Universidade Do Minho

Mestrado Integrado Em Engenharia Biomédica

Sistemas de Aprendizagem e Extracção de Conhecimento

RELATÓRIO DE DISCUSSÃO

Braga, Abril 2013

Universidade Do Minho

Mestrado Integrado Em Engenharia Biomédica

Sistemas de Aprendizagem e Extracção de Conhecimento

RELATÓRIO DE DISCUSSÃO

Docente: César Analide

Discente: Marta Paes Moreira (54459)

Conteúdo

1 Introdução

O presente relatório, elaborado no âmbito da unidade curricular de Sistemas de Aprendizagem e Extracção de Conhecimento, pretende reflectir o trabalho de pesquisa e posterior análise crítica dos conteúdos abordados ao longo das aulas teórico-práticas, nomeadamente no domínio das Redes Neuronais Artificiais, dos Algoritmos Genéticos e da Extracção de Conhecimento. Procurou-se, mais do que simplesmente constatar informação teórica, explorar as problemáticas de cada um dos temas de um ponto de vista relevante para a compreensão das suas potenciais aplicações na área específica da Engenharia Biomédica. A exposição teórica é acompanhada, no final, por uma breve secção de discussão dos conhecimentos apreendidos.

APRENDIZAGEM

— Redes Neuronais Artificiais —

2 Redes Neuronais Artificiais

Uma rede neuronal artificial consiste numa tentativa de modelar matematicamente a estrutura e a capacidade de processamento de informação das redes neuronais biológicas, com base na assumpção consensual de que a essência de operação dos agregados neuronais reside no "*controlo através da comunicação*" [1]. Assim, partindo da interconexão de unidades básicas computacionais — designadas de nodos (ou neurónios artificiais) — com capacidade de aprendizagem, uma rede neuronal é capaz de armazenar conhecimento empírico e torná-lo disponível para utilização, assemelhando-se ao funcionamento do cérebro ao adquirir o conhecimento a partir de um dado ambiente, através de um processo de aprendizagem, e armazená-lo posteriormente nas conexões (ou sinapses) entre os nodos [2]. Relativamente à capacidade de processamento, é sabido que, no domínio da computação numérica e da manipulação simbólica, os computadores digitais disponíveis actualmente apresentam um desempenho bastante superior ao do cérebro humano, embora esta relação seja facilmente invertível considerando, por exemplo, a resolução de problemas perceptuais complexos [3].

2.1 Paradigma Biológico

Consoante estabelecido anteriormente, a estrutura e funcionalidade de um neurónio artificial deriva da observação do neurónio enquanto unidade funcional dos sistemas neuronais biológicos (Figura 1). No contexto biológico, a informação atinge o neurónio através das dendrites, sob a forma de estímulos, é processada no soma e finalmente encaminhada pelo axónio. Por sua vez, no domínio artificial, a recepção da informação processa-se através de canais de entrada com um peso associado — as **sinapses** —, o que significa que a informação é multiplicada pelo peso correspondente — que pode ser excitatório, inibidor ou nulo. Na fase de activação, a informação transmitida é integrada no neurónio (normalmente, por adição dos diferentes sinais) e, posteriormente, a **transferência** é calculada como função deste **valor de activação**, determinando a saída da rede [4, 5].

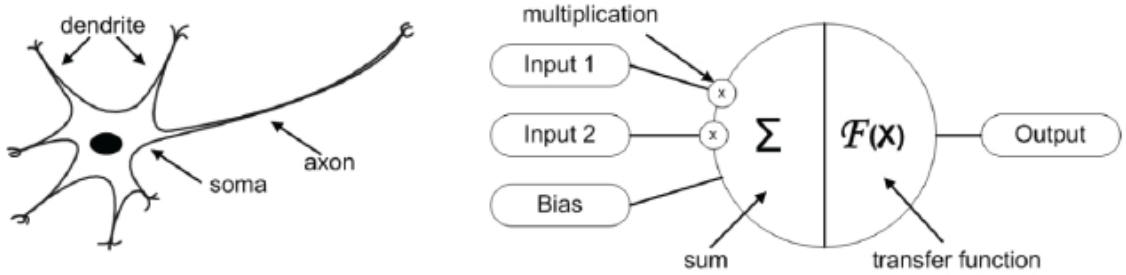

Figura 1: Estrutura comparativa de um neurónio biológico e um neurónio artificial.

A simplicidade do modelo artificial é matematicamente traduzida em (1), onde x_j, w_{ij} e y_j representam o valor de entrada, o peso e o valor de saída, respectivamente, e $\mathcal{F}$ representa a função de transferência. Pode ainda existir uma conexão extra b, denominada de *bias*, cuja entrada assume o valor +1, que estabelece uma determinada tendência ou inclinação no processo computacional [5, 6].

$$y(k) = \mathcal{F}(\sum_{i=0}^{m} w_{ij}(k) \cdot x_j(k) + b) \tag{1}$$

Em particular, a função de transferência define as propriedades do neurónio artificial e é seleccionada com base no problema que se pretende resolver. Genericamente, torna-se possível distinguir três tipos básicos de função de transferência [2, 5]:

1. **Função Limiar**, binária, que apresenta apenas dois valores de saída, conforme evidenciado em (2), que correspondem a situações em que a resposta do sistema aos valores de entrada seja superior ou inferior a um limiar específico, isto é, em que o ganho seja positivo ou negativo, respectivamente.

$$y = \begin{cases} 1, & \text{se } w_j x_j \geqslant \theta \\ 0, & \text{se } w_j x_j < \theta \end{cases} \tag{2}$$

   Este tipo de função é normalmente adoptado em modelos do tipo MCCulloch-Pitts (Figura 5), de acordo com a filosofia tudo-ou-nada pela qual se regem.

2. **Função Linear**, que se limita a executar transformações lineares simples sobre a soma das entradas pesadas e o *bias*.

3. **Função Sigmóide**, a mais amplamente utilizada no âmbito das redes neuronais artificiais, que é definida como uma função estritamente crescente que veicula um equilíbrio entre os comportamentos linear e não-linear. Contrariamente à função limiar, que permite apenas os valores binários 0 e 1, este tipo de função assume um leque de valores entre 0 e 1, e apresenta-se como sendo diferenciável.

2.2 Arquitecturas de Rede

A utilidade de um neurónio artificial adquire uma dimensão significativa apenas quando duas ou mais unidades são combinadas no sentido de construir uma rede neuronal artificial, habilitada a resolver questões complexas através do processamento de informação nas suas

unidades básicas de forma local, não linear, paralela e distribuída. A organização e distribuição das conexões entre os neurónios enquanto estruturas individuais é denominada de arquitectura de rede e pode, genericamente, ser dividida em duas classes: redes *feedforward* — por sua vez divididas em redes de camada única e redes multicamada — e redes *feedback*.

2.2.1 Redes Feedforward de Camada Única

Na sua forma mais simples, as redes *feedforward* apresentam-se como a composição de uma **camada de entrada**, constituída pelos nodos de origem — que, normalmente, não é contabilizada, devido à ausência operações de cálculo ao nível destas estruturas — e uma **camada de saída**, directamente envolvida na computação dos resultados. As conexões entre ambas são unidireccionais (convergentes ou divergentes) e, consequentemente, uma rede definida nestes parâmetros é considerada estritamente estática e acíclica [2, 6, 3].

2.2.2 Redes Feedforward Multicamada

Um caso particular de redes *feedforward* prende-se com a existência de uma ou mais **camadas ocultas** entre as camadas de entrada e saída, compostas por unidades neuronais que, de alguma forma, são responsáveis por intervir de forma útil sobre o percurso que separa os extremos da rede (Figura 2). Partindo dos nodos de origem, os padrões de activação são aplicados aos neurónios de uma primeira camada oculta, constituindo a saída desta camada a entrada da camada seguinte, e assim sucessivamente, até que se atinja a última camada, cujo conjunto de sinais de saída corresponde à resposta geral da rede aos padrões de activação inicialmente fornecidos [2, 3].

No que concerne a extensão das conexões entre os nodos das camadas, estabelece-se a distinção entre **redes completamente conectadas**, nas quais todos os nodos de cada uma das camadas estão conectados a todos os nodos da camada adjacente, e **redes parcialmente conectadas**, que apresentam algumas falhas ao nível das ligações sinápticas [2].

2.2.3 Redes Recorrentes

Contrariamente às redes apresentadas anteriormente, as redes recorrentes são sistemas dinâmicos, devido à existência de, pelo menos, uma conexão cíclica de *feedback*, que corresponde, neste contexto, aos casos em que a saída de um elemento do sistema infuencia, em parte, a entrada aplicada a esse mesmo elemento, levando à criação de um ou mais circuitos fechados para a transmissão de sinais em torno do sistema. A introdução deste tipo de conexões tem um impacto profundo sobre a capacidade de aprendizagem e o consequente desempenho da rede, forçando-a a um comportamento dinâmico não linear de natureza espacial

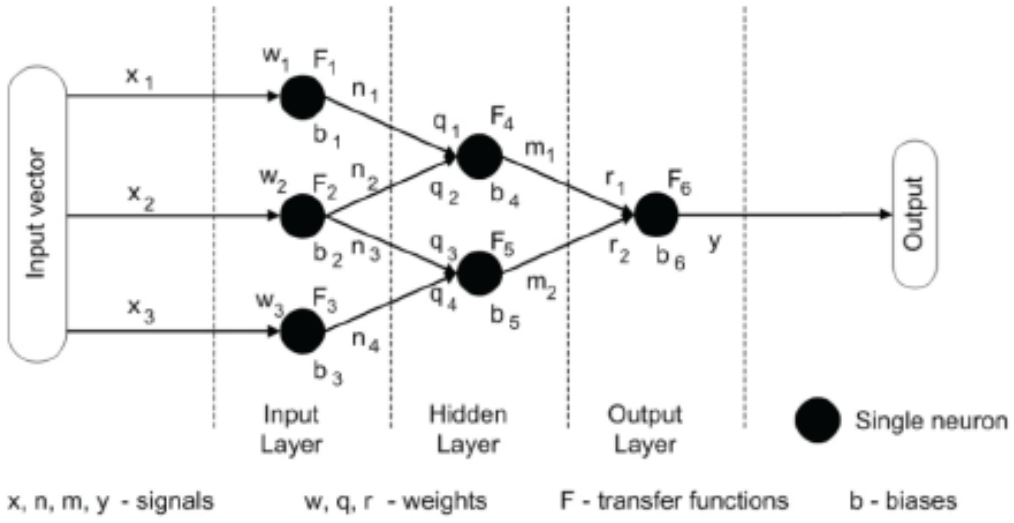

Figura 2: Rede *feedforward* multicamada.

e temporal, que requer necessariamente a definição de critérios de paragem [6, 2].

A topologia básica de uma rede deste género consiste numa estrutura **completamente recorrente**, na qual todos os nodos estão directamente conectados a todos os restantes nodos em todas as direcções (inclusive a si mesmos) (Figura 3). Estruturas como as redes de Hopfield, Elman, Jordan e bi-direccionais, entre outras, constituem casos especiais da topologia recorrente básica [5].

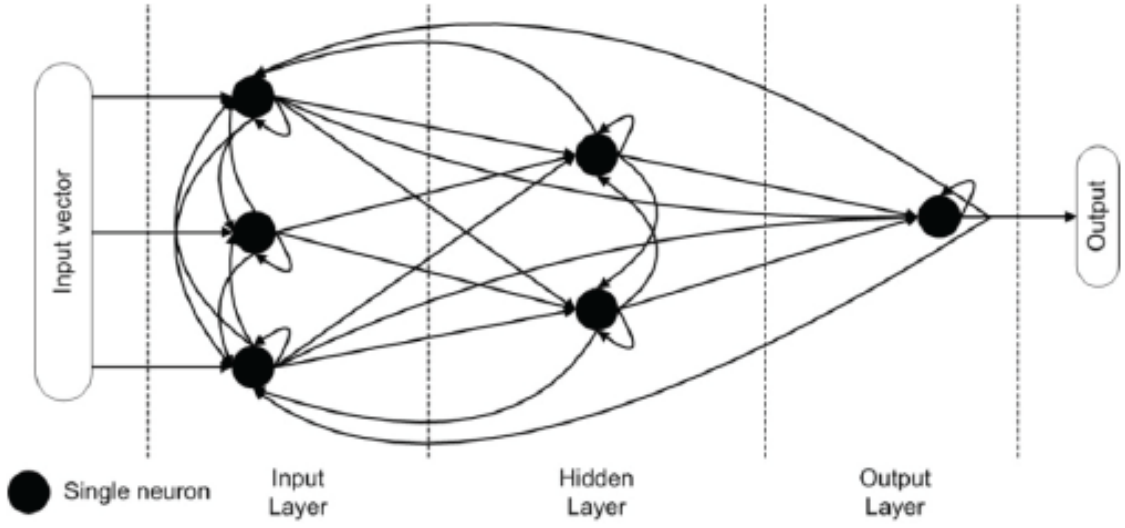

Figura 3: Rede *feedback*.

2.3 Paradigmas de Aprendizagem

A propriedade mais significativa de uma rede neuronal reside na sua capacidade de aprender a partir do ambiente em que se encontra inserida e melhorar o seu desempenho à medida que avança no processo de aprendizagem. Assim, torna-se fundamental considerar e definir as características do modelo do ambiente em que a rede opera, sendo possível distinguir dois paradigmas fundamentais: aprendizagem supervisionada e aprendizagem não supervisionada.

