

Sistemas de Aprendizagem

António Silva,
Marcelo Queirós Pinto,

¹ Universidade do Minho, Braga, Portugal
A73827@alunos.uminho.pt, mqspinto@gmail.com

Abstract. Machine Learning is a scientific area that explores the study and construction of algorithms that can learn from data. These algorithms use data to make predictions or make decisions. In this work we intend to explore several learning models such as: Artificial Neural Networks, Decision Trees, Case-Based Reasoning and a recently proposed model, based on the use of Case-Based Reasoning and Deep Learning. The conclusions of this review are that each model is good to a given set of problems, bringing several advantages and disadvantages (section 7).

Keywords: Artificial Neural Networks, Case-Based Reasoning, Decision Trees, Learning Systems, Artificial Intelligence.

Resumo. *Machine Learning* ou Aprendizagem Máquina é uma disciplina científica que explora o estudo e construção de algoritmos que podem aprender a partir de dados. Estes algoritmos utilizam os dados para fazer predições ou tomar decisões. Neste trabalho pretendemos explorar vários modelos de aprendizagem como: Redes Neurais Artificiais, Árvores de Decisão, Raciocínio Baseado em Casos e um modelo recentemente proposto, baseado na utilização de raciocínio baseado em casos e *deep learning*. As conclusões desta revisão são que cada modelo se aplica em um determinado conjunto de problemas, trazendo diversas vantagens e desvantagens, como explicado na secção 7.

Palavras-chave: Redes Neurais Artificiais, Raciocínio Baseado em Casos, Árvores de Decisão, Sistemas de Aprendizagem, Inteligência Artificial.

1 Introdução

Machine Learning ou Aprendizagem Máquina é uma disciplina científica que explora o estudo e construção de algoritmos que podem aprender a partir de dados. Estes algoritmos utilizam os dados para fazer predições ou tomar decisões.

Os algoritmos de Machine Learning são normalmente caracterizados em três categorias: Aprendizagem supervisionada, não supervisionada e por reforço. Algoritmos supervisionados requerem informação acerca das saídas desejadas para cada entrada, tendo um determinado dado ou conjunto de dados desejados para cada dado ou conjunto de dados de entrada. Depois de treinados, estes algoritmos

conseguem dar um feedback sobre o desempenho das previsões, utilizando parte dos dados de entrada para testes. Algoritmos não supervisionados são algoritmos que permitem, por exemplo, a partir de um conjunto de padrões, descobrir semelhanças entre eles, ou agrupá-los. Não têm qualquer acesso às saídas desejadas, e assim não têm possibilidade de se treinarem nem fornecerem um feedback com base nelas. Algoritmos que implementam aprendizagem por reforço interagem com um ambiente e descobrem qual é a melhor opção para cada estado, tendo um objetivo bem definido. Conseguem aprender através da tentativa e erro quais as opções mais valiosas e em que situações (estados), implementando um sistema de recompensa/castigo, levando o sistema a escolher pelas opções que lhe trazem maior recompensa.

Neste trabalho pretende-se uma revisão de 3 tipos de modelos de aprendizagem: Redes Neurais Artificiais, Árvores de Decisão e Raciocínio Baseado em Casos. Além destes, pretendemos a revisão de um modelo recentemente proposto, baseado na utilização de raciocínio baseado em casos com *Deep Learning*.

2 Redes Neurais Artificiais

2.1 Descrição

Uma Rede Neuronal Artificial (RNA) é um modelo de processamento de informação inspirado na forma como o sistema nervoso biológico, como o cérebro humano, processa informações [10]. Trata-se de uma estrutura extremamente interconectada de unidades computacionais, frequentemente designadas por neurónios ou nodos, com capacidade de aprendizagem [11]. Um neurónio é a unidade elementar de processamento de informação e tem capacidade de calcular o nível de ativação das entradas e dos pesos (explicado em pormenor na secção 2.2). Assim, a ideia básica de uma RNA é a simulação da estrutura neuronal de organismos inteligentes, a partir de um modelo matemático que permite adquirir conhecimento através da experiência.

Enquanto uma RNA vulgar poderá possuir menos de uma dezena de unidades de processamento (neurónios) e uma grande RNA pode ter centenas ou milhares de unidades de processamento (processo denominado de *Deep Learning*), o cérebro de um mamífero pode ter vários biliões de neurónios. Especificamente, o cérebro humano incorpora cerca de 10 biliões de neurónios e 60 triliões de conexões (sinapses) entre si [14]. O uso simultâneo de múltiplos neurónios permite ao cérebro processamento mais rápido que os computadores mais poderosos existentes.

No cérebro humano, acontecem os seguintes fenómenos:

- os sinais propagam-se entre os neurónios através de reações eletroquímicas complexas;
- as substâncias químicas libertadas nas sinapses provocam alterações no potencial elétrico do corpo da célula;

- quando o potencial atinge o limiar, um pulso elétrico é enviado através do axónio;
- o pulso espalha-se e normalmente chega a sinapse provocando alterações no seu potencial;
- plasticidade, isto é, em resposta a padrões de estímulos os neurónios alteram a força das suas ligações por longos períodos;

Devido à plasticidade, as ligações dos neurónios que levam as respostas afirmativas são fortalecidas enquanto as ligações negativas são enfraquecidas. Estes mecanismos permitem criar o raciocínio básico do cérebro [15].

RNB	RNA
Soma	neurónio
Dendrite	Entrada
Axónio	Saída
Sinapse	Peso

Fig. 1. Correspondência dos componentes cerebrais humanos para as RNA.

As RNA, inspirando-se nos tópicos referidos e simulando os componentes do cérebro humano (Fig. 1.), permitem a aprendizagem, usando a experiência para melhorar o seu desempenho. Assim, quando expostas a um número suficiente de amostras conseguem generalizar o conhecimento adquirido para amostras desconhecidas, com um erro dependente maioritariamente do número de amostras a que tem acesso. Este erro pode ser medido com recurso a coeficientes de regressão (como o RMSE e R^2), comparando os resultados obtidos pela RNA, em fase de treino, com os resultados conhecidos, aplicando aprendizagem supervisionada. Na secção 2.2 explicar-se-á como decorre a fase de treino.

