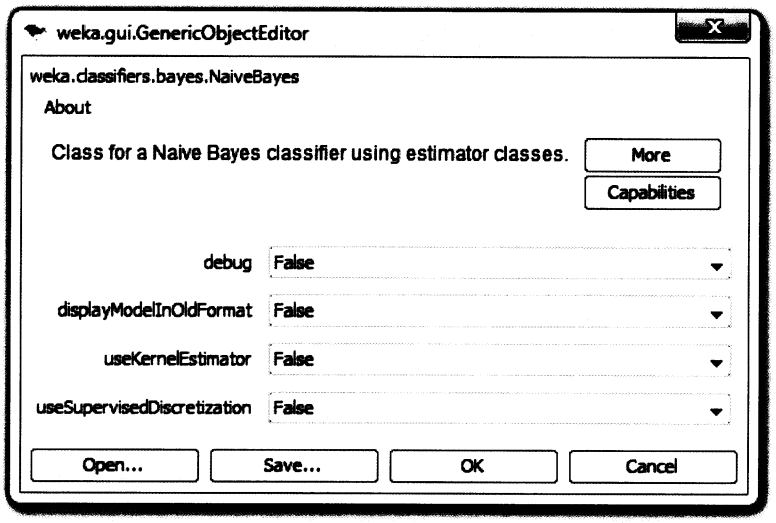


Figura A.13: Parâmetros do *Naive Bayes* na classificação do Ficheiro 1 com selecção de atributos

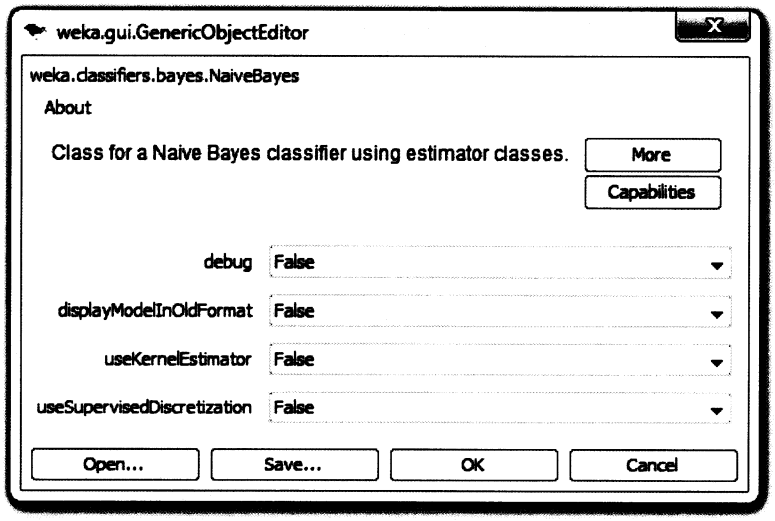


Figura A.14: Parâmetros do *Naive Bayes* na classificação do Ficheiro 2 com selecção de atributos

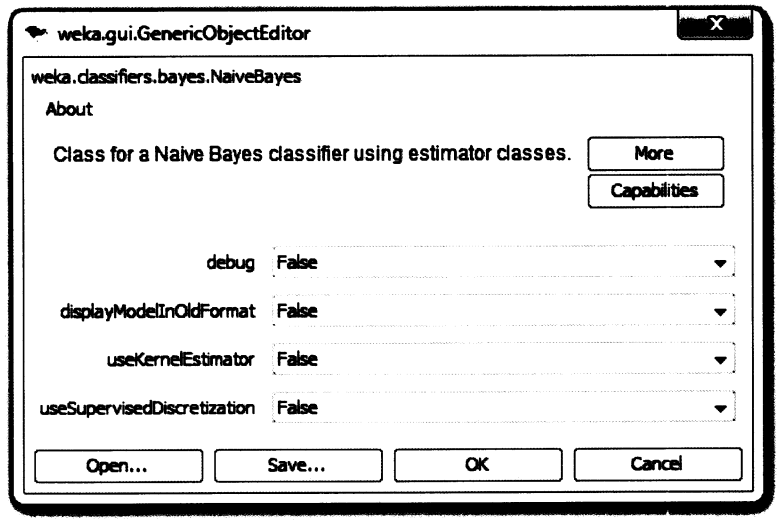


Figura A.15: Parâmetros do *Naive Bayes* na classificação do Ficheiro 3 com selecção de atributos

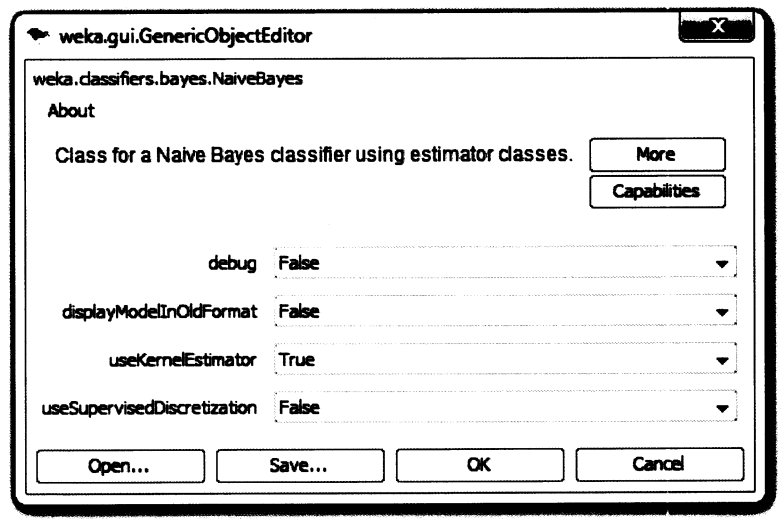


Figura A.16: Parâmetros do *Naive Bayes* na classificação do Ficheiro 4 com selecção de atributos