2.3.1 Aprendizagem Supervisionada

O método de aprendizagem supervisionada pressupõe o conhecimento de informação de experiências anteriores que permita apresentar inicialmente os resultados à rede e implica, neste sentido, a existência de um 'treinador' — um processo de controlo que conhece exactamente a resposta correcta para um dado conjunto de vectores de entrada. Assim, considerando uma dada função f de saída para m padrões de um conjunto de treino Ξ, individualmente constituídos por um vector de entrada e um vector de resposta, assume-se que, caso seja viável encontrar uma hipótese h que se aproxime de f para os membros de Ξ, então essa mesma hipótese será uma boa aproximação de f, sobretudo se o conjunto de treino contiver um grande número de padrões [7]. Considerando, numa primeira abordagem, a vertente **correctiva**, a aprendizagem é iniciada, partindo de um conjunto de casos de treino, através do cálculo do erro correspondente ao desvio entre o valor desejado e o valor efectivo de saída da rede. A magnitude do erro calculado determina o ajustamento dos pesos da rede no sentido de minimizá-lo. A aprendizagem é finalmente conseguida quando o erro é reduzido até valores considerados aceitáveis, após uma sucessão de iterações do algoritmo de treino e dos consequentes ajustamentos para todos os casos considerados. Por outro lado, a vertente **de reforço** não envolve a apresentação da resposta correcta, cabendo ao treinador apenas indicar se a resposta produzida pela rede perante uma determinada instância de treino é correcta *true* ou incorrecta *false*. A actualização dos pesos é levada a cabo com base nesta informação e, tipicamente, é concedida uma recompensa através do reforço dos pesos associados a respostas correctas e uma penalidade em caso contrário [4, 6].

No contexto da aprendizagem supervisionada correctiva, existem diversas formas de produzir os resultados apresentados à rede partindo do conjunto de treino. Um dos métodos considerados, o **método de batch**, dispõe da totalidade do conjunto e utiliza-o como um todo para calcular a função de hipótese h, existindo ainda uma variante deste método que modifica iterativamente uma dada hipótese até que uma mais aceitável seja obtida. De forma constrastante, o **método incremental** parte da selecção unitária de padrões do conjunto de

treino e utiliza-os individualmente para modificar uma dada hipótese, sendo que o método de selecção dos padrões pode ser aleatório — com substituição — ou cíclico — com iteração. No caso específico de a totalidade do conjunto ser disponibilizada e utilizada padrão a padrão, o método empregue designa-se por **método online**. Este tipo de método torna-se útil em situações em que, por exemplo, uma instância de treino constitui uma função da hipótese actual e da instância anterior, como seja no contexto de um classificador que decide sobre a próxima acção de um robô com base nos seus *inputs* sensoriais num dado instante [7].

2.3.2 Aprendizagem Não Supervisionada

Em contrapartida, a aprendizagem não supervisionada não recorre a qualquer agente externo no sentido de auxiliar o processo de correcção dos pesos, contando que não é sequer possível ter conhecimento de qual a solução a esperar da rede, e cabe à própria rede reorganizar-se de forma correspondente à sua decisão sobre quais os resultados mais adequados às características dos dados de entrada — concretizando, desta forma, a aprendizagem [4, 6]. Dentro desta classificação, torna-se possível destacar dois métodos, um dos quais igualmente **de reforço** — ou associativo — e um outro designado de **competitivo**. No primeiro caso, e conforme explicitado anteriormente, os pesos são ajustados de forma a optimizar a reprodução dos resultados desejados, enquanto que, no segundo, os elementos da rede competem entre si pelo direito de providenciar o resultado associado a uma dada entrada, contando que apenas um elemento pode responder à *query* e que este inibe necessariamente os restantes.

2.4 Regras de Aprendizagem

Enquanto problema de actualização da arquitectura de uma rede e dos pesos das conexões que a constituem, o processo de aprendizagem deve reger-se por determinadas regras que adequem o comportamento desta à execução eficiente das tarefas consideradas.

2.4.1 Regras Hebbianas

Partindo de experiências neurobiológicas, Hebb estabeleceu, em 1949, o seguinte postulado [8]: "*Quando o axónio de uma célula A se encontra próximo o suficiente para excitar uma célula B e, repetida ou persistentemente, participa no disparo desta, ocorre um processo de crescimento ou alteração metabólica, numa ou em ambas as células, tal que a eficiência de A, enquanto elemento envolvido no disparo de B, sofre um acréscimo.*". Sumariamente, este enunciado sugere que dois neurónios que se encontrem simultaneamente activos devem desenvolver um grau de interacção selectivamente superior àquele de dois neurónios cujas

actividades não estejam relacionadas — isto é, que apresentem uma interacção muito baixa ou nula —, podendo ser expandido sob a forma de uma regra bipartida [6]:

1. A activação síncrona de dois nodos de uma conexão — neste caso, designada de sinapse de Hebb — implica a incrementação progressiva da força dessa conexão, levando à formação de cadeias de associação, algumas das quais podem manter a activação subsequentemente durante um determinado período de tempo, como uma forma de memória de curto prazo (Figura 4);

2. Por outro lado, a activação assíncrona implica o enfraquecimento ou mesmo a eliminação da conexão associada aos nodos em questão.

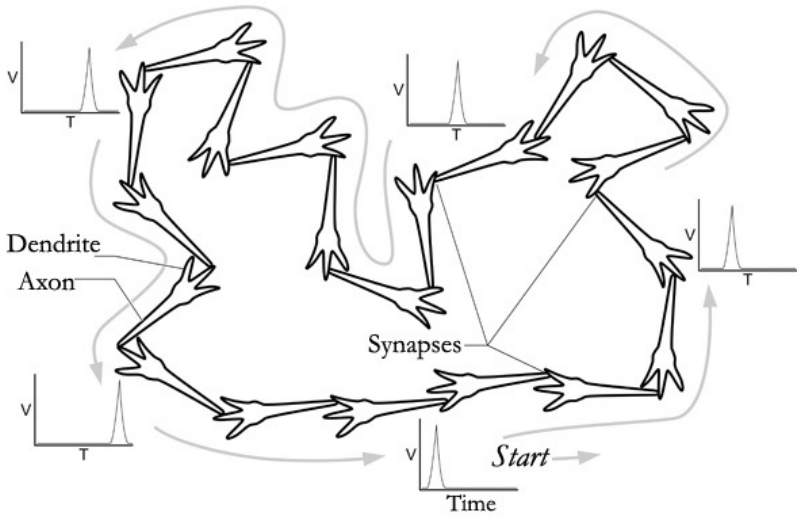

Figura 4: Hipótese original de Hebb: conjunto reverberante de células formado através de actividade correlacionada.

Matematicamente, a regra de Hebb pode ser descrita através da equação (3), em que x_i e y_j representam os valores de *output* dos neurónios i e j, respectivamente, conectados pela sinapse w_{ij}, e η simboliza a taxa de aprendizagem [3].

$$w_{ij}(t+1) = w_{ij}(t) + \eta y_j(t) x_i(t) \tag{3}$$

Uma particularidade importante desta rede prende-se com o facto de a aprendizagem ser feita localmente, isto é, a alteração do peso da sinapse depende apenas das actividades dos respectivos dois neurónios interligados, aspecto que simplifica significativamente, por exemplo, a complexidade do circuito de aprendizagem numa implementação VLSI (Very Large Scale Integration) [4, 3].

2.4.2 Regras de Correcção de Erros

Consoante mencionado anteriormente (Secção 2.3.1), no paradigma de aprendizagem supervisionada, a rede é alimentada com um dado resultado desejado (d) para cada um dos padrões de entrada, que pode ou não coincidir com o efectivo resultado de saída obtido (y). O princípio básico da regra de correcção de erros consiste na modificação dos pesos das conexões no sentido da diminuição gradual do erro dado por $(d-y)$. A aprendizagem de *perceptron* constitui em exemplo de regra cuja base assenta sobre o princípio de correcção de erros. Entende-se por *perceptron* um elemento composto por um único neurónio de pesos ajustáveis $(w_k, k = 1, 2, ..., p)$ e limiar θ (Figura 5). Dado um vector de entrada $x = (x_1, x_2, ..., x_p)^t$, a contribuição da rede para o neurónio corresponde a (4), assumindo a saída y o valor +1, para $v > 0$, e 0, para os restantes casos.

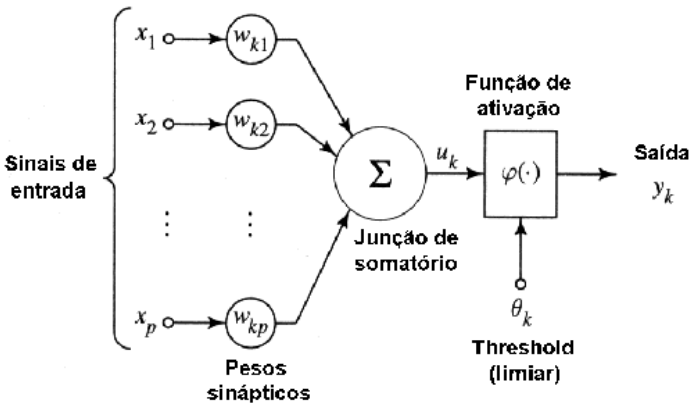

Figura 5: Neurónio de McCulloch-Pitts.

$$v = \sum_{j=1}^{n} w_k x_k - \theta \tag{4}$$

O algoritmo de aprendizagem que permite, partindo de um conjunto de padrões de treino, determinar os pesos e o limiar num *perceptron*, desenvolve-se considerando, numa primeira fase, a inicialização das variáveis mencionadas de forma aleatória, seguida do fornecimento de um vector padrão $(a_1, x_2, ..., x_n)^t$ e da subsequente avaliação da resposta do neurónio. A actualização dos pesos é efectuada com base em (5), sendo d o resultado desejado, t o número da iteração e $\eta(0.0 < \eta < 1.0)$ a taxa de aprendizagem.

$$w_j(t+1) = w_j(t) + \eta(d-y)x_j \tag{5}$$

Um outro algoritmo baseado no princípio de correcção de erros é o da aprendizagem com *back-propagation* [4].

2.4.3 Regras Competitivas

Contrariamente à aprendizagem de Hebb — em que múltiplas unidades de saída podem ser simultaneamente disparadas —, na aprendizagem baseada em regras competitivas estas unidades competem entre si pela activação. Como resultado, apenas uma delas se encontra activa numa dada altura, fenómeno conhecido como *winner-take-all*. Neste sentido, a aprendizagem competitiva tende a agrupar ou categorizar os dados de entrada, agrupando padrões semelhantes passíveis de serem representados através de uma única unidade. Ao nível da representação, a rede de aprendizagem mais simples consiste numa única camada de unidades de saída (Figura 6). Cada unidade de saída i liga-se a cada uma das restantes unidades de entrada x_j da rede através dos pesos w_{ij}, $j = 1, 2, ..., n$, e liga-se ainda a outras unidades de saída através de pesos inibitórios, apresentando, porém, um mecanismo de *self-feedback* com um peso excitatório [3].

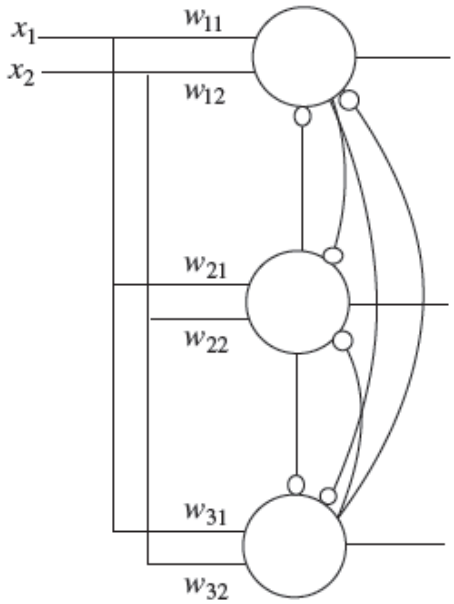

Figura 6: Rede de três unidades em competição.

Como resultado da competição, é considerada vencedora apenas a unidade i^* com a maior (ou menor) entrada de rede, de acordo com (6) ou (7). Quando se consideram vectores normalizados, as desigualdades (6) e (7) são equivalentes, e torna-se possível prevenir que um vector de peso se torne tão grande que saia vencedor do regime competitivo precocemente. Caso contrário, os restantes vectores nunca seriam actualizados, dando origem às designadas unidades mortas [4, 3].

$$w_i^* \cdot x \geqslant w_i \cdot x, \forall i \tag{6}$$

$$\|w_i^* - x\| \leqslant \|w_i - x\|, \forall i \tag{7}$$

Assim, garante-se que apenas os pesos da unidade vencedora são actualizados, encarregando-se posteriormente uma regra de actualização de rodar o vector na direcção do vector de entrada x_j. Este procedimento pode ser concretizado utilizando diferentes regras, de entre as quais se destacam [4]:

1. **Actualização com constante de aprendizagem**

   A actualização dos pesos é dada por (8), representando a constante de aprendizagem um número real entre 0 e 1 que diminui à medida que a aprendizagem progride.

   $$\Delta w_m = \eta x_j \tag{8}$$

   A plasticidade da rede pode ser controlada de forma a que tais correcções sejam mais drásticas nas primeiras iterações e se tornem progressivamente mais graduais, nas iterações seguintes.

2. **Actualização das diferenças**

   A actualização dos pesos é dada por (9) e a correcção é proporcional à diferença entre ambos os vectores.

   $$\Delta w_m = \eta(x_j - w_m) \tag{9}$$

3. **Actualização em lote**

   As correcções de peso são calculadas, acumuladas e, após um dado número de iterações, adicionadas aos pesos. A utilização desta regra garante alguma estabilidade no processo de aprendizagem.

É possível retirar da regra de aprendizagem competitiva que a rede não pára de aprender a não ser que a taxa de aprendizagem η seja igual a zero e que um padrão específico de entrada pode disparar diferentes unidades de saída, desde que em iterações distintas. Esta questão sugere alguns problemas de estabilidade dos sistemas de aprendizagem, dado que um sistema é apenas é considerado estável caso nenhum dos padrões no conjunto de treino sofrer modificações na sua categoria após um número finito de iterações. A estabilidade

pode ser atingida forçando a taxa de aprendizagem a diminuir gradualmente à medida que o processo de aprendizagem se densenvolve em direcção a zero. Contudo, este congelamento artificial levanta um outro problema relacionado com a plasticidade — isto é, a capacidade de adaptação a novos dados —, conhecido como Dilema da Estabilidade-Plasticidade de Grossberg em aprendizagem competitiva [4].