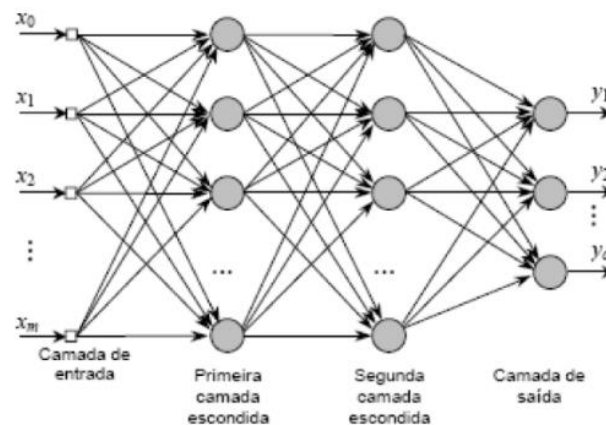


Fig. 2. Arquitetura de uma RNA.

Uma RNA é constituída por uma hierarquia de camadas, estando os neurónios organizados nessas camadas. As camadas de entrada e saída ligam a rede ao ambiente. A arquitetura de uma RNA varia conforme o problema e normalmente não é um

processo fácil saber o número de neurónios e camadas de neurónios que dará melhor desempenho. Este processo exige experiência e teste de várias arquiteturas.

Deep Neural Networks

Deep Neural Networks são RNA's com arquiteturas com múltiplas camadas ocultas para resolver problemas específicos. Enquanto nas RNA's comuns temos aprendizagem supervisionada, em DNN, normalmente não temos qualquer informação sobre as saídas desejadas, de forma a treinar a rede. Uma rede profunda pode, por exemplo, ser treinada com um grande conjunto de imagens que contêm um objeto, ou um padrão, e ao fim de milhares de imagens, ser capaz de, para qualquer imagem, identificar o objeto para a qual foi treinada, caso exista na imagem.

2.2 Capacidade de Aprendizagem

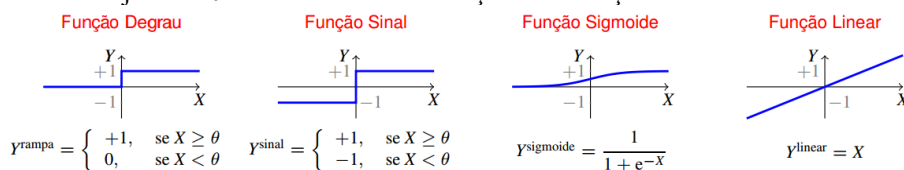
Um neurónio recebe sinais de várias entradas e produz uma saída que se separa em vários ramos que transmitem o mesmo sinal a vários neurónios. Os neurónios são conectados por ligações e cada uma destas ligações tem um peso, representando a força que cada entrada tem no neurónio. A RNA aprende pelo ajustamento repetido dos pesos.

Função de ativação

Quando um neurónio recebe sinais na sua entrada (dado pelo ambiente ou saídas de outros neurónios), este calcula o seu nível de ativação e envia o valor calculado como sinal de saída. Neste processo, o neurónio calcula a soma ponderada dos sinais de entrada:

$$X = \sum_{i=1}^n x_i w_i$$

em que x_i representa o valor de cada sinal de entrada e w_i o peso da ligação. Depois do cálculo é aplicada a função de ativação no valor X , de forma a que saída do neurónio seja entre 0 e 1. Existem várias funções de ativação:



Normalmente, é utilizada a função de ativação sigmoide.

Backpropagation

Existem dezenas de métodos de aprendizagem para RNA, no entanto o mais utilizado é o método Backpropagation, proposto por Bryson e Ho em 1969.

Simplificadamente, este método pode ser descrito em 3 fases:

Fase 1 - Inicialização:

- Atribuir a todos os pesos um número aleatório num intervalo pré-definido;

Fase 2 - Ativação:

- Cálculo das saídas dos neurónios com a função de ativação sigmoide;

Fase 3 - Treino dos pesos:

- Calcular o erro para os neurónios de saída, comparando a sua saída com a saída correta (utilizando as amostras);
- Calcular as correções a efetuar nos pesos;
- Atualizar os pesos dos neurónios de saída;
- Proceder da mesma forma para as camadas anteriores, do fim para o início.
- Voltar à fase 2 até atingir um limite de erro máximo permitido.

Assim, a aprendizagem é conseguida a partir de pequenos ajustes nos pesos com o objetivo de reduzir a diferença entre a saída atual e a saída desejada, isto é, os pesos são atualizados de forma a que os resultados sejam consistentes com os exemplos de treino.

2.3 Ferramentas de Desenvolvimento

Existem várias ferramentas de desenvolvimento existentes, as quais podemos dividir em 2 áreas:

- Ferramentas/bibliotecas que exigem conhecimentos de programação, mas que proporcionam maior personalização do algoritmo, assim como pré-tratamento dos dados, podendo-se referir opencv e encog para a linguagem C++, PyBrain, Theano e opencv para Python, netlab, Jabin wrote e nntool para MATLAB, JNNS para Java, ,
- Ferramentas *user-friendly*, não necessitando de programação ou quase nenhuma, podendo proporcionar a alteração de parâmetros na interface de aplicação como Justnn, Weka e EasyNN Plus.

De seguida apresenta-se a interface para uma RNA no software WEKA, no dataset iris fornecido pelo mesmo software. O WEKA permite ao utilizador especificar a arquitetura da rede e os parâmetros que controlam o treino de uma RNA.