A.3 J48 Decision Tree

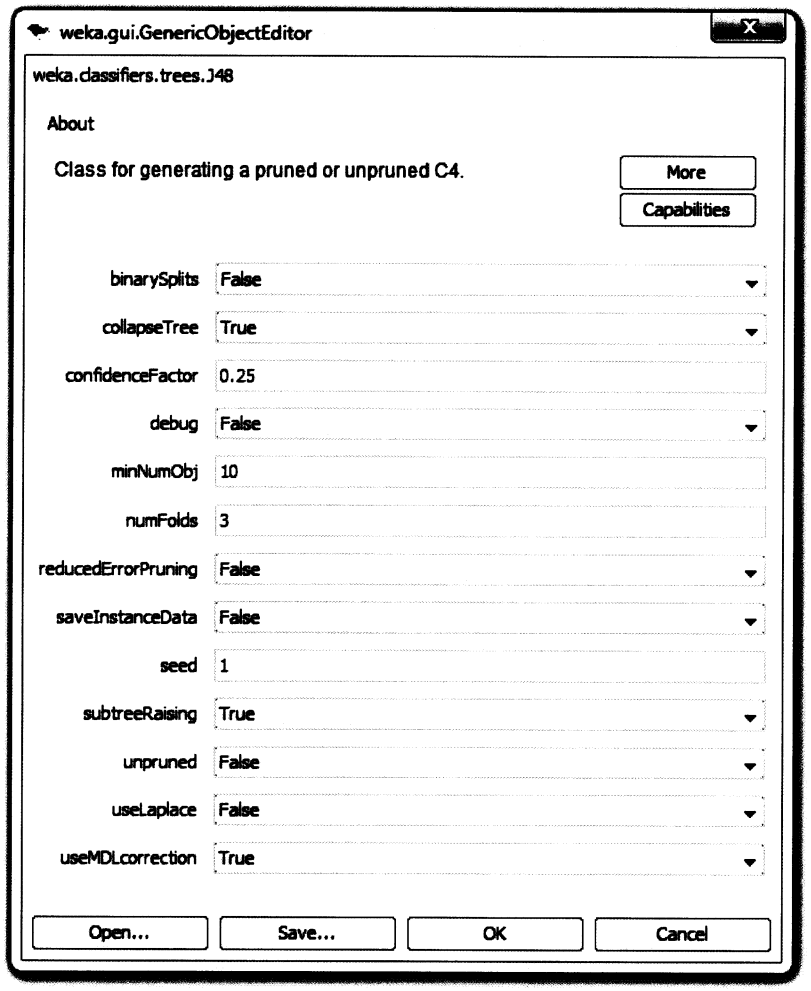


Figura A.17: Parâmetros do *J48 Decision Tree* na classificação do Ficheiro 1 sem selecção de atributos

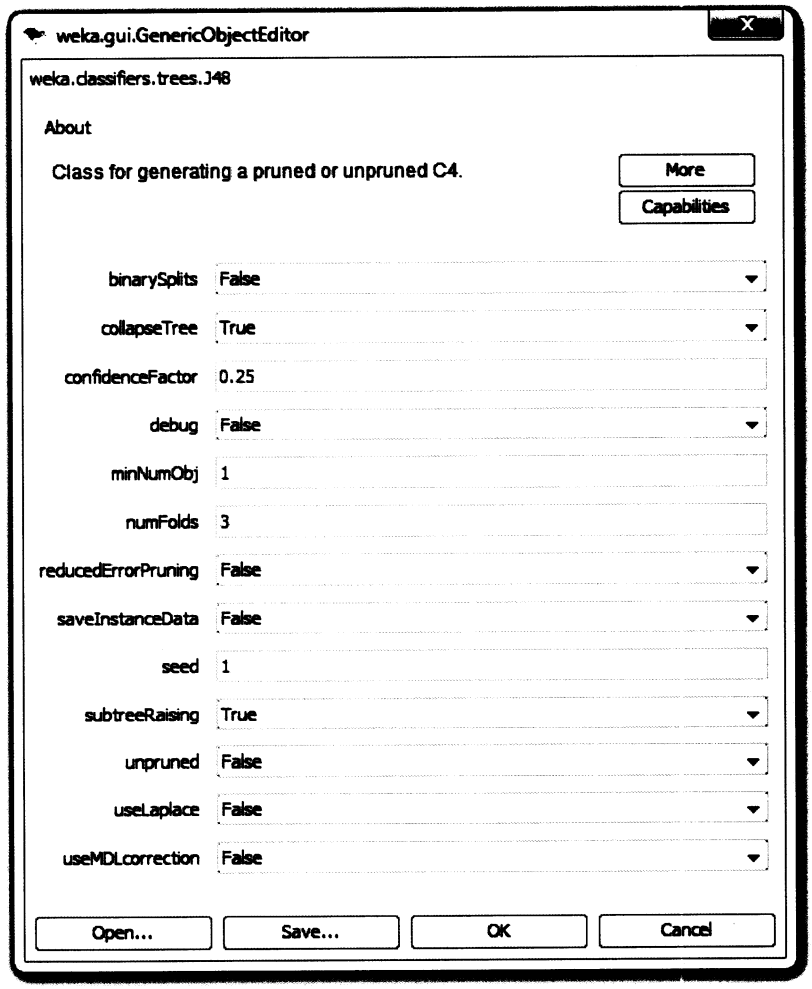


Figura A.18: Parâmetros do *J48 Decision Tree* na classificação do Ficheiro 2 sem selecção de atributos