A forma mais conhecida de aprendizagem competitiva é a quantização vectorial para compressão de dados, que tem sido amplamente utilizada no processamento de imagem e de fala, com o propósito de optimizar a eficiência do armazenamento, transmissão e modelação. Os objectivos que lhe são inerentes prendem-se com a representação de um conjunto ou distribuição de vectores de entrada com um número relativamente pequeno de vectores de peso ou um *codebook*. Uma vez construído e atingido um consenso sobre este por parte do transmissor e do receptor, o *codebook* necessita apenas de transmitir ou armazenar o índice do vector de peso correspondente ao vector de entrada, que será aquele que, no *codebook*, se encontrar mais próximo. Outros exemplos de aplicação desta regra são as redes SOM (Self-Organizing-Maps) — por exemplo, as redes de Kohonen e ART (Adaptive Resonance Theory) [4, 6, 3].

2.4.4 Regras Estocásticas

Contrariamente às regras definidas até este ponto, o paradigma estocástico não recorre a uma rede com o propósito de encontrar ou verificar determinados resultados para dados conjuntos de entrada, mas antes de modelar o comportamento estatístico de uma parte do universo em estudo, tomando uma dada distribuição de padrões a partir da qual procura construir um modelo interno capaz de gerar, de forma independente, uma distribuição semelhante. Assim, de acordo com esta regra de aprendizagem, a operação da rede deve incluir alguns passos probabilísticos, o que imputa necessariamente instabilidade ao seu estado, uma vez que, mesmo considerando entradas que não variem, o estado das unidades constituintes da rede varia constantemente. Em última análise, deixando uma rede deste género operar livremente durante um período de tempo suficientemente longo, durante o qual são registados os estados que esta atravessa, é possível construir uma distribuição de probabilidades sobre esses mesmos estados — embora seja necessário assegurar que a distribuição é apenas medida após a rede ter atingido um estado de equilíbrio [9].

As **máquinas de Boltzmann** constituem um exemplo de rede neuronal estocástica, mais especificamente de redes recorrentes e simétricas constituídas por unidades binárias. Neste caso específico, um subconjunto de **neurónios visíveis** interage com o ambiente, contrariamente a um outro subconjunto de **neurónios ocultos**. As máquinas de Boltzmann operam em dois modos fundamentais, que, se alternados com frequência aproximadamente

semelhante, permitem reduzir a entropia cruzada entre a distribuição livre da rede e a distribuição alvo [6, 3, 9].

1. $Phase^{+}$, em que as unidades visíveis se encontram fixados ao valor de um padrão específico. Quando é atingido o equilíbrio, os pesos são incrementados entre quaisquer unidades cujo estado corresponda a 'On' (+1). Esta fase é repetida um elevado número de vezes, com cada padrão α a ser fixado com uma frequência correspondente à probabilidade P_{α}^{+} que se pretende modelar;

2. $Phase^{-}$, em que tanto as unidades visíveis como ocultas podem operar livremente. Assim que se atinge o equilíbrio (e nunca antes), recolhem-se amostras suficientes para obter uma média fiável das actividades das unidades, decrementando-se posteriormente o peso entre quaisquer duas unidades que se encontrem no estado 'On' — num processo denominado de desaprendizagem.

O objectivo final desta vertente de aprendizagem prende-se, então, com o ajustamento dos pesos das conexões de forma a que os estados das unidades visíveis satisfaçam uma dada distribuição de probabilidade desejada. A alteração no peso w_{ij} da conexão é dada por (10), em que η é a constante de aprendizagem e $\tilde{\rho}_{ij}$ e ρ_{ij} são as correlações entre os estados das unidades i e j quando a rede opera nos modos $Phase^{+}$ e $Phase^{-}$, respetivamente.

$$\Delta w_{ij} = \eta(\tilde{\rho}_{ij} - \rho_{ij} \tag{10}$$

Com base nestes pressupostos, a aprendizagem de Boltzmann pode ainda ser encarada como um caso especial da aprendizagem de correcção de erros, na qual o erro é medido não como a diferença directa entre os resultados desejados e os resultados obtidos, mas como a diferença entre as correlações entre as saídas de dois neurónios em condições de operação fixas e livres [3, 9].

2.4.5 Regras Baseadas na Memória

As regras baseadas na memória, associadas à designada aprendizagem com atraso (ou 'preguiçosa'), diferem das regras apresentadas anteriormente pelo facto de se limitarem a armazenar os padrões, numa memória de pares entrada-saída, adiando quaisquer generalizações até que surja um novo padrão nunca antes fornecido à rede que force a examinação da relação deste com aqueles armazenados. Assim, a função objectivo é estimada localmente e de forma distinta para cada novo padrão apresentado, o que constitui uma vantagem relativamente aos retantes métodos, em que esta função é estimada uma única vez para a totalidade do conjunto de padrões. Por outro lado, o esforço computacional exigido por

este tipo de abordagem é significativamente mais elevado, uma vez que a maior parte do processamento se dá não aquando da apresentação dos exemplos de treino, mas na etapa da classificação [6].

2.5 Exemplo de Aplicação Biomédica

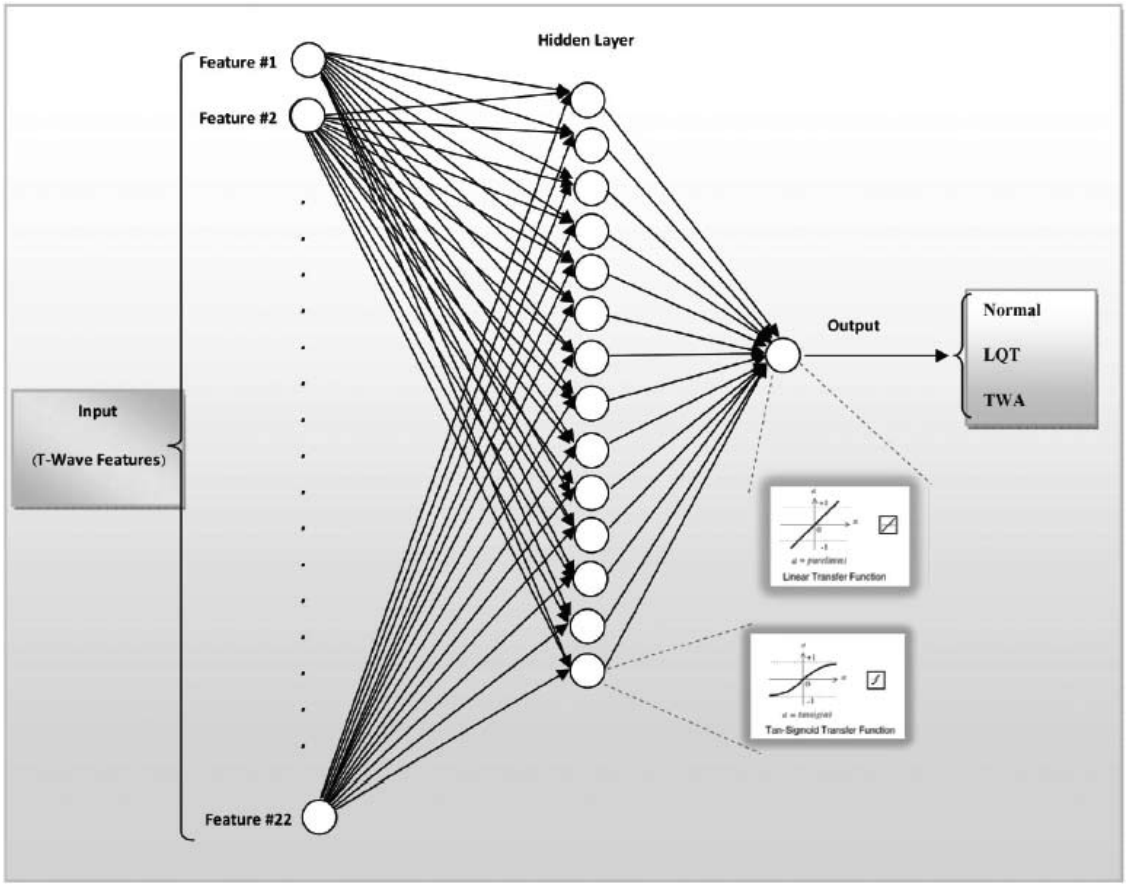

Figura 7: Rede neuronal para a previsão da profundidade de anestesia em contexto cirúrgico.

Um exemplo possível de aplicação dos modelos descritos em ambiente clínico consiste na previsão da profundidade de anestesia em contexto cirúrgico. Neste caso específico, foi utilizada uma rede composta por 15 neurónios de entrada, 20 neurónios na camada oculta e um neurónio na camada de saída. A fase de treino baseou-se em dados recolhidos de 25 pacientes — 8 dos quais reservados para uma posterior fase de teste —, ao longo de 20 horas, devidamente convertidos de forma a permitir a extracção das características a utilizar na entrada da rede, e na definição do índice BIS [1] como saída desejada. Por último, os

[1]Número adimensional entre 100 (estado alerta) e 0 (estado isoeléctrico), que permite monitorizar a profundidade de anestesia considerando como normais valores entre 40 e 60, para fins de intervenção cirúrgica.

dados reservados foram utilizados na fase de teste da rede, permitindo comparar os valores à saída com os valores patentes no monitor BIS. Refira-se que foi necessário, após os primeiros testes, aplicar algumas alterações à rede ilustrada na Figura 7, nomeadamente a diminuição do número de neurónios de entrada para 10 e a adição de uma outra camada de abstracção com 10 neurónios, bem como a omissão de cinco características que apresentavam uma correlação fraca com o índice de BIS. Os resultados obtidos com esta segunda estrutura, treinada ao longo de 30000 iterações, figuram na Imagem 8 [10].

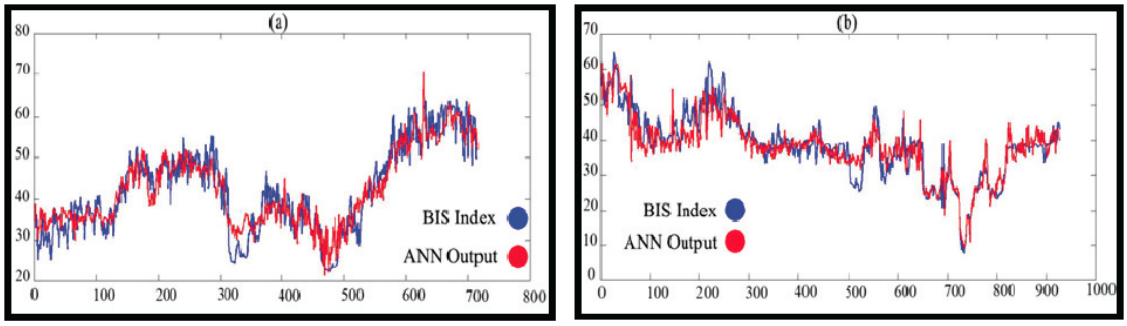

Figura 8: Resultados do monitor BIS e da rede neuronal, associados a correlações de 93% para os dados do paciente (a) e 90% para os do paciente (b).

APRENDIZAGEM

— Algoritmos Genéticos —

3 Algoritmos Genéticos

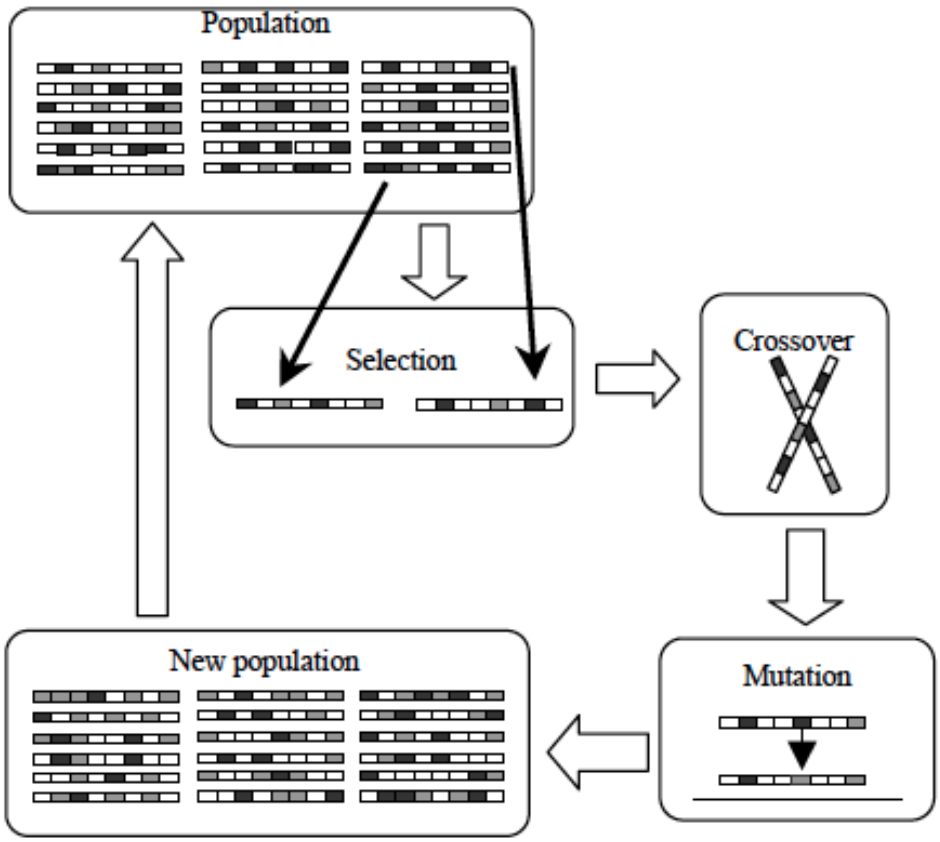

Figura 9: Processo típico de um algoritmo genético.