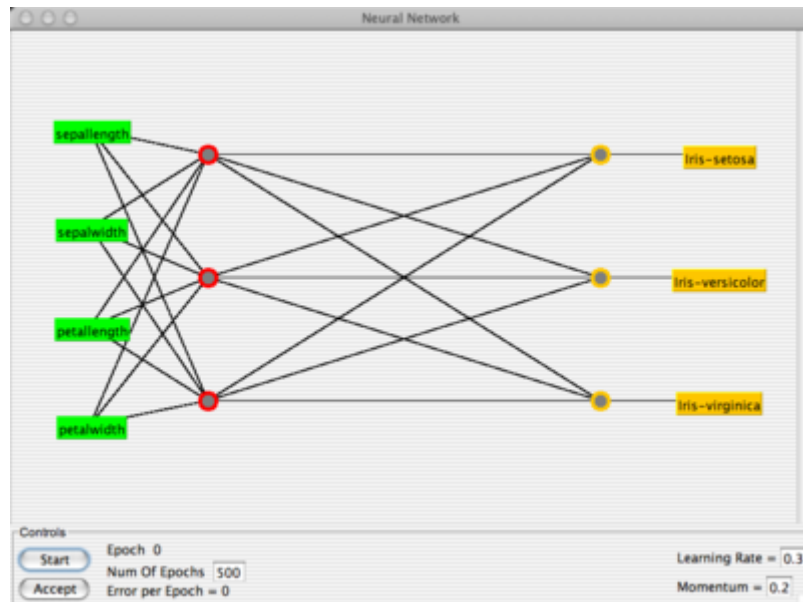


Fig. 3. Interface do software WEKA na utilização de RNA's.

As ferramentas mencionadas destinam-se a diferentes objetivos, sejam eles por exemplo educacionais, investigação ou indústria. Para fins industriais são normalmente adotadas soluções mais eficientes como bibliotecas para C++ (ex: opencv).

2.4 Soluções existentes no mercado

As RNA têm sido usadas para resolver uma grande variedade de tarefas. De seguida destaca-se algumas áreas onde as redes neuronais são uma solução comum:

- Reconhecimento de padrões;
- Reconhecimento de caracteres manuscritos / máquina;
- Reconhecimento de voz;
- Visão computacional;
- Reconhecimento de impressões digitais;
- Segurança em redes informáticas;
- Aprendizagem de hábitos dos utilizadores, e passa dessa forma a reconhecê-los;
- Auxílio no diagnóstico médico.

usadas para resolver uma grande variedade de tarefas. De seguida destaca-se algumas áreas onde as redes neuronais.

Quanto a soluções específicas, existem inúmeras. Assim destacam-se algumas em 2017:

Determination of sugar content in whole Port Wine grape berries combining hyperspectral imaging with neural networks methodologies

Autores:

Gomes V., Fernandes A, Lopes P., Pereira L., Faia A.Pinto P.

Descrição:

Trata-se da utilização de imagem hiperespectral em uvas para a avaliação do seu conteúdo de açúcar para posterior utilização na vinicultura. Atualmente, os vinhos são feitos na indústria com recurso a análises laboratoriais, são recolhidas algumas amostras e o açúcar é medido em laboratório de forma a poder-se fazer um vinho padronizado, recolhendo uvas com os níveis de açúcar ideais. Esta metodologia visa a utilização de uma simples câmara hiperespectral para fotografar uma uva e posterior indicação do nível de açúcar numa interface gráfica (ex: smartphone), em cerca de 1 segundo. Para isto, a RNA é treinada com imagens espectrais de algumas centenas de amostras de uva e são fornecidos os resultados laboratoriais dessas amostras (aprendizagem supervisionada).

A Deep Learning Based Artificial Neural Network Approach for Intrusion Detection

Autores:

Roy S, Mallik A., Gulati R., Obaidat M, Krishna P.

Descrição:

Trata-se da aplicação de redes neuronais (neste caso profundas) em antivírus, mais especificamente no sistema de intrusão (IDS), utilizando vários parâmetros os hábitos do utilizador para poder prever que tipo de alertas devem ser considerados um possível ataque;

A deep learning approach for the analysis of masses in mammograms with minimal user intervention

Autores:

Dhungel N., Carneiro G., Bradley A.

Descrição:

Metodologia integrada para deteção, segmentação e classificação de massas mamárias de mamografias com intervenção mínima do médico. A RNA utiliza parâmetros de treino como forma, tamanho aparência e localização do tumor.

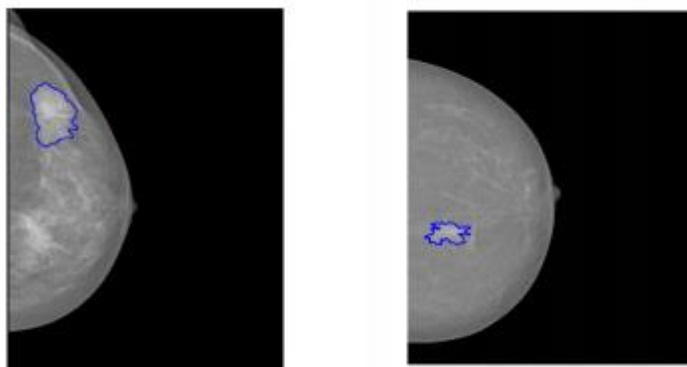


Fig. 4. Massa maligna vs massa benigna [19].

Outros:

- Recentemente a Google mostrou ser capaz de retirar marcas de água de qualquer fotografia, treinando o seu algoritmo de Deep RNA's com várias imagens com o mesma marca de água.
- No dia 4 de Outubro de 2017, a Huawei apresentou o seu mais recente Smartphone, Mate 10 pro, que vem equipado com uma NPU (Neural Network Processing Unit), a primeira no mundo, visto que este smartphone utiliza vários algoritmos com redes neuronais.

3 Árvores de Decisão

3.1 Descrição

As árvores de decisão são uma ferramenta para tomada de decisões, cujo algoritmo e resultados podem ser visualizados através de um grafo, em que as suas arestas correspondem às divergências resultantes das tomadas de decisão.

A sua área de aplicação não se limita a nenhuma área em particular, no entanto o tipo de problemas que resolvem envolvem classificação.

Exemplos de tarefas de classificação poderão ser diagnósticos médicos a partir de sintomas, determinar o valor teórico de uma jogada num jogo de xadrez ou deduzir as condições atmosféricas num dado dia a partir de observações.

Embora à primeira vista possa parecer um espectro limitado de aplicações úteis, a maior parte das tarefas, por muito complexas que sejam, podem ser transformadas em problemas de classificação, dividindo a tarefa em várias tomadas de decisão. Esta última característica é particularmente importante, visto que torna possível a representação de problemas complexos em diagramas de fácil leitura por parte de seres humanos.