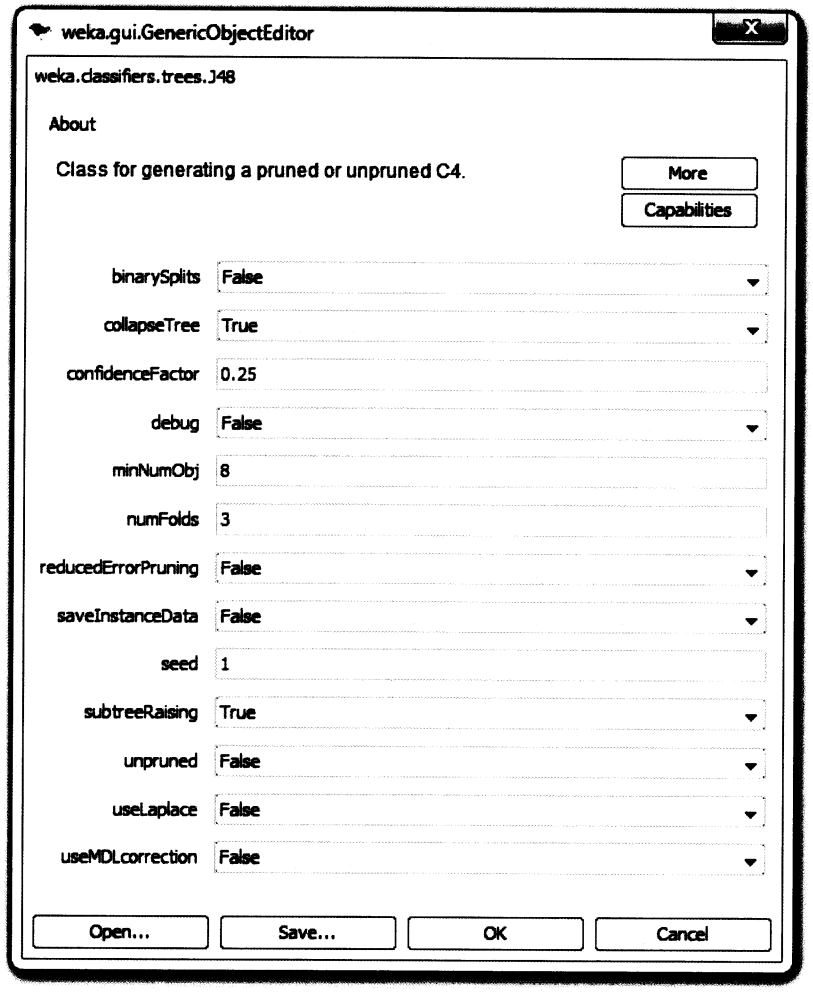


Figura A.19: Parâmetros do *J48 Decision Tree* na classificação do Ficheiro 3 sem selecção de atributos

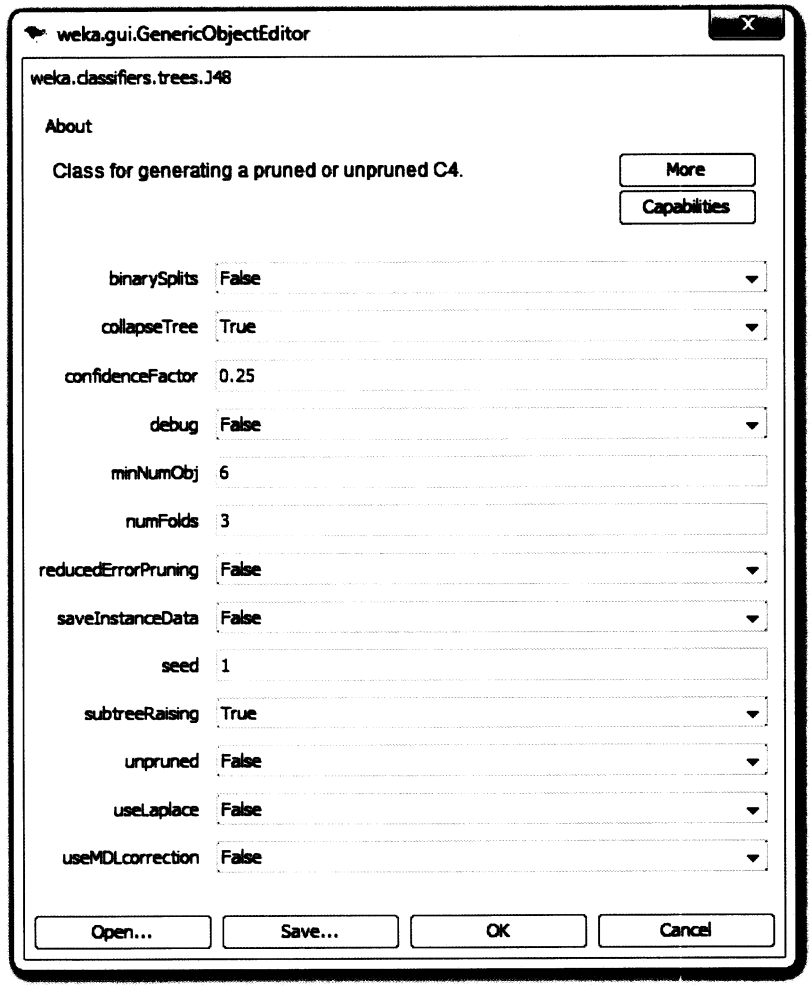


Figura A.20: Parâmetros do *J48 Decision Tree* na classificação do Ficheiro 4 sem selecção de atributos