Os algoritmos genéticos constituem uma técnica iterativa robusta de procura e optimização desenvolvida com base nos princípios da teoria genética e evolucionária, que codifica as variáveis de decisão de um problema em *strings* finitas de alfabetos com uma determinada cardinalidade. Neste contexto, as *strings* candidatas à solução do problema denominam-se de **cromossomas**, os alfabetos de **genes** e os valores desses mesmos genes de **alelos**. Um outro conceito fundamental no estudo dos algoritmos genéticos reside na noção de **população**, isto é, do conjunto de soluções candidatas disponíveis, uma vez que o seu tamanho — normalmente definido pelo utilizador — condiciona seriamente a escalabilidade e o desempenho deste tipo de algoritmos. Por exemplo, populações demasiado reduzidas potenciam a convergência prematura e, consequentemente, conduzem a soluções abaixo das desejáveis, enquanto que populações demasiado grandes podem representar gastos desnecessários de tempo computacional. Naturalmente, o objectivo dos algoritmos genéticos prende-se com a evolução no sentido da obtenção de soluções desejáveis, pelo que se torna necessário implementar medi-

das que determinem a **aptidão** relativa das soluções candidatas existentes. Estas medidas podem consistir em funções objectivas — um modelo matemático ou simulação computacional — ou funções de carácter subjectivo — nas quais cabe a um humano a escolha entre as melhores e as piores soluções [11, 12].

Uma vez codificado o problema, a evolução (Figura 9) dá-se executando o seguinte conjunto de etapas [11]:

1. **Inicialização**

   A população inicial de soluções candidatas é gerada de forma aleatória, a partir do espaço de procura.

2. **Reprodução**

   Posteriormente à inicialização da população (ou à criação de uma nova população de descendentes), os valores de aptidão das soluções candidatas são avaliados, seguindo-se o processo de selecção, que aloca um maior número de cópias das soluções que provem ter valores de aptidão mais elevados, impondo um mecanismo de *survival of the fittest* à população. Alguns dos mecanismos utilizados nesta etapa são descritos em detalhe em seguida, na Secção 3.1.

3. **Cruzamento**

   O cruzamento parte da combinação de dois ou mais cromossomas para criar descendentes idealmente mais aptos, podendo ser implementado recorrendo a qualquer um dos mecanismos definidos na Secção 3.2.1. Refira-se que o desempenho geral do sistema depende consideravelmente de um mecanismo de cruzamento bem estruturado.

4. **Mutação**

   Enquanto que o cruzamento opera sobre dois ou mais cromossomas, a mutação ocorre ao nível local e modifica aleatoriamente a solução. No geral, este processo envolve uma ou mais modificações ao nível dos alelos e pode ser efectuado de diversas formas, consoante explicitado na Secção 3.2.2.

5. **Substituição**

   Por último, as novas soluções geradas devem substituir a população que lhe deu origem, existindo igualmente variadas abordagens passíveis de utilizar, como sejam a substituição elitista e outras descritas na Secção 3.3.

6. Repetição das etapas 2-5 até que uma critério de paragem se verifique.

3.1 Selecção

O processo de selecção, que actua maioritariamente ao nível dos cromossomas, rege-se pela ideia de priorização dos indivíduos mais aptos — isto é, aqueles que se encontram mais próximos da solução —, ao permitir que estes passem os respectivos genes para a geração seguinte, incentivando, assim, a que os membros da população se tornem progressivamente mais aptos, à medida que as gerações progridem. O valor de aptidão é determinado com base numa função objectivo ou através de um julgamento subjectivo específico para o problema em questão. Por sua vez, o processo crítico de selecção dos melhores indivíduos — designado de **pressão de selecção** —, deve ser cuidadosamente ponderado, contando que, se a pressão aplicada for demasiado baixa, a taxa de convergência no sentido da solução óptima é igualmente baixa, e, caso seja demasiado elevada, existe a possibilidade de o sistema cair num local óptimo devido à perda de diversidade na população [13]. Neste contexto, torna-se possível distinguir, essencialmente, três classes de mecanismos ditos tradicionais: **selecção proporcionada**, **selecção por ranking** e **selecção por torneio**.

3.1.1 Selecção Proporcionada

Na selecção proporcionada, o número de vezes que se espera que um indivíduo se reproduza é igual à sua aptidão dividida pela média das aptidões da totalidade da população. Um dos mecanismos mais comuns de implementação deste mecanismo é o **método da roleta** (Figura 10), que equivale conceptualmente à atribuição, a cada indivíduo, de uma porção da roleta de área proporcional à sua aptidão. Nestes parâmetros, a roleta é rodada n vezes para uma população de n indivíduos, sendo que, em cada uma destas acções, o indivíduo que se encontra sob o marcador é seleccionado como progenitor para a geração seguinte. Uma das principais críticas apontadas a este método prende-se com o facto de, para uma população reduzida, o acaso que lhe está inerente poder resultar numa alocação desproporcionada de descendentes aos indivíduos [14, 12].

Pese embora o método da roleta seja, na prática, o mais generalizado, a selecção proporcionada pode ainda socorrer-se de outros métodos, como sejam a **amostragem determinística**, que difere do anterior no sentido de que apenas a parte inteira do resultado da divisão da aptidão individual pela aptidão média da população é utilizada para efeitos de selecção. A totalidade da fracção destina-se apenas à ordenação dos indivíduos e ao consequente preenchimento dos lugares livres na nova população com aqueles que apresentem melhor colocação; a **amostragem estocástica dos restos**, que recupera, numa primeira fase, o mecanismo de selecção determinístico, recorrendo depois ao método da roleta para o preenchimento das posições livres; e a **amostragem estocástica universal**, na qual, em

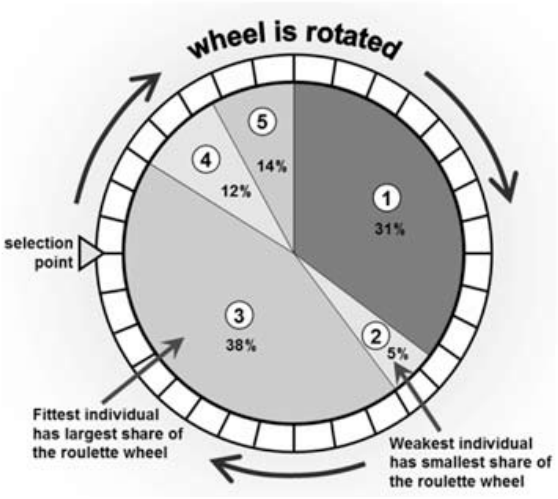

Figura 10: Esquematização do método da roleta.

vez de se efectuarem n rotações da roleta para seleccionar n indivíduos, se efectua apenas uma rotação, mas com n marcadores igualmente espaçados. [13, 12].

Uma das grandes desvantagens da utilização de métodos de selecção proporcionada encontra-se relacionada com a convergência prematura, uma vez que, ao colocar o foco de acção apenas sobre os indivíduos mais aptos, a variância de aptidão da população torna-se rapidamente baixa, deixando de haver uma base diferencial de exploração, o que, em última análise, acaba por provocar a estagnação da evolução.

3.1.2 Selecção por Ranking

Na selecção por *ranking*, os indivíduos da população são ordenados de acordo com um valor de *rank*, calculado com base na respectiva aptidão, não dependendo, assim, da sua aptidão absoluta. Este processo reduz a ocorrência de situações de atribuição de descendentes a um grupo restrito de indivíduos mais aptos, controlando a pressão de selecção no sentido da manutenção dos valores esperados independentemente das diferenças de aptidão absoluta. Considerando a variante linear, os indivíduos de uma população são pontuados ($rank$) e é-lhes atribuído um valor esperado ($ExpVal$), de acordo com a equação definida em (11). Uma vez concluída a atribuição de valores esperados, a amostragem da população — isto é, a selecção de progenitores — pode ser efectuada recorrendo a métodos de amostragem estocástica universal [12].

$$ExpVal(i,t) = Min + (Max - Min)\frac{rank(i,t) - 1}{n - 1} \tag{11}$$

3.1.3 Selecção por Torneio

A selecção por torneio apresenta similaridades relativamente à selecção por *ranking* ao nível da pressão de selecção, mas revela-se bastante mais eficiente, a nível computacional. Na selecção por torneio, dois indivíduos da população são escolhidos de forma aleatória, escolhendo-se ainda (e de forma igualmente aleatória) um número $r \in [0, 1]$. Caso se verifique que r é menor do que um dado parâmetro k — correspondente a uma probabilidade de selecção pré-determinada, geralmente superior a 0.5 — o mais apto dos dois indivíduos é seleccionado como progenitor; em caso contrário, é seleccionado o indivíduo menos apto dos dois. Posteriormente, ambos são devolvidos à população original, podendo voltar a ser seleccionados. Releve-se que o número de indivíduos seleccionados pode variar entre o número mínimo de dois e o máximo de n, respeitante ao número total de indivíduos que compõem a população [13, 12].

3.2 Operadores Genéticos

Os operadores genéticos são utilizados no âmbitos dos algoritmos genéticos com o propósito de criar diversidade — mutação — e combinar soluções existentes em novas soluções — cruzamento. A diferença essencial entre ambos reside no nível de operação, respectivamente unário e binário.

3.2.1 Cruzamento

A forma mais simples de cruzamento envolve **uma única posição** (Figura 11), escolhida de forma aleatória, e a troca, entre dois progenitores, dos blocos que a sucedem, dando origem a dois descendentes. Um dos aspectos negativos que surge associado a este tipo de cruzamento é o denominado **bias posicional**, uma vez que os *schemas* passíveis de serem criados ou destruídos através do cruzamento dependem fortemente da localização dos *bits* no cromossoma, o que levanta ainda o problema da possibilidade de perda de funcionalidade, devido ao facto de não ser viável assumir que os *bits* relacionados com a funcionalidade se encontram em posições contíguas. Ao favorecer a manutenção de *schemas* curtos intactos, o cruzamento num único ponto leva, muitas vezes, à preservação de **hitchhikers** — *bits* que não fazem parte do *schema* desejado, mas que, por estarem próximos numa *string* são acoplados ao *schema* benéfico aquando da sua reprodução. Uma última desvantagem deste mecanismo diz respeito à existência de um aparente tratamento preferencial de *loci* específicos, já que os segmentos trocados entre dois progenitores contêm sempre os extremos finais das *strings*. No sentido de reduzir o efeito destas condicionantes, é prática comum utilizar, em alternativa, **duas posições** de cruzamento, escolhidas de forma aleatória, ou

mesmo recorrer ao **cruzamento uniforme parametrizado** (Figura 11), em que a troca acontece em cada posição com uma dada probabilidade p (tipicamente, entre 0.7 e 0.8) [12].

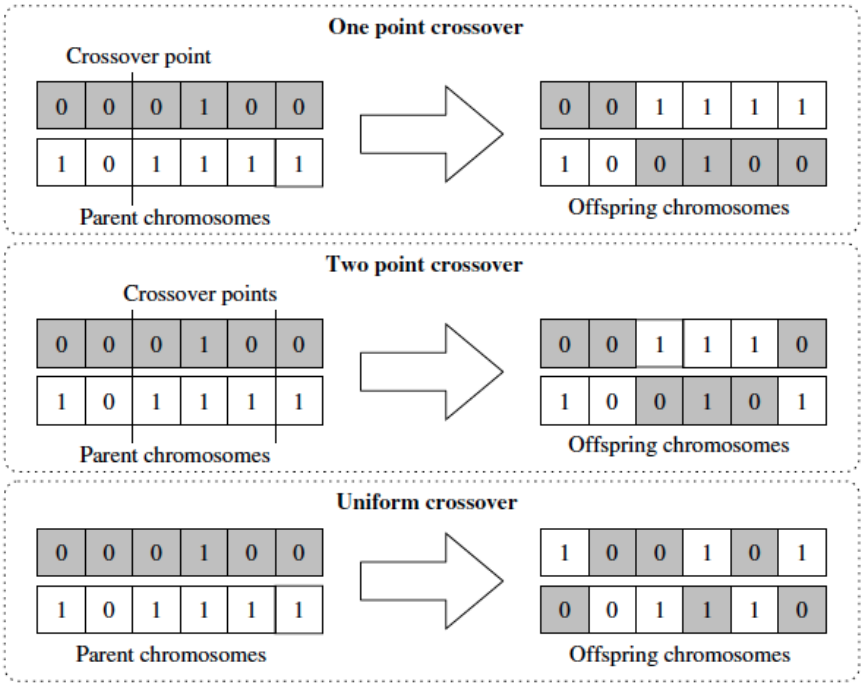

Figura 11: Métodos de cruzamento de posição única, de duas posições e uniforme.

3.2.2 Mutação

Contrariamente a outros métodos de computação evolucionários, o papel central de instrumento de variação e inovação nos algoritmos genéticos não recai sobre a mutação, mas sim sobre o operador de cruzamento. Neste contexto, a mutação surge, então, não como fonte única de variação, mas como elemento que assegura a população contra a estagnação permanente num *locus* em particular, e apresenta diversos mecanismos de implementação, consoante o tipo de genoma considerado, de entre os quais se destacam a **mutação uniforme**, a **mutação uniforme múltipla** e a **mutação gaussiana** [14]. Na mutação uniforme, um símbolo v_k, $(k \in 1, 2, ..., n)$, aleatoriamente seleccionado é substituído por um número v'_k aleatório contido no intervalo $[v_k^{max}, v_k^{min}]$, dando origem ao cromossoma $s_v^{t+1} = (v_1, ..., v'_k, ..., v_m)$. É viável seleccionar $n \in 1, 2, ..., n$ símbolos em simultâneo, escolhidos de forma aleatória, passando este mecanismo a denominar-se de mutação uniforme múltipla. Por último, na mutação gaussiana, todos os símbolos de um cromossoma são mutados de forma a originar $s_v^{t+1} = (v'_1, ..., v'_k, ..., v'_m)$, em que $v_k = v_k + f_k$ e f_k se reporta a um número aleatório retirado de uma distribuição gaussiana específica. Considere-se, nas

apresentações anteriores, que s_v representa o cromossoma seleccionado para a operação, k a posição do símbolo no cromossoma, $t \in 0, 1, ..., t$ o número da geração e v_k^{min}, v_k^{max} os limites inferior e superior, respectivamente [14].

3.3 Substituição

A introdução das novas soluções na população, isto é, a transição para uma nova geração, pode ser encarada, genericamente, de três formas distintas, que podem envolver a delecção de todos os cromossomas do conjunto actual de soluções e a substituição dos mesmos com o mesmo número de cromossomas recém-criados (**delete-all**); ou apenas a delecção de n cromossomas antigos e a substituição por n novos (**steady-state**), existindo a possibilidade de verificar que não se adicionam cromossomas duplicados à população (**steady-state-no-duplicates**). A vertente mais comum é a de *delete-all*, já que é de fácil implementação e não requer a definição de parâmetros, contrariamente aos mecanismos de *steady-state*, que implicam a definição do número e de quais os cromossomas a apagar e substituir [11].