Modo de funcionamento

Iremos abordar o algoritmo de geração de árvores de decisão mais utilizado no mercado: ID3 (*Iterative Dichotomiser 3*) proposto por JR Quinlan [5]. Este algoritmo, devido à sua natureza *greedy*, não garante que a árvore gerada seja a melhor mas consegue, a partir de conjuntos de dados de grande volume, gerar uma boa árvore sem requerer muito esforço computacional.

Na figura em baixo, encontram-se os dados de treino a partir dos quais se irá gerar uma árvore de decisão:

No.	Attributes			Class
	Outlook	Humidity	Windy	
1	sunny	high	false	N
2	sunny	high	true	N
3	overcast	high	false	P
4	rain	high	false	P
5	rain	normal	false	P
6	rain	normal	true	N
7	overcast	normal	true	P
8	sunny	high	false	N
9	sunny	normal	false	P
10	rain	normal	false	P
11	sunny	normal	true	P
12	overcast	high	true	P
13	overcast	normal	false	P
14	rain	high	true	N

Fig. 5. Dados de treino.

Os dados possuem três atributos e pretende-se estabelecer uma relação entre os valores para cada atributo e o resultado ou classe que resulta da combinação desses valores.

Para isso, é necessário separar os dados pelos atributos em subconjuntos:

Outlook			Wind		Humidity	
Sunny	Overcast	Rain	False	True	High	Normal
2 P	4 P	3 P	6 P	3 P	6 P	6 P
3 N	0 N	2 N	2 N	2 N	3 N	1 N

Efectuada a separação, é importante verificar a “pureza” dos subconjuntos. Um subconjunto é puro quando não há incerteza quanto à classificação dos dados após a separação correspondente.

- Subconjunto puro: ($aP/0N$) ou ($0P/aN$) onde ‘a’ representa o número total de dados para o atributo a que se refere o subconjunto. Por exemplo, na figura acima, *Overcast* é um subconjunto puro ($4P/0N$). O resultado de um dado apresentar *Overcast* como valor de *Outlook* é sempre ‘P’ (não existe incerteza).

A quantidade de incerteza num atributo é medida através da entropia ou quantidade de informação (em bits):

$$H(x) = - \sum_x p_P(x) \log_2 p_N(x)$$

Este conceito de entropia vai ser necessário para determinar a ordem pela qual os nodos aparecem sendo que, para gerar a árvore de menor tamanho, pretende-se que os

nodos estejam ordenados pelo ganho de informação que apresentam com a separação dos dados.

Ou seja:

$$\text{ganho de informação} = H_{\text{antes separação}} - H_{\text{após separação}}$$

A título de exemplo, consideremos o atributo *Outlook*:

- Para o valor de *Sunny*: $H = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971 \text{ bits}$
- Para o valor de *Overcast*: $H = -1\log_2 1 - 0\log_2 0 = 0 \text{ bits}$
- Para o valor de *Rainy*: $H = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971 \text{ bits}$

➤ Informação total antes da separação:
 $-(\frac{5}{14}\log_2\frac{5}{14} + \frac{9}{14}\log_2\frac{9}{14}) = 0.940 \text{ bits}$

➤ Informação total após separação:
 $\frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971 = 0.693 \text{ bits}$

$$\text{ganho de informação (Outlook)} = 0.940 - 0.693 = 0.247$$

Fazendo o mesmo para os restantes atributos:

$$\text{ganho de informação (Humidity)} = 0.152$$

$$\text{ganho de informação (Windy)} = 0.048$$

Chegamos então à conclusão que o nodo a partir do qual a árvore deve “crescer” será o correspondente ao atributo *Outlook*, pelo que os subconjuntos não puros do mesmo deverão ser os nodos a partir dos quais crescem os restantes atributos.

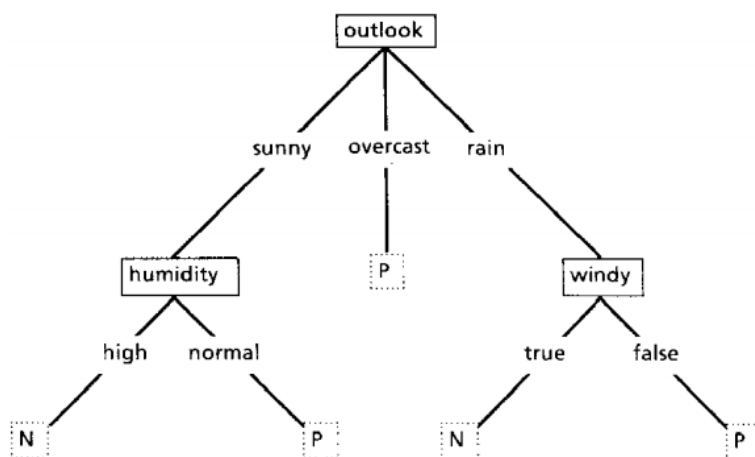


Fig. 6. Árvore de decisão resultante.

As árvores de decisão não trabalham apenas com atributos com valores categóricos (no último exemplo *Sunny, Overcast, Rain*), mas também com valores contínuos, como intervalos de valores. Neste último caso, devem ser impostos limites (*thresholds*) aos valores contínuos de modo a gerar subconjuntos de valores acima e abaixo desses limites.

Vantagens e Desvantagens das Árvores de Decisão

Vantagens:

- São fáceis de interpretar por seres humanos (conseguimos entender decisões);
- Não requerem muita preparação ao nível dos dados;
- Conseguem lidar com maior parte dos atributos irrelevantes, visto que são ignorados;
- Conseguem lidar com dados categóricos e numéricos;
- É possível validar os modelos através de testes estatísticos;
- São compactas: o número de nodos é menor ou igual ao número de atributos;
- Os tempos de teste são rápidos: o algoritmo é $O(\text{profundidade da árvore})$.