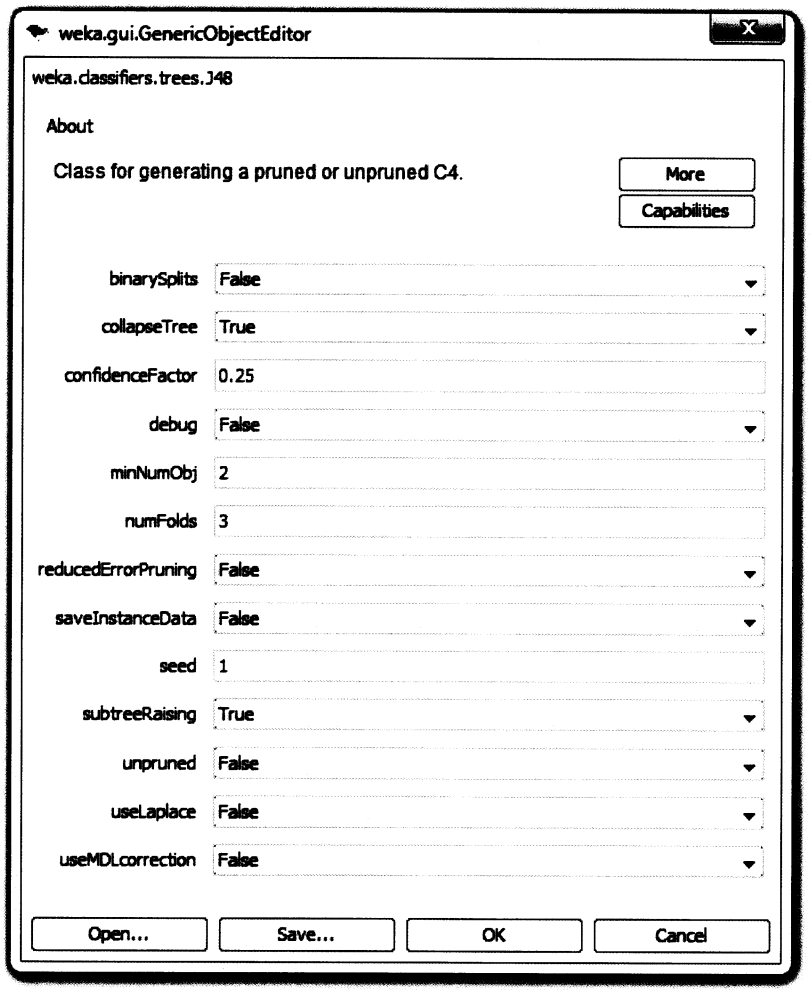


Figura A.21: Parâmetros do *J48 Decision Tree* na classificação do Ficheiro 1 com selecção de atributos

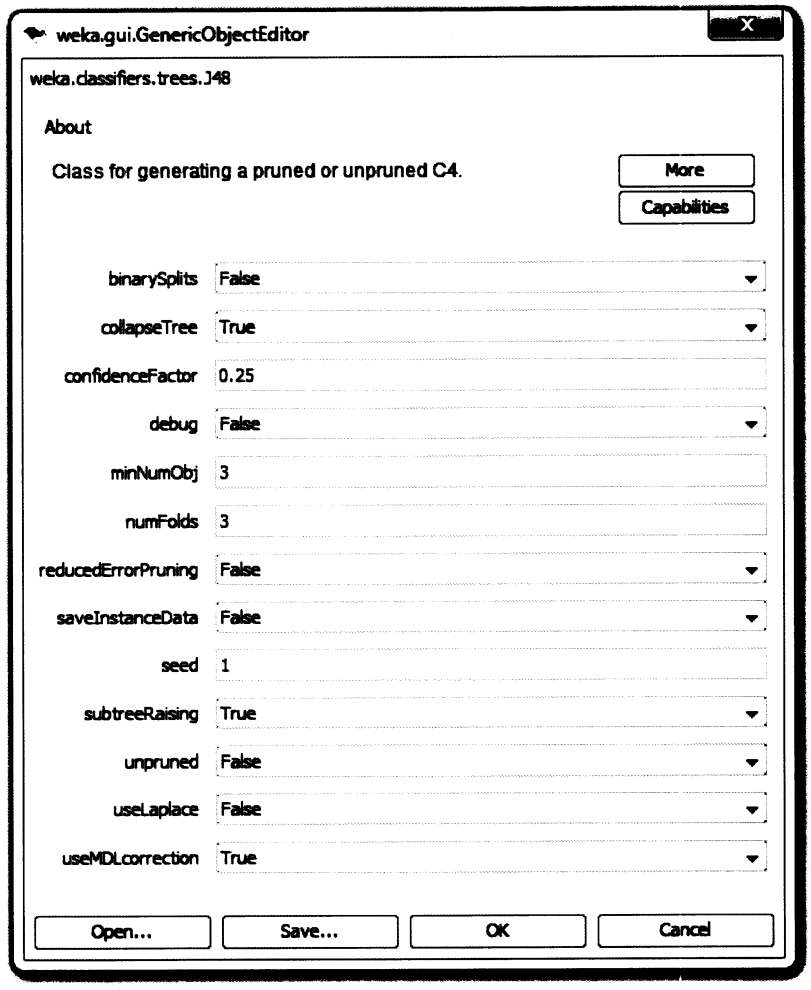


Figura A.22: Parâmetros do *J48 Decision Tree* na classificação do Ficheiro 2 com selecção de atributos