3.4 Parâmetros

Para além dos parâmetros especificamente inerentes à tarefa que se pretende desenvolver, um algoritmo genético compreende um conjunto próprio de parâmetros que podem ser ajustados no sentido de optimizar o seu desempenho. Pese embora não seja viável particular valores, pode considerar-se um conjunto de sugestões baseadas em conhecimentos empíricos prévios.

3.4.1 Dimensão da População

Conforme explicitado na secção introdutória, a definição da dimensão da população assume um papel fundamental na escabilidade e desempenho de um algoritmo genético, contando que populações desproporcionadas — quer acima, quer abaixo dos valores adequados — representam desvantagens a nível computacional, seja por conduzirem à convergência prematura ou a gastos desnecessários de tempo computacional. A dimensão típica das populações situa-se entre os 100 e os 1000 cromossomas [12, 15].

3.4.2 Taxa de Sobrevivência

Este parâmetro, que está directamente relacionado com a etapa de substituição, determina a percentagem de cromossomas de uma dada população que sobrevive em cada geração. No caso de se revelar demasiado reduzido, existe o risco de a população perder rapidamente

soluções que, no conjunto actual, não perfilam entre as melhores, mas que podem vir a evoluir nesse sentido, paralelamente a outras que, parecendo mais favoráveis, acabam por adoptar o sentido contrário; de forma semelhante, uma taxa de sobrevivência excessivamente elevada conduz a um inevitável desperdício de tempo computacional em candidatos fracos. Consideram-se aceitáveis taxas na ordem dos 30% [12, 15].

3.4.3 Taxa de Mutação

A definição da taxa de mutação determina a probabilidade de ocorrência de uma mutação em cada uma das gerações, sendo que se aponta como desejável um *schema* que siga um raciocínio conducente, em cada geração e após o processo de avaliação da totalidade dos indivíduos, a que a população da geração seguinte seja constituída por cópias inalteradas dos **S** cromossomas mais aptos — **selecção elitista** — e pelo maior número de cromossomas mutados (**M** vezes cada) que seja possível introduzir na população **P**, dado o confinamento de espaço disponível [15].

3.4.4 Taxa de Cruzamento

No seguimento do ponto anterior, a taxa de cruzamento surge como parâmetro determinante da quantidade de indivíduos que, não estando envolvidos no processo de mutação, representam o cruzamento entre dois cromossomas aleatoriamente seleccionados que façam parte do referido agregado **S**. Este mecanismo acelera, geralmente, a convergência e, conforme explorado na Secção 3.2.1, pode apresentar resultados mais ou menos benéficos, de acordo com o problema em questão [12, 15].

3.5 Exemplo de Aplicação Biomédica

Um exemplo prático da aplicação de algoritmos genéticos na área clínica consiste no apoio ao desenvolvimento de terapias *antisense*, que consistem na introdução, directamente no interior de uma célula, de um fragmento de DNA ou RNA capaz de se hibridizar com uma sequência complementar do mRNA localizado no citoplasma ou no núcleo, inibindo os mecanismos de tradução. Pretende-se, com esta abordagem, minimizar ou impedir o desenvolvimento de estados patológicos infecciosos e distúrbios genéticos. Assim, torna-se importante, numa primeira fase, modelar um oligonucleótido sequencialmente próximo do gene de interesse. Este processo deve obedecer a um conjunto específico de restrições, quer ao nível do comprimento mais adequado para o oligonucleótido, quer ao nível da sua composição, entre outros. Neste universo específico, uma abordagem com base em algoritmos genéticos revela-se bastante apropriada, já que o problema proposto implica um espaço de

procura volumoso (constituído por todos os oligonucleótidos passíveis de emparelharem com um dado gene), bem como uma complexidade adequada.

A primeira tarefa reside na escolha de uma representação, que pode ser, por exemplo, um *array* de caracteres (A,C,G,T) com uma determinada extensão e um inteiro que expressa a posição inicial da região de *binding* no DNA-alvo. A função de avaliação irá, posteriormente, pontuar o oligonucleótido com base num determinado número de critérios, atribuindo-lhe uma pontuação S em cada uma dessas características. A função de avaliação retorna, depois, um número $0.0 < F < 1.0$, com $F = S_1 \cdot w_1 + S_2 \cdot w_2, ...$, em que as constantes w_n representam os pesos associados a cada critério.

EXTRACÇÃO DE CONHECIMENTO

4 Extracção de Conhecimento

Como primeira definição, pode encarar-se, de um prisma simplista, a extracção de conhecimento como sendo uma análise automática e exploratória de repositórios de dados de grande volume. Esta questão revela-se particularmente relevante na actualidade, uma vez que a acessibilidade e a abundância de informação assistem a um crescimento exponencial, que requer necessariamente o desenvolvimento de técnicas e ferramentas capazes não só de acompanhá-lo, mas também de optimizar e direccionar a sua utilização no sentido do enriquecimento de áreas como o *marketing* e, em particular, a prestação de cuidados de saúde.

4.1 Processo

O processo de extracção de conhecimento desenvolve-se iterativamente ao longo de sete passos essenciais, de entre os quais quatro se reportam ao pré-processamento dos dados e à consequente preparação para a etapa de *mining* [16]. Podem considerar-se ainda dois passos adicionais, que dizem respeito à tomada de decisão relativamente ao domínio da aplicação, nomeadamente à definição de objectivos e estratégias de abordagem ao problema; e à selecção e génese do conjunto de dados sobre o qual se pretende operar a extracção [17].

4.1.1 Limpeza

Um dos primeiros (e mais importantes) passos em qualquer tarefa de processamento de dados corresponde à verificação de quão correctos — ou, pelo menos, de quão conformes a um conjunto de regras — são os valores contidos nos dados. Este processo encontra-se intimamente relacionado com a aquisição e definição de dados, compreendendo, genericamente, as fases de definição e determinação do tipo de erros, de procura e identificação de instâncias de erro, e de correcção desses mesmos erros. A última fase torna-se especialmente difícil de automatizar fora de um domínio rígido e bem definido, assim como a identificação de erros que envolvem relacionamentos entre um ou mais campos. Genericamente, pode considerar-se que a limpeza de dados se centra sobre duas irregularidades em particular: os valores em falta e o ruído [16, 18, 19].

1. **Valores em falta**

   A problemática dos valores em falta é transversal à totalidade dos conjuntos de dados sujeites a pré-processamento, seja devido a omissões propositadas na fonte de informação ou simplesmente por razões intrínsecas ao processo de recolha de dados de várias proveniências, especialmente propício a perdas inadvertidas, e é, muitas vezes, agravada por comportamentos negligentes relativamente à estruturação das bases de

dados e à planificação dos procedimentos utilizados para a recolha de dados. Apesar de existirem várias formas de colmatar estas ocorrências, nem todas respeitam a preservação das relações entre atributos, pelo que é indispensável a ponderação na escolha de abordagem e a consideração das possíveis consequências. Por exemplo, a forma mais intuitiva de reagir perante um elemento incompleto seria simplesmente ignorá-lo, o que não se revelaria muito eficaz, já que a percentagem de valores em falta por atributo varia consideravelmente, mas outros métodos aparentemente mais plausíveis, como a utilização de constantes globais ou de medidas de tendência central, influenciam os dados, de certa forma. A abordagem mais popular consiste em determinar o valor mais provável do conjunto de dados, com base nos atributos dos restantes elementos, e utilizá-lo para preencher o valor em falta.

2. **Ruído**

   Uma outra irregularidade que pode influenciar de forma negativa o processo de extracção de conhecimento é a presença de erros ou variâncias aleatórios na medição de uma variável. A eliminação desta componente pode ser alcançada, genericamente, por intermédio de técnicas de *smoothing* de dados, de entre as quais se destacam o *binning*, a regressão e a detecção de *outliers*. A primeira técnica procede à suavização através da ordenação, partição e distribuição dos dados por estruturas designadas de *bins*, aplicando, posteriormente, uma consulta à vizinhança, com base na qual atribui um valor específico a cada um dos elementos de cada um dos *bins*. Por sua vez, a regressão linear envolve a descoberta da linha ou superfície multidimensional que melhor adapta dois ou mais atributos e a análise de *outliers* socorre-se de técnicas como, por exemplo, o *clustering* para organizar valores similares em grupos e, assim, potenciar a identificação de valores desajustados dos restantes.

4.1.2 Integração

Na grande maioria das vezes, o conjunto de dados que se pretende submeter ao processo de extracção de conhecimento consiste, na verdade, em vários conjuntos provenientes de diferentes fontes, pelo que é necessário proceder à sua integração e lidar, numa primeira fase, com as problemática da heterogeneidade, da redundância e da inconsistência de informação [16]. Actualmente, um conceito amplamente disseminado para efeitos de integração é o de *data warehousing*, que veicula um ponto único e consistente de acesso aos dados de uma entidade ou organização, possibilitando a desacreditação dos conceitos de compartimentação e dispersão [19].

4.1.3 Selecção

Numa outra perspectiva, há que considerar o facto de, na prática, uma fracção considerável dos atributos que são apresentados aos esquemas de aprendizagem serem claramente irrelevantes ou redundantes, pelo que se torna importante seleccionar apenas um subconjunto de atributos a utilizar no processo de aprendizagem. Embora existiram esquemas que procuram, de forma independente, seleccionar os atributos apropriadamente, ignorando aqueles que não apresentam qualquer utilidade, o seu desempenho pode ser melhorado através de mecanismos de pré-selecção. A grande maioria destes mecanismos pressupõe a pesquisa do espaço de atributos (Figura 12) em uma de duas direcções: do topo para a base — **forward selection** — ou da base para o topo — **backward elimination** —, sendo que, em cada patamar, uma alteração local ao subconjunto de atributos é efectuada, quer pela adição, quer pela delecção de um único atributo, respectivamente. O método de *forward selection* parte de um conjunto vazio de atributos, que vai sendo iterativamente preenchido e avaliado, de forma a clarificar se as adições se reflectem numa melhoria de desempenho dos subconjuntos. A pesquisa termina quando nenhum atributo adicionado contribui para a melhoria do desempenho do subconjunto actual. O método de *backward elimination* funciona de forma análoga, mas contrária, partindo de um conjunto completo de atributos e procedendo iterativamente à sua eliminação [19].

4.1.4 Transformação

A etapa de transformação — que precede, em casos específicos como o do *data warehousing*, a de selecção — tem como objectivo transformar e consolidar os dados no sentido de tornar o *mining* mais eficiente e facilitar a compreensão dos padrões encontrados. De entre as técnicas utilizadas nesta etapa, destacam-se o smoothing, a construção de atributos, a agregação, a discretização, a normalização e a generalização [19].

4.1.5 Mining

Muitas vezes tratado erroneamente como sinónimo de extracção de conhecimento, o *data mining* consiste apenas numa das etapas que compõem a globalidade do processo de extracção de conhecimento, pese embora constitua o seu núcleo duro, ao envolver a inferência de algoritmos que exploram os dados, desenvolvem o modelo e são responsáveis pela descoberta de padrões até então desconhecidos. O tipo de *mining* a utilizar — por exemplo, classificação, regressão ou *clustering* — depende fortemente dos objectivos delineados para a extracção de conhecimento, bem como das etapas anteriores, considerando-se essencialmente duas variantes: **mining supervisionado** e **mining descritivo**. A grande maioria dos me-

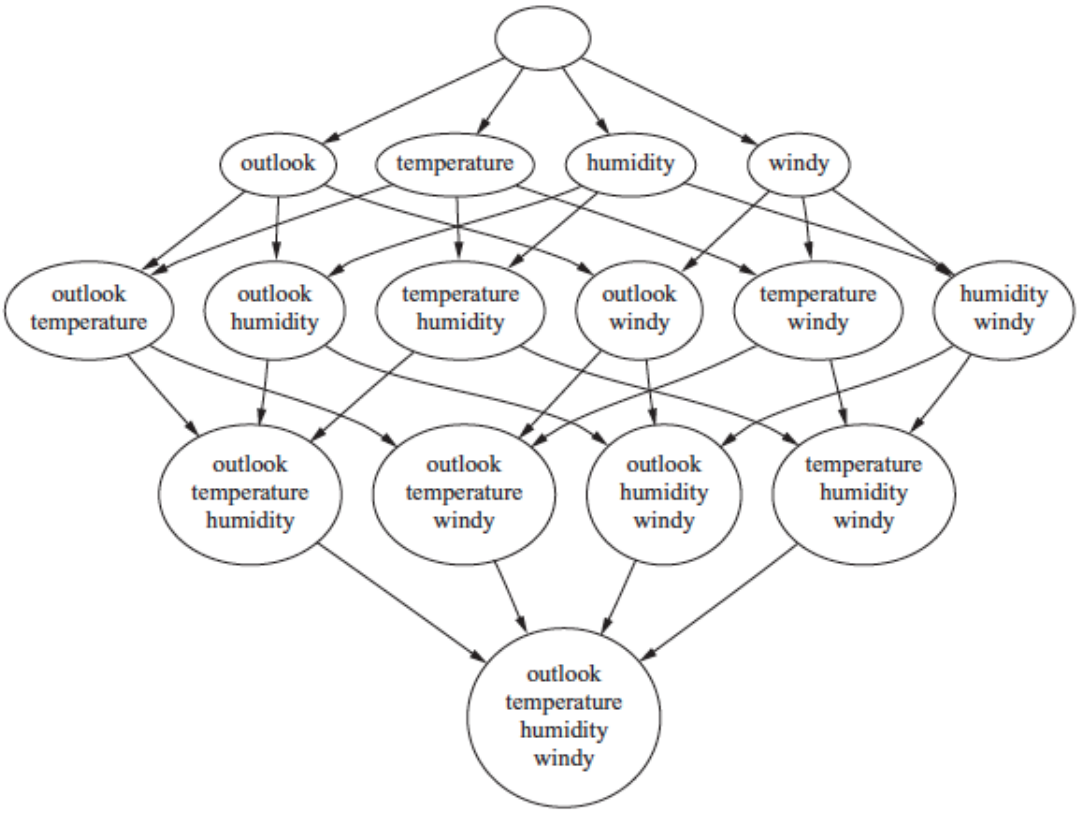

Figura 12: Espaço de atributos para um conjunto de dados meteorológicos.

canismos são baseados na aprendizagem indutiva, na qual um modelo é construído explícita ou implicitamente através da generalização de um número suficiente de exemplos de treino, assumindo-se que o modelo assim treinado pode ser aplicado a casos futuros. Uma vez definida a estratégia a aplicar, torna-se necessário decidir sobre o método específico a utilizar na procura de padrões, que pode passar, por exemplo, pelo domínio das redes neuronais ou das árvores de decisão, antes de proceder à aplicação efectiva do algoritmo e à optimização dos respectivos parâmetros até que se atinjam os resultados pretendidos [17].