Desvantagens:

Só conseguem dividir os dados na horizontal ou vertical, visto esta divisão é feita um atributo de cada vez. Isto significa que num conjunto de dados onde uma divisão na diagonal seja mais apropriada, as árvores de decisão necessitam de mais iterações para obter um conjunto puro do que outros algoritmos

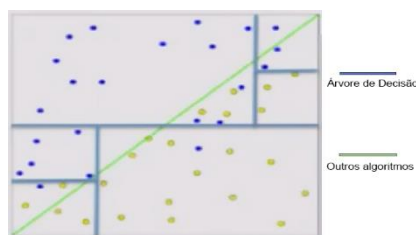


Fig. 7. Representação visual da divisão em subconjuntos.

Ganhos de informação tendem a favorecer atributos com muitos valores. Por exemplo, um atributo “dia” com os valores dia1, dia2, ..., dia15 provavelmente teria todos os seus subconjuntos como sendo unitários e consequentemente puros. Pelo ganho de informação que o atributo representa, poderia à primeira vista ser um atributo bastante útil, mas na verdade não funciona em novos dados (por exemplo, um dado de teste com dia16).

A forma de solucionar este problema seria manualmente retirar os atributos que obviamente não contribuem para a aprendizagem do sistema ou, usando um rácio de ganho, penalizar os atributos com um grande número de valores

Overfitting: como o algoritmo irá continuar a dividir os dados até a árvore ser perfeita, corre-se o risco de esta se tornar demasiado específica aos dados de treino e não generalizar bem para os casos de teste.

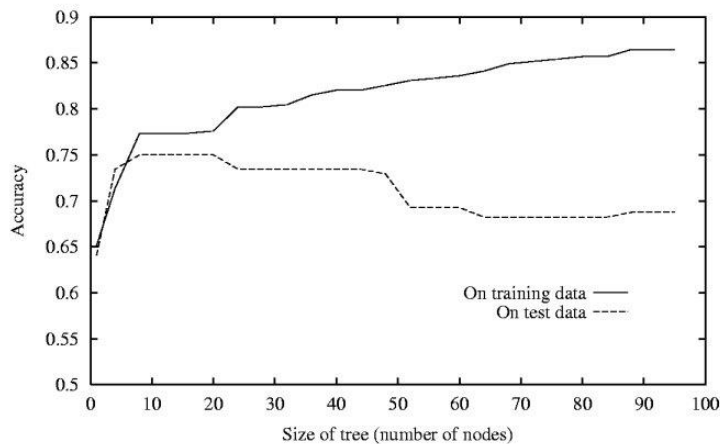


Fig. 8. Relação entre a evolução da precisão nos dados de treino e nos dados de teste.

Existe uma solução denominada pruning que faz com que a árvore não cresça demasiado e acabe com subconjuntos de um único elemento nas folhas, no entanto nem todas as implementações de árvores de decisão possuem este mecanismo (por exemplo, a livreria de Python scikit)

O algoritmo é greedy, pelo que o resultado poderá não ser o melhor.

3.2 Capacidade de aprendizagem

As árvores de decisão exibem aprendizagem na medida em que o algoritmo escolhido (por exemplo, o ID3) irá processar os dados de treino e tentará perceber como as diferentes combinações de valores para cada atributo dos dados influenciam o resultado ou classe à qual pertencem.

No caso das árvores de decisão em particular, a noção de “conteúdo aprendido” é ainda mais fácil de visualizar através do conceito de informação ganha descrito acima. Cada atributo possui um ganho de informação associado pelo que, caso um atributo apresente um ganho de zero, é possível afirmar que este não contribui para a aprendizagem e vice-versa.

Dados	Atributo A	Atributo B	Resultado/Classe
1.	Y	X	P
2.	Z	X	P
3.	Y	X	P
4.	Z	X	P
5.	Y	V	N
6.	Z	V	N
7.	Y	V	N
8.	Z	V	N

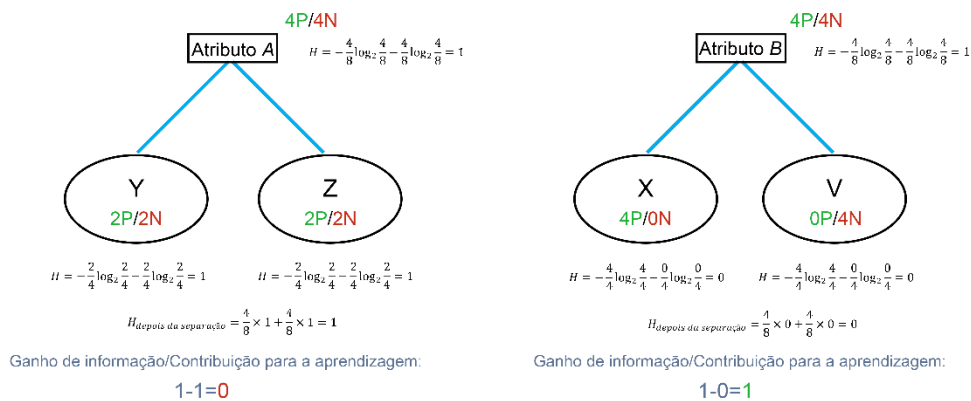


Fig. 9. Contribuição para a aprendizagem de dois atributos.

Um atributo “inútil” possui o mesmo nível de entropia (ou certeza) antes e depois da separação, ou seja, ao dividir um conjunto de dados pelos diversos valores que compõem o atributo, não é possível chegar a uma conclusão quanto à influência do mesmo no resultado.

Um atributo confere o máximo de aprendizagem quando os seus subconjuntos são todos puros.

3.3 Ferramentas de Desenvolvimento

As árvores de decisão podem ser implementadas em qualquer linguagem de programação através de algoritmos, como o ID3, e algumas linguagens possuem já

livrarias com árvores de decisão implementadas. Existem, no entanto, soluções já implementadas no âmbito de várias áreas para além de *Machine Learning*.

- scikit (livraria de Python): <http://scikit-learn.org/stable/modules/tree.html>
- SMILES: <http://users.dsic.upv.es/~flip/smiles/>
- BigML oferece árvores de decisão como serviço: <https://bigml.com/>
- SPM mineração de dados e análise preditiva <https://www.salford-systems.com/>
- DTREG modelação preditiva <https://www.dtreg.com/>

Estas ferramentas recebem um ficheiro com os dados formatados e uma determinada decisão da qual a se quer saber a viabilidade e apresentam ao utilizador a previsão com base nos casos de treino, muitas vezes com o diagrama da árvore para ser possível verificar o que levou ao resultado obtido.