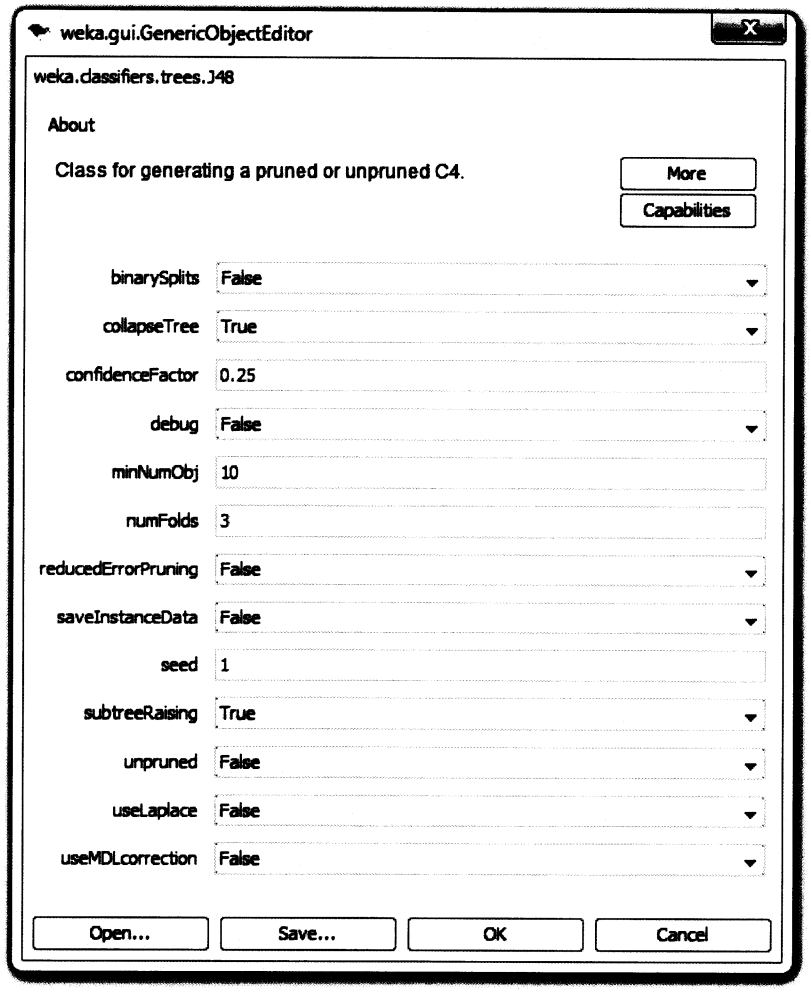


Figura A.23: Parâmetros do *J48 Decision Tree* na classificação do Ficheiro 3 com selecção de atributos

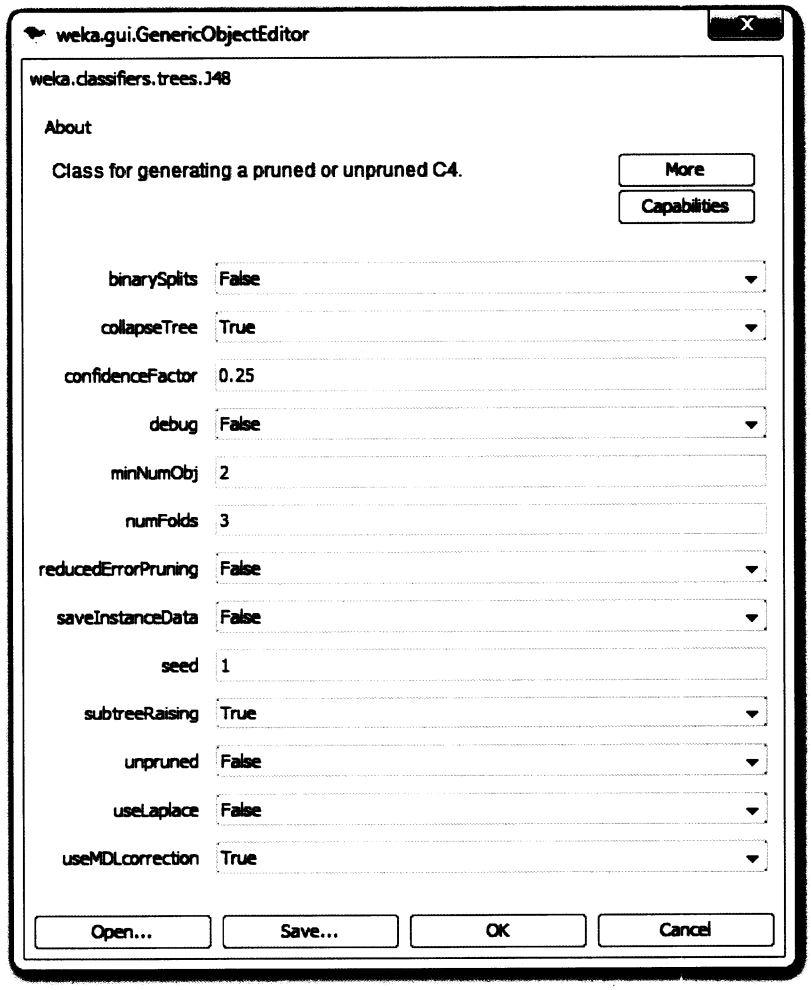


Figura A.24: Parâmetros do *J48 Decision Tree* na classificação do Ficheiro 4 com selecção de atributos

A.4 Support Vector Machines

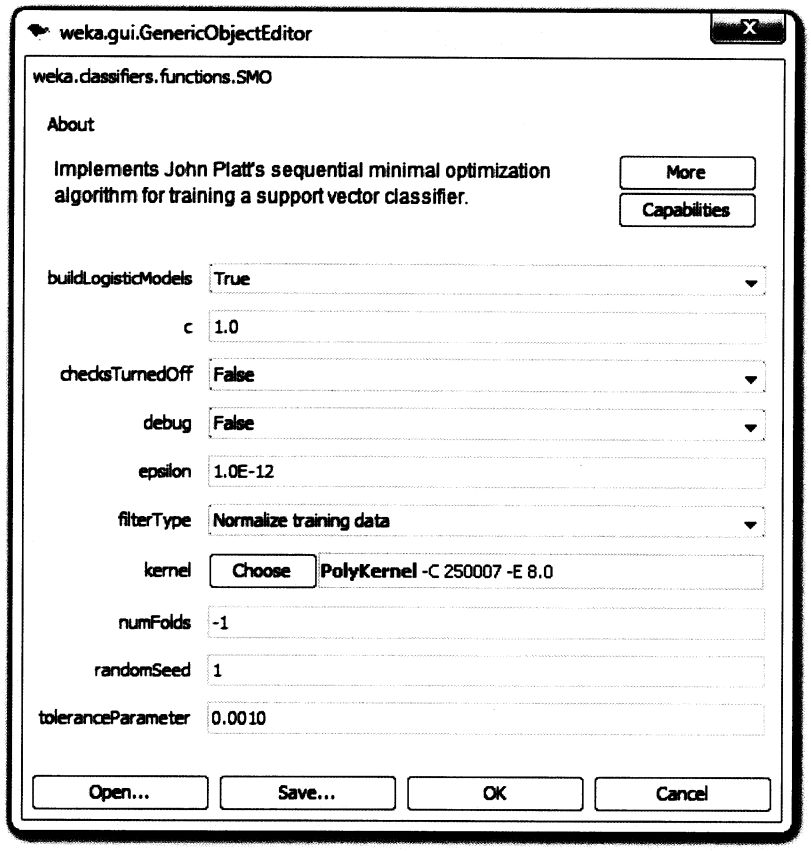


Figura A.25: Parâmetros do SVM na classificação do Ficheiro 1 sem selecção de atributos