4.1.6 Avaliação de padrões

Contando que o volume de padrões gerados no âmbito de um sistema de extracção de conhecimento pode ascender a grandezas na ordem dos milhões, uma das etapas fundamentais diz respeito à aferição da fracção de padrões que representam efectivamente interesse para o utilizador. Assim, considera-se um conjunto de critérios aos quais um padrão deve obedecer no sentido de ser considerado de interesse e, consequentemente, portador de conhecimento,

como sejam a exigência de ser facilmente compreendido pelo ser humano, a potencial utilidade e o facto de validar a hipótese que um utilizador procura confirmar. Geralmente, este interesse não é quantificado apenas com base em **medidas objectivas**, recorrendo adicionalmente a **medidas subjectivas** (e, de acordo com [20], ainda a **medidas imparciais**) capazes de traduzir e representar as necessidades e objectivos do utilizador, quer através da contradição de uma crença inicial, quer pelo fornecimento de informação estratégica sobre a qual este pode actuar [16, 20].

4.1.7 Apresentação do conhecimento

Por último, a etapa de apresentação diz respeito ao dilema de como utilizar as técnicas de visualização e representação de conhecimento no sentido de comunicar ao utilizador a informação de forma clara e eficiente, podendo ser construída com base em técnicas orientadas aos pixéis, de projecção geométrica e baseadas em ícones, hierarquias ou gráficos [16].

4.2 Regras de Associação

O propósito das regras de associação baseia-se na extracção de correlações de interesse, padrões frequentes, associações or estruturas casuais a partir de um conjunto de dados armazenado numa base transaccional ou outro tipo de repositório. O *mining* mediado por este tipo de estruturas pretende encontrar regras de associação que satisfaçam o apoio e a confiança mínimos de uma dada base de dados, decompondo-se, normalmente, em dois sub-problemas: encontrar os conjuntos de dados cujas ocorrências excedam um limiar pré-definido na base de dados e gerar regras de associação a partir desses conjuntos, obedecendo às restrições de confiança mínima. Considerando $L_k = \{I_1, I_2, ..., I_k\}$ um conjunto definido nos parâmetros anteriores, as regras de associação são geradas tendo como primeira regra $\{I_1, I_2, ..., I_{-1}\} \Rightarrow I_k$, cujo interesse (ou não) é indicado pela confiança que lhe está associada. De seguida, outras regras são geradas apagando os últimos elementos do antecendente e inserindo-os no consequente, sendo igualmente consultadas as respectivas medidas de confiança, num processo iterativo que se prolonga até que o antecedente se encontre vazio.

Em muitos casos, as abordagens algorítmicas dão origem a um número avultado de regras de associação (também elas consideravelmente extensas), o que dificulta a compreensão e validação das mesmas por parte do utilizador e limita, em consequência, a utilidade dos resultados da etapa de *mining*. A eficiência destes algoritmos pode, porém, ser melhorada através de métodos como a amostragem da base de dados, a adição de restrições adicionais à estrutura dos padrões e a paralelização [21].

4.3 Classificação

Na sua essência, a classificação consiste numa forma supervisionada de análise de dados cujo objectivo reside na extracção de modelos capazes de descrever classes de dados relevantes. O processo ao longo do qual se desenvolve é composto de uma primeira fase de **aprendizagem** (ou treino), durante a qual se constrói o modelo com base na análise de conjuntos pré-determinados de dados por parte de um algoritmo de classificação, e uma segunda de **classificação** propriamente dita, em que se utiliza o modelo construído para rotular os dados em análise [16]. De entre os vários métodos disponíveis, destacam-se, nesta secção, a indução de árvores de decisão, a classificação Bayesiana, a classificação baseada nos k-vizinhos mais próximos e, por último, o raciocínio baseado em casos.

4.3.1 Indução de Árvores de Decisão

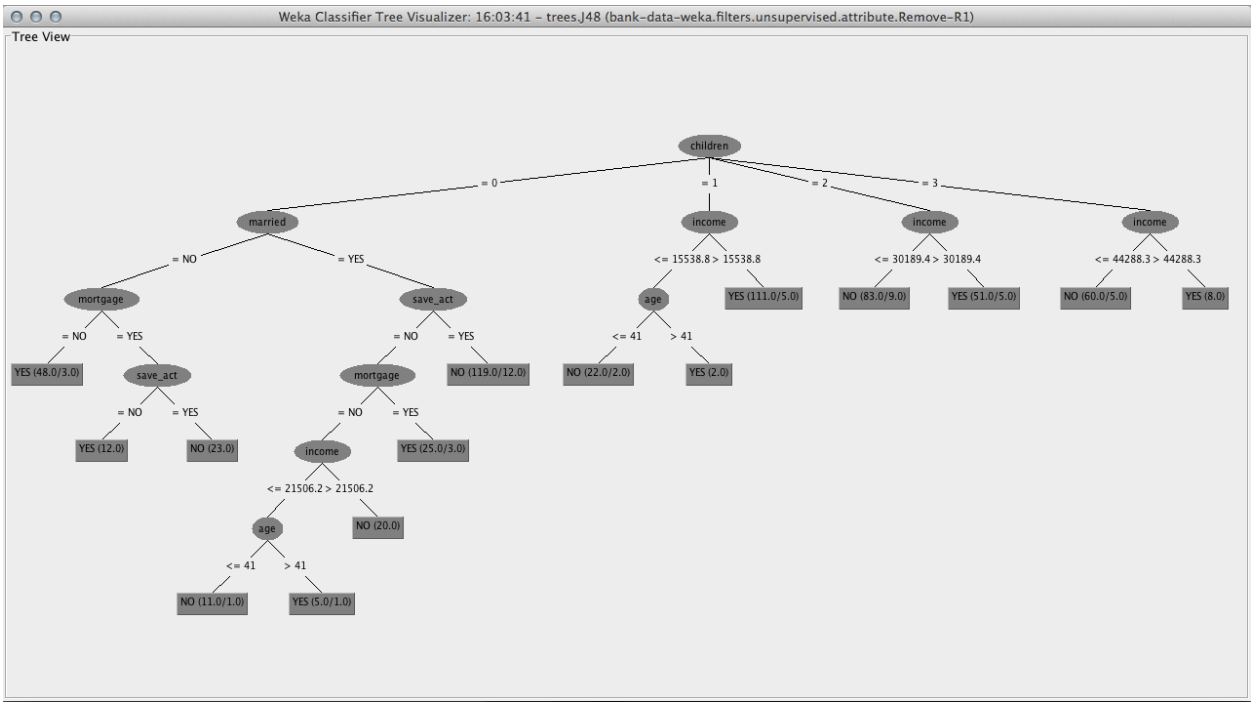

Figura 13: Árvore de decisão gerada no *software* WEKA, a partir do ficheiro *bank-data.arff*, utilizando o classificador J48.

Embora se baseie num método relativamente simples, a classificação utilizando árvores de decisão (Figura 13) é amplamente utilizada na extracção de conhecimento, na medida em que não requer qualquer conhecimento acerca do domínio ou definição de parâmetros e representa o conhecimento adquirido de uma forma intuitiva [16]. Os nodos de uma árvore de decisão envolvem o teste de um atributo em particular, no qual se estabelece uma comparação entre o

valor desse atributo e uma constante, gerando-se, no caso de atributos nominais, um número de sub-nodos correspondente ao número de valores possíveis do atributo; e determinando-se, no caso de atributos numéricos, se o valor do atributo é maior ou menor do que uma constante pré-determinada, procedendo-se a uma divisão do nodo em dois ou três sub-nodos, consoante o tratamento dado aos valores em falta ou a abordagem considerada mais pertinente. Por seu lado, as folhas dizem respeito à definição da classificação que é aplicada a todas as instâncias que atingem este nível, podendo ainda corresponder a um conjunto de classificações ou a uma distribuição de probabilidade sobre todas as classificações possíveis. Considerando uma instância desconhecida, a classificação processa-se abordando a estrutura da árvore no sentido *top-down*, obedecendo aos valores ou atributos testados em nodos sucessivos até que se atinja uma folha, ponto em que a instância é classificada [19]. Há, porém, que ter em consideração que, aquando da construção de uma árvore de decisão, muitos dos ramos reflectem anomalias derivadas da presença de ruído ou *outliers* nos dados, pelo que devem empregar-se métodos de *pruning* de forma a corrigir este problema [16].

4.3.2 Classificação Bayesiana

As redes *Bayesianas* referem-se a um formalismo de representação que associa a estatística e a inteligência artificial utilizado, no contexto de *mining*, para descobrir modelos de interacção complexos em bases de dados volumosas, podendo ser utilizadas em tarefas como, por exemplo, a análise de dados provenientes de inquéritos e a caracterização de perfis de cliente. Uma rede Bayesiana compreende duas componentes: um grafo directo acíclico, cujos nodos e arcos representam variáveis estocásticas e dependências directas entre variáveis, respectivamente; e uma distribuição de probabilidade, que quantifica essas mesmas dependências. Considere-se um cenário em que duas variáveis (A e B) controlam o valor de uma terceira (C) e assumem dois estados (*true* ou *false*) (Figura 14). Por observação da figura, depreende-se que os arcos que apontam para C traduzem a acção conjunta de A e B, e que a ausência de qualquer arco entre estas duas variáveis descreve a independência marginal existente entre ambas. A rede *Bayesiana* assim definida permite decompor a distribuição de probabilidade conjunta das três variáveis em três distribuições distribuições de probabilidade — uma distribuição condicional para a variável C dados os nodos ascendentes, e duas distribuições marginais para A e B. O facto de a rede sumarizar as dependências significativas entre as variáveis que a compõem sem qualquer desintegração de informação indica que este tipo de redes podem funcionar na qualidade de sistemas de simulação capazes de prever o valor de variáveis desconhecidas sob condições hipotéticas e, da mesma forma, encontrar o conjunto de condições iniciais mais provável que levam a uma determinada observação. A derivação de uma rede *Bayesiana* a partir de um conjunto de dados pressupõe, então, a indução tanto

da estrutura gráfica das dependências condicionais como das distribuições condicionais que a quantificam.

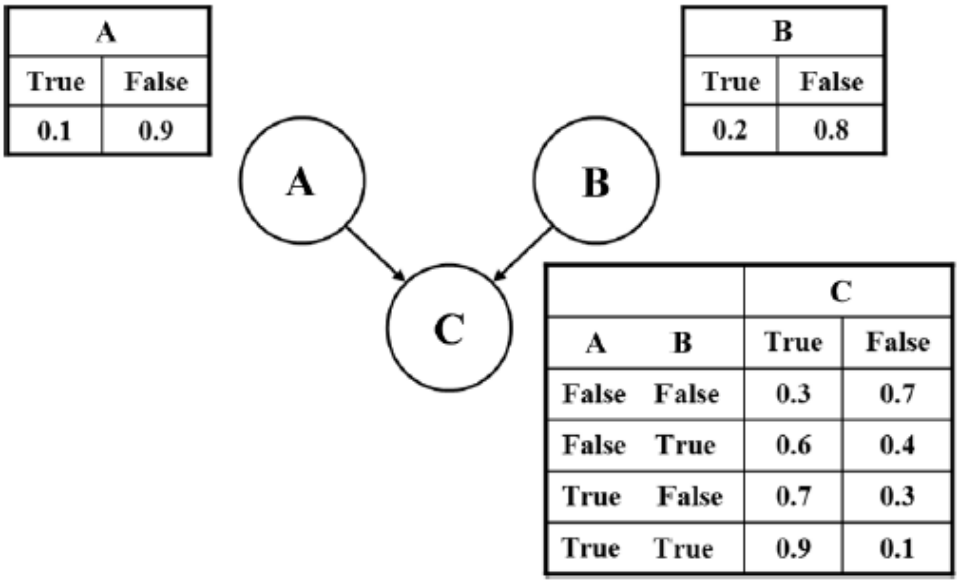

Figura 14: Rede representativa do impacto de duas variáveis (nodos A e B) numa terceira (nodo C), associada às respectivas tabelas descritivas de distribuição condicional.

No contexto específico da extracção de conhecimento, a classificação *Bayesiana* consiste tipicamente em treinar, numa primeira fase, um classificador com um conjunto de treino e utilizá-lo, posteriormente, para classificar um dado conjunto de teste, considerando uma componente de supervisão (ou sinal de treino) que providencia ao classificador um método de avaliação da medida de dependência entre os atributos e as classes. A classificação de um caso com atributos de valor $y_{1k}, ..., y_{vk}$ é, depois, efectuada através do cálculo da distribuição de probabilidade $p(C|y_{1k}, ..., y_{vk})$ da variável da classe, dados os valores dos atributos, sendo o caso rotulado sob a correspondência de maior probabilidade. Dado que a grande maioria das redes *Bayesianas* impõem uma restrição à topologia da rede, torna-se necessário factorizar a probabilidade conjunta $p(y_{1k}, ..., y_{vk,ck})$ da classe e dos atributos como $p(c_k)p(y_{1k,...,y_vk|c_k})$. O exemplo mais simples deste caso é o classificador *Naïve Bayes*, que assume a total independência entre atributos [16, 22].