3.4 Soluções existentes no mercado

Selection of myocardial electrogram features for use by implantable devices

Autores:

W.J. Gibb (Cardiovascular Res. Inst., California Univ., San Francisco, CA, USA),
D.M. Auslander, J.C. Griffin

Descrição:

Uso de árvores de decisão para identificar *features* a implementar em dispositivos de implante cardíaco.

Decision trees for automated identification of cosmic-ray hits in Hubble Space Telescope images. Publications of the Astronomical Society of the Pacific, 107:1--10, March 1995.

Autores:

Steven Salzberg, Rupali Chandar, Holland Ford, Sreerama Murthy e Rick White

Descrição: Uso de árvores de decisão para filtrar ruído do telescópio espacial Hubble onde é necessário, a partir de uma imagem, determinar quais os pixéis correspondentes à informação útil. O método produz classificadores com mais de 95% de precisão a partir de uma única imagem.

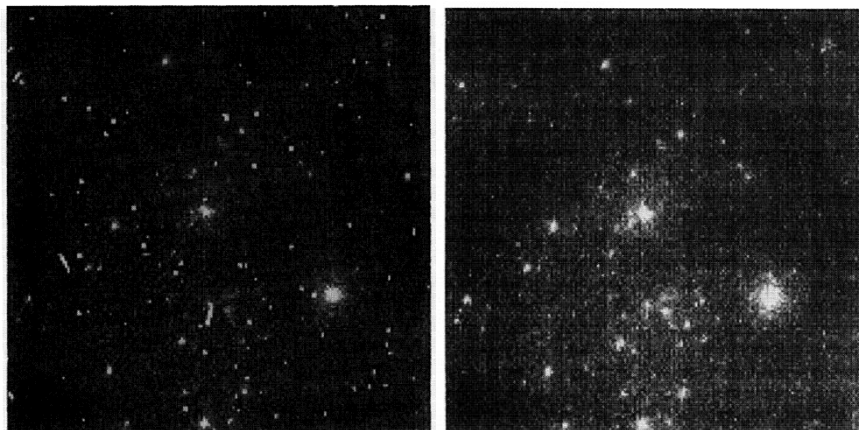


Fig. 10. Imagem original à esquerda e imagem com filtro aplicado à direita

Applying machine learning to agricultural data. Computers and Electronics in Agriculture, 12(4):275--293, June 1995.

Autores:

R.j. Mcqueen, s. R. Garner, C.G. Nevill-Manning, e I.H. Witten.

Descrição:

Aplicação de árvores de decisão na agricultura e criação de gado, para substituir a necessidade de testes empíricos que são processos extremamente lentos e sujeitos a erros. No artigo sublinha-se o potencial deste método de aprendizagem, mas admite-se que os algoritmos ainda têm de evoluir para acomodarem estruturas de dados de grande dimensão multidimensionais e não apenas pequenas estruturas bidimensionais.

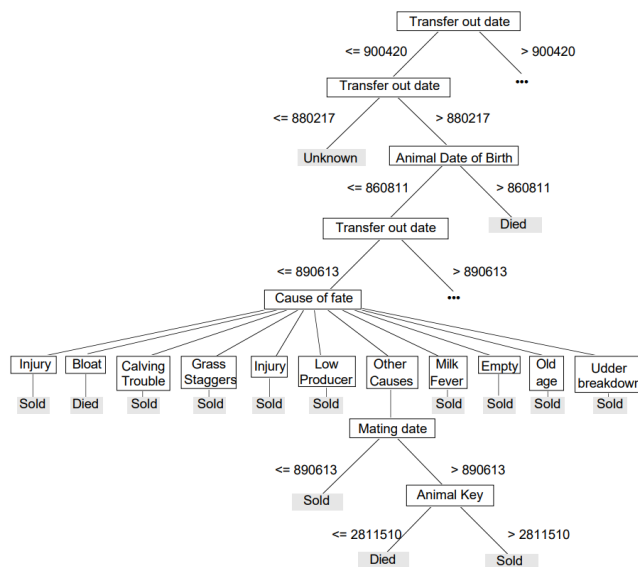


Fig. 11. Árvore de decisão gerada a partir de dados relativos a estados de animais de gado.

Three dimensional object recognition using similar triangles and decision trees.
Pattern Recognition, 26(5):727, May 1993.

Autor:

Lilly Spirkovska

Descrição:

O sistema TRIDEC é capaz de distinguir conjuntos de objetos num espaço tridimensional mesmo com mudanças nas posições, escalas e rotações destes objetos. A classificação dos objetos é conseguida através da construção de uma árvore de decisão a partir de informação proveniente de um número limitado de exemplos com translações, rotações e escalas aplicadas.

4 Raciocínio Baseado em Casos

4.1 Descrição

Um Sistema Baseado em Casos é um tipo de sistema que utiliza Raciocínio Baseado em Casos para resolver problemas, ou seja, o seu funcionamento recorre ao processo de resolver novos problemas baseando-se em soluções de problemas similares anteriores, podendo ainda haver uma adaptação das soluções ao caso específico.

O sistema parte de um conjunto de casos que conhece, e ainda extrai conhecimento a partir de novos casos, adicionando-os à base de casos, depois de encontrar uma solução adequada para o problema.

Segundo os investigadores na área da inteligência artificial, entre outras, o raciocínio baseado em casos define-se como um “método de desenvolvimento de um sistema especialista baseado em conhecimento que desempenha exemplos de experiências passadas num domínio específico que pode ser aplicado em novos problemas” (DUTA, 1993), ou, mais extensamente, como “um raciocínio baseado na memória de experiências prévias. Um raciocinador utilizando experiências antigas (casos) poderá utilizar esses casos para sugerir soluções para problemas, para apontar potenciais novos problemas com uma solução a ser computada, para interpretar uma nova situação e fazer previsões acerca do que poderá acontecer, ou criar argumentos que justifiquem algumas conclusões” (Kolodner, 1993).