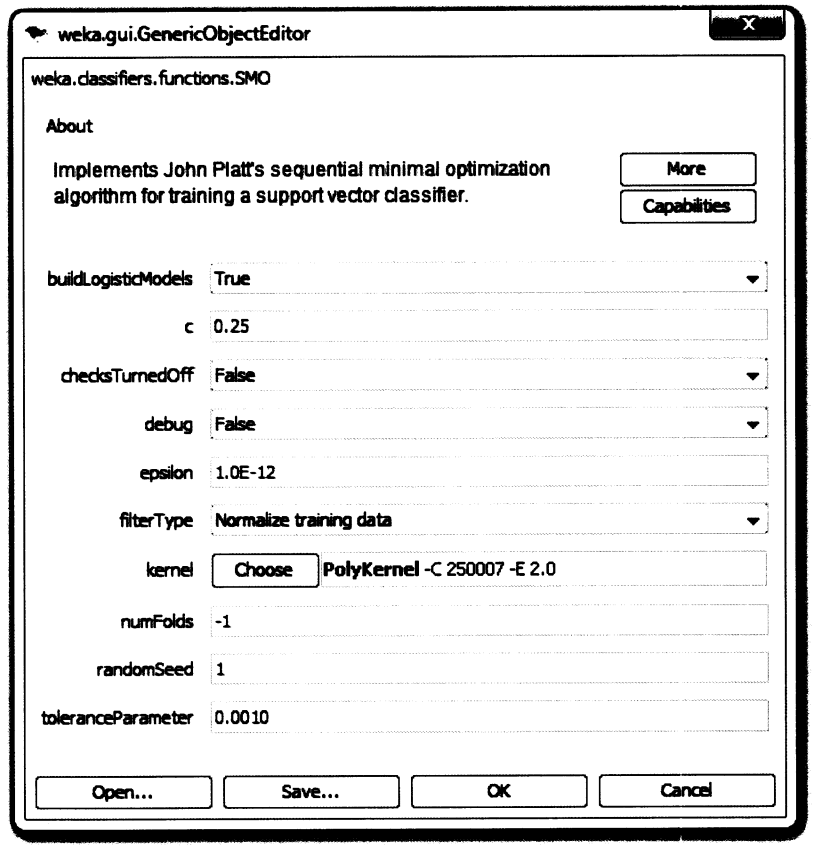


Figura A.26: Parâmetros do SVM na classificação do Ficheiro 2 sem selecção de atributos

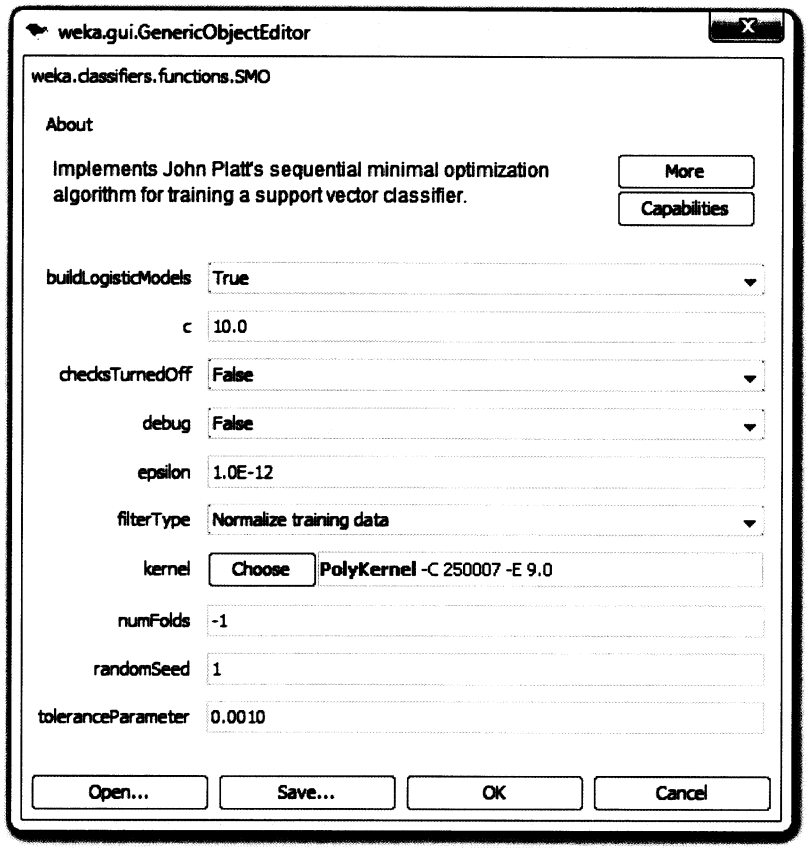


Figura A.27: Parâmetros do SVM na classificação do Ficheiro 3 sem selecção de atributos