4.3.3 K-Nearest Neighbors

Os métodos de classificação discutidos anteriormente inserem-se no domínio dos *eager learners*, uma vez que constroem um modelo de generalização antes de receberem novas instâncias para classificar, estando assim prontos e ávidos para classificar novas instâncias

desconhecidas. No domínio oposto, surgem os *lazy learners*, no qual se insere a abordagem do vizinho mais próximo, em que o classificador espera até ao último instante para construir o modelo de classificação. Embora este tipo de abordagens representem um gasto significativo de recursos computacionais, permitem modelar espaços de decisão complexos com formas hiperpoligonais que seriam bastante mais difíceis de descrever recorrendo a *eager learners*.

Os classificadores que recorrem à noção de vizinho mais próximo baseiam-se na aprendizagem por analogia, ou seja, na comparação de instâncias de teste com instâncias de treino que lhe sejam similares. As instâncias de treino encontram-se armazenadas num espaço n-dimensional de padrões, que, aquando da apresentação de uma instância desconhecida, vai ser perscrutado em busca das k instâncias que lhe são mais próximas — os k vizinhos mais próximos. A proximidade é definida em termos de uma métrica de distância, normalmente euclideana, e, de forma a prevenir que determinados atributos se sobreponham a outros de menor extensão, é aplicado um processo de normalização antes do seu cálculo. Por fim, a classificação é feita com base na classe mais comum entre os k vizinhos devolvidos [16].

4.3.4 Raciocínio Baseado em Casos

Ainda no âmbito dos *lazy learners* surgem os classificadores regidos pelo raciocínio baseado em casos, que recorrem a uma base de dados de soluções para resolver novas instâncias de problemas e armazenam cada caso como uma descrição simbólica complexa. Facilmente se depreende que este tipo de classificadores podem ser utilizados na área clínica para auxiliar o tratamento e diagnóstico de pacientes, considerando a existência de um historial de casos e tratamentos.

Relativamente à forma como este método se desenvolve, sempre que seja necessário classificar um novo caso, o *reasoner* verifica, numa primeira fase, se existe na base de dados algum caso semelhante e, em caso afirmativo, a solução associada a esse caso é retornada. Pelo contrário, caso não exista qualquer caso idêntico, inicia-se uma procura baseada em casos que, embora não sejam completamente coincidentes com o procurado, contêm componentes que são similares, e o *reasoner* tenta combinar as soluções dos casos vizinhos numa proposta de solução, o que representa um dos desafios deste tipo de classificação, uma vez que se torna necessário encontrar métodos apropriados que sustentem o processo de combinação. Um outro desafio relaciona-se com o facto de, atingido um determinado volume de casos armazenados, o desempenho do classificador sofrer um decréscimo e, consequentemente, o tempo de procura e processamento de novos casos aumentar de forma significativa, sendo necessário ter em atenção o progresso da base de dados e editá-la, quando necessário [16].

4.4 Segmentação

Contrariamente à classificação, a segmentação (ou *clustering*) aplica-se em situações em que não se considera a noção de classe, mas antes de agrupamentos naturais, no interior dos quais os elementos apresentam um elevado grau de similaridade entre si e, simultaneamente, uma dissimilaridade considerável com os elementos de outros agrupamentos [16, 19]. No domínio específico do *data mining*, a segmentação deve obedecer a determinados requisitos, nomeadamente no que diz respeito à questão da escalabilidade, uma vez que a grande maioria dos algoritmos está optimizado para conjuntos de dados pouco volumosos. Outras questões prendem-se, por exemplo, com a capacidade de lidar com atributos não numéricos e mesmo com dados mais complexos, como imagens e gráficos, com o alargamento desejável da descoberta a formas arbitrárias (não esféricas) e com a robustez face ao ruído contido nos dados [16].

4.5 Metodologias

As metodologias para extracção de conhecimento compreendem um conjunto de passos que tem como objectivo auxiliar o desenvolvimento de um sistema de extracção de conhecimento, nomeadamente ao nível da etapa de *data mining*, procurando facilitar a compreensão e implementação da resolução de problemas de negócio. Neste sentido, abordam-se brevemente as metodologias CRISP-DM e SEMMA, bem como o *standard* PMML.

4.5.1 CRISP-DM

A motivação para o desenvolvimento da metodologia CRISP-DM (CRoss-Industry Standard Process dor Data Mining) surgiu pelo consenso de que, face ao interesse crescente e generalizado do mercado de *data mining*, a indústria necessitava de um projecto padronizado. Esta metodologia, que incorpora conhecimento prático e retira o foco somente da tecnologia para partilhá-lo com as necessidades do utilizador, é descrita em termos de um modelo hierárquico de processos constituído por conjuntos de tarefas descritas nos seguintes quatro níveis, organizados de acordo com um gradiente crescente de especificidade [23, 24]:

1. **Fase**

   O primeiro nível de abstracção procura organizar o processo de *mining* num determinado número de fases, cada uma das quais composta por várias ramificações secundárias de tarefas genéricas.

2. **Tarefa genérica**

A este nível, o objectivo tem que ver com a garantia de completude e estabilidade, respeitantes à abrangência da totalidade do processo de *data mining* e de todas as possíveis aplicações do mesmo, e à garantia de validade do modelo face a desenvolvimentos inesperados, respectivamente.

3. **Tarefa especializada**

   Avançando na especificidade, o terceiro nível procura especificar e descrever a forma como as acções das tarefas do nível anterior devem ser executadas em situações específicas de actuação.

4. **Instância de processo**

   O quarto e último nível consiste num registo das acções, decisões e resultados de uma execução efectiva de *data mining*. Uma instância de processo é organizada de acordo com as tarefas definidas nos níveis superiores, mas representa o que acontece numa execução em específico e não apenas o que acontece de um ponto de vista geral.

O ciclo de vida de um projecto de *data mining* (Figura 15), nomeadamente as fases, as tarefas e as relações entre tarefas que compreende, é representado por intermédico de um modelo de referência, no qual se torna impossível identificar todas as relações, já que estas podem existir entre quaisquer tarefas, dependendo dos objectivos, dos antecedentes, dos interesses específicos do utilizador e, acima de tudo, dos dados. Este ciclo consiste em seis fases, cujas saídas determinam a fase (ou o conjunto de tarefas) que deve ser executada de seguida, estando as dependências mais importantes e frequentes assinaladas com setas, na Figura 15. A fase inicial, denominada de **estudo do negócio**, centra-se na análise dos objectivos e requisitos do projecto, segundo a perspectiva do negócio, e na posterior conversão deste conhecimento num enunciado de *data mining* e num plano preliminar construído de forma a atingir os objectivos delineados. Segue-se o **estudo**

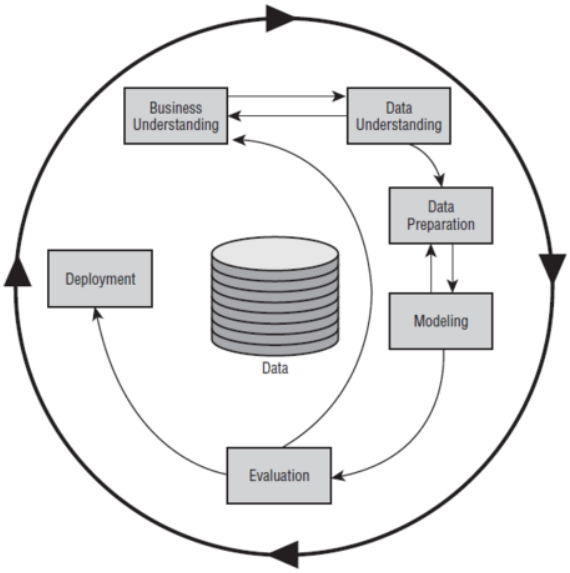

Figura 15: Fases do modelo de referência da metodologia CRISP-DM.

do negócio, que parte de uma colecção de dados inicial e prossegue com uma sequência de actividades que permitem ao utilizador familiarizar-se com esta, identificar possíveis problemas de qualidades, retirar as primeiras ilações e detectar subconjuntos que motivem a formulação de uma hipótese relativamente à informação oculta. Posteriormente, desenvolve-se a fase de **preparação dos dados**, que abrange todas as actividades necessárias à construção do conjunto de dados final — aquele que será disponibilizado à ferramenta de modelação, mais tarde — a partir da colecção inicial, sofrendo inevitavelmente várias optimizações. Estas actividades incluem a selecção de tabelas, registos e atributos, bem como a transformação e limpeza dos dados. A fase seguinte, de **modelação**, selecciona e aplica diversas técnicas de modelação, calibrando ainda os respectivos parâmetros para valores óptimos. Dado que algumas técnicas estabelecem requisitos específicos para a forma como os dados são apresentados e existem, tipicamente, várias técnicas para o mesmo tipo de problemas, pode ser necessário, aqui, regressar à fase inicial de preparação. Após esta fase, dá-se a submissão dos dados e corporiza-se o modelo que, embora aparente apresentar um elevado grau de qualidade, deve ser submetido a uma **avaliação** e revisão dos passos que levaram à sua criação, de forma a atestar se realmente cumpre os objectivos de negócio. Por último, atinge-se a fase de **implementação**, que pode, dependendo dos requisitos, ser tão simples quanto gerar um relatório ou tão complexa quanto implementar todo o processo de *data mining* [23, 24].

4.5.2 SEMMA

A metodologia SEMMA (Sample, Explore, Modify, Model, Assess), cuja designação deriva da noção de divisão do processo de *data mining* do SAS Institute, considera, por sua vez, cinco etapas distintas e é dotada de uma capacidade de ajustamento e actualização face ao surgimento de nova informação. Assim, existe, inicialmente, uma etapa de **amostragem**, que envolve a extracção de uma porção de um conjunto de dados suficientemente grande para conter informação relevante, mas suficientemente pequena para permitir uma manipulação simples e rápida. O facto de se procurar basear o processo de *data mining* numa amostra representativa reduz significativamente o volume e tempo de processamento necessários à extracção da informação essencial para o negócio. As tendências ou agrupamentos inerentes aos dados são, depois, exploradas visual e numericamente, na fase de **exploração**, de forma a refinar o processo de procura de anomalias e tendências imprevistas, procurando-se ainda compreender os dados de forma aprofundada, bem como as possíveis relações que lhes são subjacentes. Com base nos resultados da exploração, procede-se à **modificação** dos mesmos através da criação, selecção e transformação das variáveis para o processo de construção do modelo, que se segue na etapa de **modelação**. A definição das técnicas de construção de modelos de *data mining* — que incluem técnicas de aprendizagem automática e modelos

estatísticos — deve suportar-se do pressuposto de que cada modelo possui propriedades e características singulares dependentes dos dados e adequados a situações específicas. Por último, a etapa de **avaliação** pressupõe a aplicação do modelo à amostra de dados e a subsequente verificação do respectivo desempenho, de forma a efectuar ajustes, caso se revele necessário [24].

4.5.3 PMML

No domínio da representação e partilha de soluções analíticas de previsão entre aplicações, o PMML (Predictive Model Markup Language) corresponde ao modelo padrão *de facto*, actualmente, permitindo aos utilizadores construir, visualizar e implementar com extrema facilidade as soluções pretendidas utilizando diferentes plataformas e sistemas, com base num *schema* que especifica a forma como os elementos de linguagem contidos num ficheiro PMML devem ser utilizados. A estrutura utilizada pelo PMML para descrever um modelo de *data mining* é bastante intuitiva, sendo composta por vários elementos que encapsulam diferentes funcionalidades consoante os dados, o modelo e os resultados considerados. Sequencialmente, esta estrutura pode ser descrita através dos seguintes elementos [25]:

1. **Cabeçalho**

   O elemento de cabeçalho contém informação genérica acerca do documento PMML, como seja informação de *copyright* para o modelo, a respectiva descrição e informação acerca da aplicação utilizada para gerar o modelo, apresentando ainda um atributo para um *timestamp* que pode ser empregue para especificar a data de criação do modelo.

2. **Dicionário de dados**

   As definições dos campos passíveis de serem utilizados por um modelo encontram-se definidas no dicionário de dados, e é neste elemento que um campo é definido como contínuo, categórico ou ordinal. Com base nesta definição, os intervalos apropriados de valores são definidos, assim como o tipo de dados. O dicionário é ainda utilizado para enumerar a lista de valores válidos, inválidos e em falta para cada um dos conjuntos de entrada.

3. **Transformações de dados**

   As transformações permitem o mapeamento dos dados do utilizador numa forma mais adequada aos modelos de *mining*, podendo este processo recorrer a diversas técnicas, como sejam a **normalização**, a **discretização**, o **mapeamento de valores**, a aplicação de **funções** a um ou mais parâmetros e a **agregação**. A capacidade

de representação das transformações de dados, bem como de métodos de tratamento de valores omissos e *outliers*, constituem, conjuntamente com os parâmetros que definem os modelos, um dos conceitos-chave do PMML.

4. **Modelo**

   O último elemento diz respeito à definição do modelo de *data mining*, cuja representação se inicia com um **mining schema** e termina com a efectiva representação do modelo. O *schema* refere uma lista de todos os campos integrantes do modelo, compreendendo adicionalmente informação específica acerca de cada um deles, e especifica os procedimentos de tratamento dos valores especiais supracitados. Os restantes elementos — **alvos** e **especificidades do modelo** — fazem referência, respectivamente, à permissão de ajustamento das variáveis previstas e à especificação dos detalhes do modelo em si, assim que o *schema* esteja configurado.

4.6 Exemplo de Aplicação Biomédica

Considerando a quantidade de conhecimento e dados armazenados em bases de dados médicas, torna-se imperativo o desenvolvimento de ferramentas especializadas que permitam não só aceder aos dados, mas também proceder à sua análise, à descoberta de conhecimento e à efectiva utilização dos recursos disponíveis, uma vez que a dificuldade de extrair informação útil acompanha o aumento do volume de dados. Neste contexto específico, o processo de descoberta de conhecimento em bases de dados (e, por associação, o *data mining*), assume um papel fulcral na gestão do rápido crescimento e globalização da informação, pese embora alguns ambientes pequem ainda pela insuficiente standardização, vendo a sua aplicação consequentemente limitada a conjuntos de dados recolhidos para diagnósticos específicos, prognósticos, monitorização, suporte à terapia ou para outros propósitos relacionados com a gestão do paciente. Um dos factores que contribuiu significativamente para o desenvolvimento desta dimensão no seio da actividade médica e hospitalar diz respeito à incompletude, incorrecção e carácter esparso associadas à recolha de, por exemplo, dados provenientes de registos de pacientes e investigações específicas, que implicam necessariamente a existência de técnicas capazes de lidar com estas características. Adicionalmente, conjuntos de dados provenientes de rotinas de monitorização — quer aguda, em unidades de cuidados intensivos, quer discreta e prolongada, no caso de pacientes com condições crónicas — acrescem ao factor anteriormente exposto a aquisição de medições de um conjunto de parâmetros em instantes distintos, obrigando a que a componente temporal seja imbuída no processo de análise [26].