Num estudo sobre diagnóstico nutricional, a autora Maria Alice Lagos Thé (2001), refere que “uma vantagem fornecida por sistemas inteligentes que utilizam raciocínio baseado em casos é a habilidade de estocar conhecimentos de vários especialistas, tornando a tarefa do investigador representar mais a realidade e ser propenso a menores erros”. Nesta investigação, a partir da validação do modelo Fuzzy e da metodologia de RBC, a autora conseguiu a implementação da tarefa nutricional próxima dos 100% com a entrada de um novo caso na base de casos. Mencionando Watson (1997), a autora refere que “quando empregado o paradigma de relembrar o prévio episódio mais similar, os humanos são sujeitos a colocar emoções que podem evitar recuperação relevante das soluções, a passo que um sistema computacional, logo que realiza requerimentos teóricos, garante que as similaridades das experiências sejam recuperadas”.

Este tipo de sistema tem grande aplicabilidade, porque permite a obtenção de boas soluções tendo-se pouco domínio do problema, podendo propor soluções em casos em que é desconhecido o estado completo do problema. Chega-se à solução apenas a partir do conhecimento adquirido de outros casos, espera-se assim uma melhoria no desempenho quando o número de casos da base de casos aumenta.

4.2 Capacidade de Aprendizagem

Para se explicar a capacidade de aprendizagem deste tipo de sistemas, apresenta-se uma figura ilustrativa do funcionamento de um sistema deste tipo:

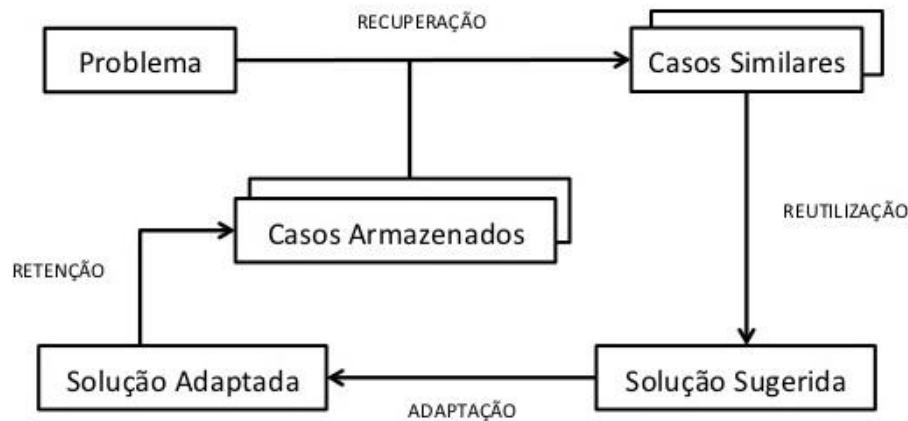


Fig. 12. Funcionamento básico de um Sistema Baseado em Casos.

De seguida, serão explicadas as etapas ilustradas na figura, descrevendo estas o funcionamento ao qual recorre um Sistema Baseado em Casos.

Recuperação

Dada uma descrição de um problema, ou simplesmente as características de um problema, esta etapa é responsável por encontrar o caso ou casos mais similares à situação atual, utilizando os índices da Base de Casos, um identificador único. Esta etapa baseia-se nos índices e na organização de memória para dirigir a procura dos casos potencialmente úteis, ou do caso mais similar.

Os algoritmos mais simples baseiam-se no exame exaustivo das características dos casos e recuperam o caso com o maior número de características idênticas. Outros buscam o melhor caso (Kolodner, 1989), utilizando-se de heurísticas para reduzir e dirigir a procura. Entre esses pode-se citar: busca serial (Simoudis et al, 1993), busca hierárquica (Maher e Zhang 1991) e busca por simulação paralela (Domeshek 1993).

Entre os métodos mais conhecidos de recuperação de casos estão o algoritmos de vizinhança, de indução, indução guiada por conhecimento e recuperação de padrões, que são utilizados sozinhos ou combinados.

Reutilização

A partir do caso recuperado é feita a reutilização da solução associada àquele caso. A solução do caso recuperado pode ser transferida para o novo problema diretamente como sendo a sua solução, ou pode ser efetuada uma adaptação/revisão da solução.

Adaptação/Revisão

Quando uma situação é fornecida, o algoritmo de recuperação traz o caso mais similar que encontra. Normalmente, o caso selecionado não é perfeitamente igual à descrição do problema do utilizador. Caso existam diferenças entre o problema do utilizador e o caso devolvido na etapa recuperação estas devem ser tidas em conta. O

processo de adaptação procura diferenças salientes entre as duas descrições e aplica regras de forma a ajustar a solução.

Embora as regras aparentem ser bastante específicas, construí-las para adaptação é muito mais simples do que desenvolver um sistema puramente baseado em regras, desde que os casos armazenados tenham uma razoável cobertura sobre o domínio do problema. A adaptação é feita a partir de um conjunto menor de regras, resultando em uma maior eficiência e eficácia na obtenção da solução ideal.

Retensão/Aprendizagem

Processo final, posterior ao arranjo da solução para o problema em questão. O objetivo é armazenar o novo caso e a sua respetiva solução para futuras recuperações. O sistema irá decidir que informações armazenar e de que forma.

4.3 Ferramentas de Desenvolvimento

Um RBC pode ser desenvolvido em qualquer linguagem de programação: existem boas implementações em C#, Java ou ASP.NET. No entanto, para um desenvolvimento user-friendly existem algumas ferramentas:

- MyCbr (<http://www.mycbr-project.net/>):
- COLIBRI (<http://gaia.fdi.ucm.es/research/colibri>)
- CBR Shell (<http://www.research-innovation.ed.ac.uk/Opportunities/Case-Based-Reasoning>)
- Cognitive Systems Inc. - ReMind (no longer trading)(<https://ai-cbr.cs.auckland.ac.nz/tools/remind.html>);
- Esteem Software Inc. – ESTEEM (<https://ai-cbr.cs.auckland.ac.nz/tools/esteem.html>);
- Inductive Solutions Inc. – CasePower (<https://ai-cbr.cs.auckland.ac.nz/tools/inductive.html>);
- IET-Intelligent Electronics – TechMate (<https://ai-cbr.cs.auckland.ac.nz/tools/inference.html>);
- Intellix – KnowMan (<https://ai-cbr.cs.auckland.ac.nz/tools/knowman.html>);

Estas ferramentas permite a definição amigável de funções para a similaridade entre variáveis, importação de dados, entre outras funções, sem qualquer programação.