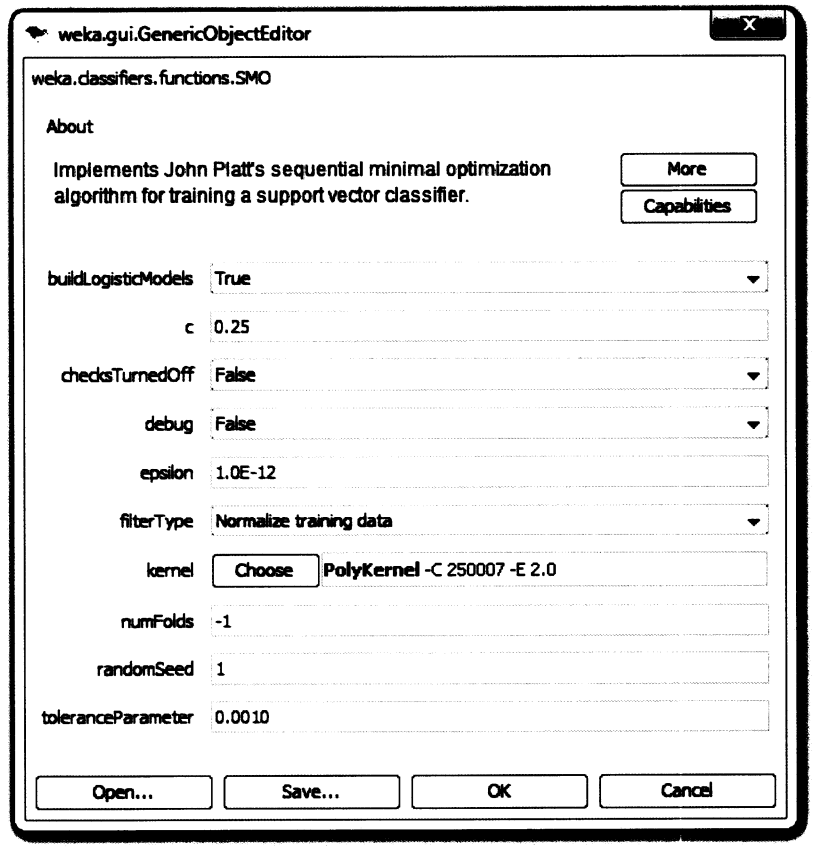


Figura A.28: Parâmetros do SVM na classificação do Ficheiro 4 sem selecção de atributos

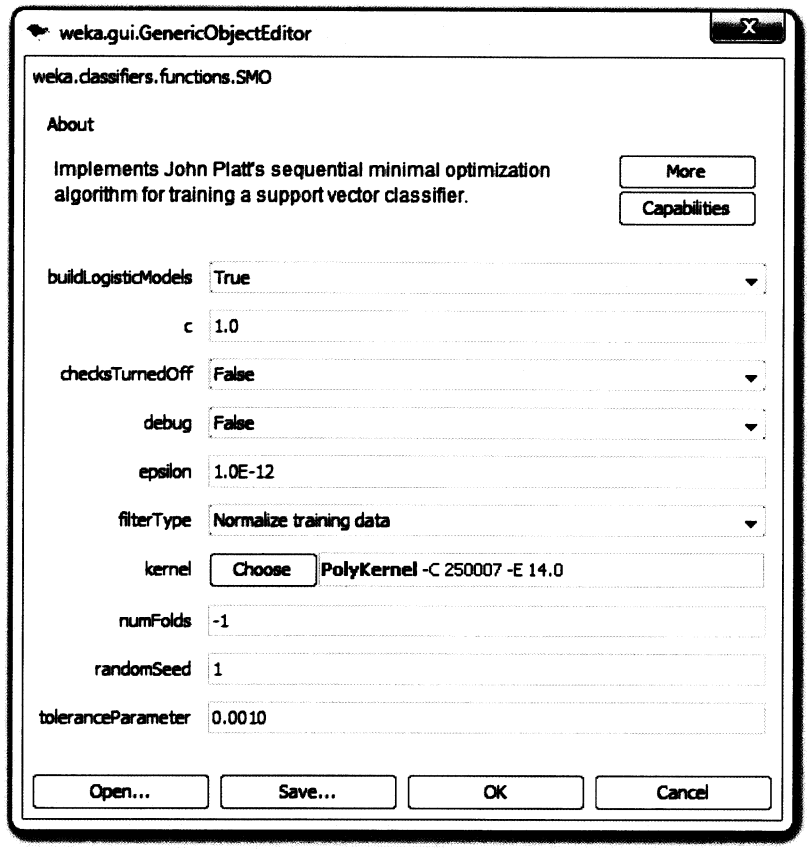


Figura A.29: Parâmetros do SVM na classificação do Ficheiro 1 com selecção de atributos