5 Conclusão e Discussão

Apresenta-se, nesta última secção, uma breve análise das ilações retiradas ao longo da elaboração do trabalho para cada uma das áreas de conhecimento abordadas: Redes Neuronais Artificiais, Algoritmos Genéticos e Extracção de Conhecimento.

5.1 Redes Neuronais Artificiais

Uma das grandes vantagens que advém da utilização de redes neuronais provém da sua capacidade de aprender com o ambiente, o que se revela de grande utilidade, por exemplo, em aplicações em que a complexidade dos dados ou tarefas envolvidos inviabilizam a implementação mediada por outro tipo de abordagens. Assim, estas estruturas podem ser aplicadas em áreas tão diversas como a classificação, o processamento de dados, a robótica e o apoio à decisão.

A escolha da topologia de rede a utilizar depende directamente da especificidade e da representação dos dados do problema em causa, sendo que este processo deve ser suportado por um conhecimento teórico dos modelos e dos algoritmos de aprendizagem existentes. Um dos factores mais determinantes nesta escolha diz respeito à complexidade, uma vez que uma decisão desajustada relativamente a esta característica pode levar a resultados pobres ou mesmo errados, bem como a comportamento indesejados, como seja a memorização. Da mesma forma, o paradigma a imbuir no sistema deve ser seleccionado de acordo com os requisitos particulares do problema apresentado, sendo que a aprendizagem supervisionada se encontra intimamente relacionada com tarefas de classificação, enquanto que a não supervisionada é sobretudo empregue em aplicações que se insiram no domínio dos problemas de aproximação (como a modelação estatística, a compressão, a filtragem e o *clustering*). É possível ainda considerar o paradigma de reforço, que é, neste trabalho, definido enquanto variante da aprendizagem supervisionada, que se preocupa essencialmente com a forma como a rede deve agir sobre um determinado ambiente de forma a maximizar a noção de recompensa a longo prazo e é particularmente indicado para problemas que incluam conflitos entre recompensas de longo e curto prazo, estando a sua área de abrangência sobretudo ligada ao controlo robótico, às telecomunicações e a tarefas gerais que envolvam decisões sequenciais. Independentemente do paradigma utilizado, o objectivo da aprendizagem centra-se sobre a definição dos valores dos pesos com base nos dados fornecidos à rede no sentido de encontrar uma solução, caso exista.

A utilização de redes neuronais tem vindo a consolidar-se enquanto nova tendência no domínio da estatística médica, apresentando já um impacto considerável em áreas específicas como a citologia cervical, a mamografia e a detecção precoce de enfartes agudos do miocárdio.

Estes modelos são especialmente apelativos devido à capacidade de reprodução simultânea da interacção dinâmica entre múltiplos factores, o que permite elaborar estudos de complexidade e se revela fulcral na aquisição de conhecimento aprofundado acerca de determinadas patologias ou no aumento da compreensão das possíveis implicações inerentes a associações anormais entre variáveis.

5.2 Algoritmos Genéticos

Os algoritmos genéticos seguem, à semelhança das redes neuronais, uma inspiração biológica que advoga o estudo dos sistemas naturais em prol do desenvolvimento e maturação de métodos computacionais capazes de suprir e optimizar necessidades como a adaptabilidade, a paralelização massiva, a aprendizagem e a manipulação positiva de elevados níveis de complexidade. Numa perspectiva curiosamente inversa, estes algoritmos podem também ser empregues no sentido de melhorar o conhecimento acerca dos próprios sistemas naturais, ao permitirem a realização de experiências que não seriam possíveis na esfera quotidiana e a simulação de fenómenos difíceis, ou mesmo impossíveis, de captar e analisar recorrendo a ferramentas puramente analíticas. Porém, e assim como acontece com qualquer sistema de aprendizagem, a utilização de algoritmos genéticos deve reger-se por uma gestão consciente do universo do problema considerado, sendo que nem sempre esta abordagem é a mais adequada. Nesta linha de raciocínio, deve evitar-se a utilização destes métodos sempre que se anteveja claramente que é possível encontrar uma solução analítica ou circunscrita do problema, em quaisquer situações que requeiram, à partida, repetibilidade ou soluções em tempo real, e em casos particulares que sugiram explicitamente que existem outros paradigmas mais adequados, como sejam as redes neuronais artificiais. Por outro lado, esta estratégia evolucionária deve ser tida em especial consideração em cenários que envolvam um espaço de soluções possíveis muito grande e altamente dimensional, e sempre que o problema considerado contenha restrições e não-linearidades.

5.3 Extracção de Conhecimento

Inevitavelmente, os sistemas de extracção de conhecimento — com ênfase na etapa de *data mining* — apresentam uma forte relação de proximidade com as questões debatidas no contexto da aprendizagem, uma vez que exploram técnicas de ambos os domínios específicos abordados no sentido de desenvolver e optimizar os métodos que lhes são subjacentes. Em particular, o *data mining* assemelha-se de forma significativa à aprendizagem com redes neuronais, alargando, porém, o foco da procura para além da precisão, até à dimensão da eficiência e da escalabilidade dos métodos utilizados. Estes dois aspectos constituem, aliás,

um dos maiores desafios no âmbitos da extracção de conhecimento, já que os grandes volumes de informação com que lidam requerem necessariamente a definição de estratégias que sejam não só eficazes e eficientes, mas também de dimensão controlável e, até uma certa extensão, de implementação previsível. Logicamente, o desenvolvimento dos sistemas de extracção de conhecimento deve ainda suportar-se de algoritmos paralelos e distribuídos capazes de optimizar todo o processo de procura de padrões. Actualmente, as redes neuronais artificias são consideradas ferramentas padrão na extracção de conhecimento, revelando-se de extrema utilidade em tarefas como o reconhecimento de padrões, a análise de séries de tempo, a previsão e o *clustering*, e desempenhando mesmo o papel de módulo nuclear em determinados pacotes de *softwar* comerciais de *data mining*. Esta popularidade deve-se, em parte, ao facto de o processo de modelação ser extremamente adaptativo e amplamente determinado pelas características dos padrões apreendidos (e aprendidos) a partir dos dados, no processo de aprendizagem, o que se adapta de forma ideal às necessidades dos problemas de extracção de conhecimento, contando que, embora o volume de dados envolvido seja extremamente elevado, a quantidade de padrões relevantes não é, à partida, conhecida ou passível de especificação. Uma outra vantagem das redes neuronais, neste contexto, tem que ver com a sua capacidade de aproximar ou representar relações complexas, o que dota estas estruturas de flexibilidade na modelação dos processos inerentes à génese de dados e na descoberta de relações desconhecidas entre a totalidade de dados de um dado conjunto. Por último, a tolerância à falha que ostentam potencia ainda o processamento dos referidos conjunto de *dirty data*, uma vez que, contrariamente aos modelos estatísticos, as redes neuronais não requerem uma estrutura probablística bem definida, sendo capazes de lidar com padrões imprecisos ou dados que compreendam informação incompleta ou afectada por ruído. Naturalmente, a aplicação de modelos específicos de redes neuronais depende do tipo de problema a resolver, sendo, por exemplo, as redes *feedforward* multicamada mais apropriadas para tarefas de classificação e as redes de Kohonen e Self-Organizing Maps particularmente úteis no processo de *clustering*, entre outras aplicações. Por sua vez, a aplicação de abordagens evolucionárias na extracção de conhecimento centra-se sobretudo nos mecanismos de *clustering*, uma vez que este processo pode ser encarado como um problema de optimização que parte de uma população de estruturas de *clustering* — codificadas como cromossomas — e emprega operadores genéticos para forçar a sua convergência num *cluster* globalmente óptimo. Neste contexto específico, a função de avaliação mais apropriada consiste na inversa do erro quadrático, uma vez que *clusters* com um erro quadrático reduzido apresentam um maior valor de aptidão e, consequentemente, constituem as melhores soluções para o problema. Contudo, uma das limitações das abordagens baseadas em algoritmos genéticos para efeitos de *clustering* está relacionada com a sensibilidade inerente à selecção dos parâmetros

de controlo, sendo necessário adaptá-las de forma a incluir heurísticas específicas para os problemas considerados.

De um ponto de vista empírico, há que considerar, na esfera da extracção de conhecimento, que, face a um dado problema, a escolha de abordagem circunscreve de forma significativa o volume de informação útil que é possível retirar do conjunto de dados envolvido, pelo que se torna essencial, para além da capacidade de resolver um problema específico, evoluir os sistemas no sentido de quantificá-lo e aumentar a sua compreensão. Assim, um outro desafio consiste na adaptação do desempenho deste tipo de sistemas à especificidade dos domínios em que são utilizados e na consequente eliminação (ou minoração) do dilema de escolha de métodos a empregar face aos factores considerados em cada caso. O desempenho dos sistemas de extracção de conhecimento não se encontra, contudo, apenas dependente do seu esqueleto interno, desempenhando as características qualitativas dos dados que lhes são fornecidos — grande parte dos quais, geralmente, consiste nos designados conjuntos de *dirty data* — um papel determinante. Conforme explorado, a solução lógica para contornar esta situação prende-se com a enfatização da etapa de pré-processamento, nomeadamente da limpeza dos dados e, acima de tudo, da persistência deste processo, tendo em vista o objectivo último de assegurar padrões de qualidade contínuos.

6 Referências

[1] R. Rojas. The biological paradigm. In *Neural Networks - A Systematic Introduction*. Springer-Verlang, 1996.

[2] S. Haykin. Introduction. In *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, 1998.

[3] A. Jain, J. Mao, and K. Mohiuddin. Artificial neural networks: A tutorial. *IEEE Computer*, 29:31–44, 1996.

[4] R. Rojas. Perceptron learning. In *Neural Networks - A Systematic Introduction*. Springer-Verlang, 1996.

[5] J. Bešter A. Krenker and A. Kos. Introduction to the artificial neural networks. In K. Suzuki, editor, *Artificial Neural Networks — Methodological Advances and Biomedical Applications*. InTech, 2011.

[6] P. Cortez and J. Neves. Redes Neuronais Artificiais. *Apontamentos de apoio à disciplina de Sistemas Inteligentes*, Unidade de Ensino, Departamento de Informática, Escola de Engenharia, Braga, Portugal, 2000.

[7] Nils J. Nilsson. Introduction to machine learning: An early draft of a proposed textbook., 1996.

[8] Donald O. Hebb. *The Organization of Behavior*. John Wiley, New York, 1949.

[9] S. Roweis. Boltzmann Machines. *Lecture Notes*, Courant Institute of Mathematical Sciences New York University, 1995.

[10] A. Dehnavi H. Rabbani and M. Ghanatbari. Estimation the depth of anesthesia by the use of artificial neural network. In K. Suzuki, editor, *Artificial Neural Networks — Methodological Advances and Biomedical Applications*. InTech, 2011.

[11] G. Kendall K. Sastry, D. Goldberg. Genetic algorithms. In E. Burke and G. Kendall, editors, *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer, 2005.

[12] M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1998.

[13] R. Sivaraj and T. Ravichandran. A review of selection methods in genetic algorithm. *International Journal of Engineering Science and Technology*, 3(5), 2011.

[14] Lance D. Chambers. *The Practical Handbook of Genetic Algorithms: Applications*. CRC Press, Inc., 2nd edition, 2000.

[15] Michael Levin. Design and implementation of genetic algorithms for solving problems in the biomedical sciences, 1995.

[16] Jiawei Han, Micheline Kamber, and Jian Pei. Data preprocessing. In *Data Mining — Concepts and Techniques*. Morgan Kaufmann Publishers Inc., 3rd edition, 2011.

[17] O. Maimon and L. Rokach. Introduction to knowledge discovery and data mining. In Oded Maimon and Lior Rokach, editors, *The Data Mining and Knowledge Discovery Handbook*, pages 1–15. 2010.

[18] J. Maletic and A. Marcus. Data cleansing: A prelude to knowledge discovery. In Oded Maimon and Lior Rokach, editors, *The Data Mining and Knowledge Discovery Handbook*, pages 20–32. 2010.

[19] I. Witten, E. Frank, and M. Hall. Input: Concepts, instances and attributes. In *Data Mining: Practical Machine Learning Tools and Techniques*, pages 39–60. Morgan Kaufmann Publishers Inc., 3rd edition, 2005.

[20] S. Sahar. Interestingness measures - on determining what is interesting. In Oded Maimon and Lior Rokach, editors, *The Data Mining and Knowledge Discovery Handbook*, pages 603–612. 2010.

[21] S. Kotsiantis and D. Kanellopoulos. Association rules mining: A recent overview.

[22] P. Sebastiani, M. Abad, and M. Ramoni. Bayesian networks. In Oded Maimon and Lior Rokach, editors, *The Data Mining and Knowledge Discovery Handbook*, pages 175–208. 2010.

[23] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. Crisp-dm 1.0 step-by-step data mining guide. Technical report, The CRISP-DM Consortium, 2000.

[24] M. Santos and C. Azevedo. *Data Mining : Descoberta de Conhecimento em Bases de Dados*. FCA Editores, 2005.

[25] A. Guazzelli, M. Zeller, W. Lin, and G. Williams. PMML: An open standard for sharing models. *The R Journal*, 1(1):60–65, 2009.

[26] N. Lavrac and B. Zupan. Data mining in medicine. In Oded Maimon and Lior Rokach, editors, *The Data Mining and Knowledge Discovery Handbook*, pages 1111–1136. 2010.