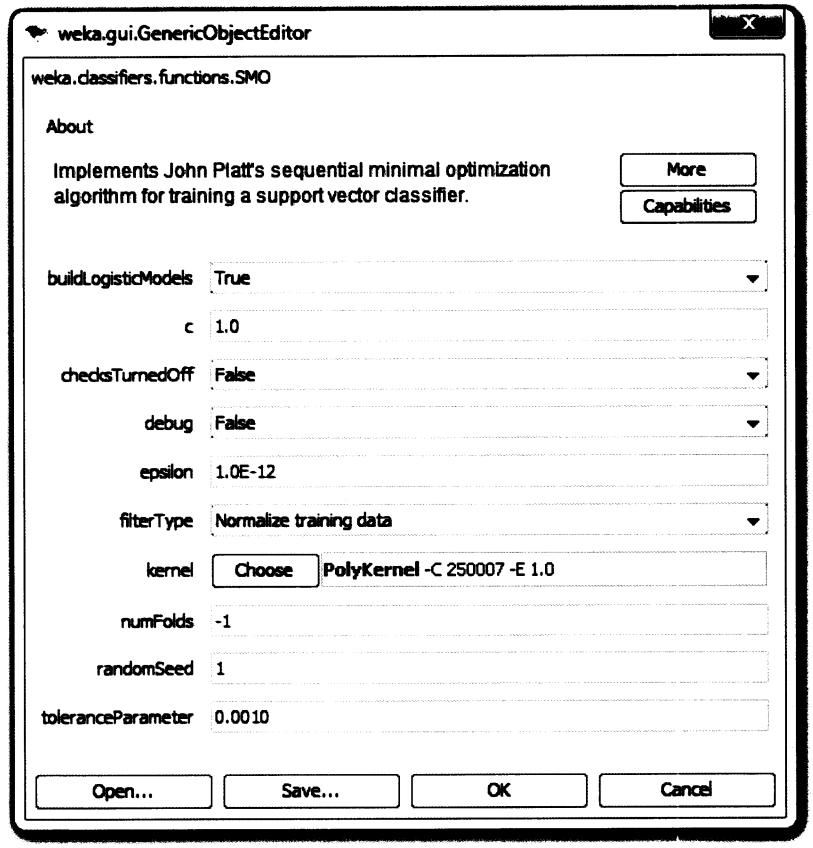


Figura A.30: Parâmetros do SVM na classificação do Ficheiro 2 com selecção de atributos

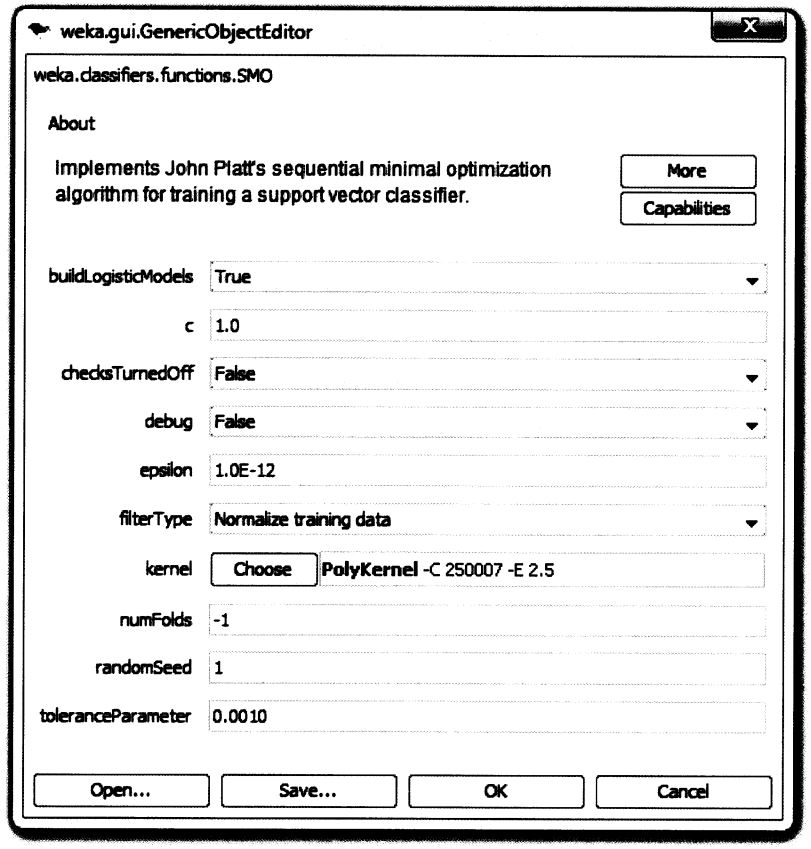


Figura A.31: Parâmetros do SVM na classificação do Ficheiro 3 com selecção de atributos

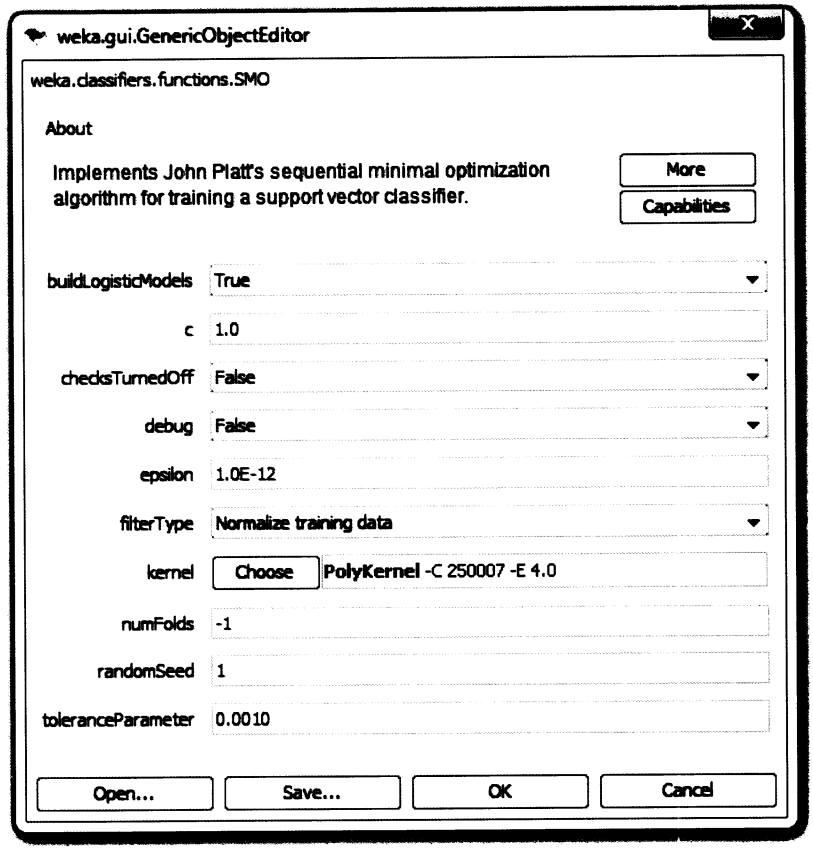


Figura A.32: Parâmetros do SVM na classificação do Ficheiro 4 com selecção de atributos