4.4 Soluções existentes no mercado

Nesta secção serão mencionadas soluções de mercado com sistemas CBR e será dada uma explicação acerca do que foi feito em cada trabalho mencionado.

Forecasting of manufacturing cost in mobile phone products by case-based reasoning and artificial neural network models

Autores:

1. Department of Information Management Yuan Ze University Taoyuan Taiwan, R.O.C
2. Department of Naval Architecture National Kaohsiung Marine - University Kaohsiung Taiwan

Descrição:

Os fabricantes de telemóveis em Taiwan fizeram grandes esforços em propor as cotações racionais para as empresas de telemóveis internacionais com a ambição de ganhar a outros fabricantes. No entanto, há várias incertezas e problemas a serem resolvidos na estimativa dos custos de fabricação para os fabricantes de telefones. Não existe um modelo existente que possa ser aplicado diretamente na previsão dos custos de produção. Este trabalho faz a primeira tentativa de desenvolver um sistema híbrido de Raciocínio Baseado em Casos e Redes Neurais Artificiais como um modelo de previsão de custo unitário de produto para a Mobile Phone Company. De acordo com a fórmula de custo do telemóvel e as opiniões dos peritos, um conjunto de fatores qualitativos e quantitativos são analisados e determinados. Os fatores qualitativos são aplicados na Sistema Baseado em Casos para recuperar um caso semelhante a partir da base de casos para um novo produto de telemóvel e as redes neurais são usadas para encontrar a relação entre os fatores quantitativos e a previsão de custo unitário de produto prevista. Finalmente, experiências intensivas são conduzidas para testar a eficácia de seis diferentes modelos de previsão. O modelo proposto neste trabalho é comparado com os outros cinco modelos e o valor de MAPE do modelo proposto é o menor. Esta pesquisa fornece um novo modelo de previsão com alta precisão para empresas de fabricação de telemóveis.

CookIIS Mobile: A Case-Based Reasoning Recipe Customizer for Android Phones

Autores:

Kerstin Bach, Klaus-Dieter Althoff, Julian Satzky, and Julian Kroehl (Competence Center Case-Based Reasoning, German Research Center for Artificial Intelligence (DFKI) GmbH) - University of Hildesheim Institute of Computer Science - Intelligent Information Systems Lab

Descrição: CookIIS é uma aplicação Android que permite a gestão de receitas, permitindo ao utilizador inserir as receitas que faz diariamente. Após um período inicial de recolha de casos, o software consegue sugerir receitas ao utilizador, tendo em conta o seu gosto pessoal e as receitas que o mesmo costuma fazer.

Case-Based Reasoning for Selecting Study Program in Senior High School

Autores:

Sri Mulyana, Sri Hartati, Retantyo Wardoyo, Edi Winarko (Department of Computer Sciences and Electronics) - Gadjah Mada University

Descrição:

Esta aplicação é utilizada para auxiliar os alunos na seleção do programa de estudo. Os casos utilizados no estudo incluem resultados do teste de inteligência, interesse do aluno e notas de vários assuntos. Os casos com maior grau de similaridade são recomendados como soluções.

Case-based reasoning combined with statistics for diagnostics and prognosis

Autores:

T. Olsson and P. Funk

Descrição:

Hoje em dia, muitas abordagens utilizadas para diagnóstico são baseadas em um modelo preciso. Isso exclui o diagnóstico de muitos tipos complexos de máquinas que não podem ser modeladas e simuladas facilmente ou sem grande esforço. O objetivo deste software foi mostrar que, ao incluir a experiência humana, é possível diagnosticar máquinas complexas quando não há modelos limitados ou simulações disponíveis. Isso também permite o diagnóstico em uma aplicação dinâmica onde as condições mudam e novos casos são frequentemente adicionados. Assim, cada novo caso resolvido aumenta o poder de diagnóstico do sistema.

5. Sistema inteligente para determinar qualidade da informação

Quando se utiliza um modelo de aprendizagem, como os 3 abordados neste trabalho, é essencial determinar a qualidade da informação de forma a validar os casos que utilizamos para treinar um modelo e o caso ao qual vamos aplicar o modelo, uma vez que a informação pode ser desconhecida, contraditória, ambígua, imperfeita, parcial e não ser quantitativa.

A importância de a informação ser qualitativa, quando necessário, é que podemos saber o intervalo onde o atributo cai (com determinado grau de confiança). Nos artigos “A Soft Computing Approach to Quality Evaluation of General Chemistry Learning in Higher Education” e “Dyscalculia: A Behavioural Vision” é apresentada a prova teórica nos termos da lógica programática para o cálculo da qualidade da informação.

A qualidade da informação (QoI) denota um intervalo em que 0 significa que a informação é desconhecida e 1 significa que é totalmente conhecida. Para a aproximação da qualidade da informação utiliza-se um intervalo de confiança que representa a probabilidade de o valor real cair no intervalo. De notar que este sistema permite avaliar todas as possibilidades quando há mais que uma (intervalos qualitativos), de forma a ser o mais preciso possível. Além disso permite-nos ser flexíveis na forma de representação do conhecimento.

Assim, antes de qualquer sistema de aprendizagem ser aplicado em dados, os dados devem ser submetidos a este sistema inteligente para determinar a qualidade da informação, permitindo tomar decisões em relação aos dados.

6. “AI is no more riding a one-trick pony”

Em 1986, Rumelhart, Hinton e Williams mostraram experimentalmente que o método Backpropagation podia gerar representações internas úteis de dados recebidos em camadas ocultas de redes neurais. [20]. Nas décadas seguintes até hoje, a utilização de RNA's com Backpropagation mostrou-se uma excelente forma para resolver inúmeros problemas, desde ao reconhecimento de padrões, reconhecimento facial ou medicina, criando-se algoritmos robustos. Mais recentemente, a utilização de RNA's profundas tem vindo a revolucionar várias áreas, como por exemplo a imagem médica: Este tipo de algoritmos tornaram-se capazes de, por exemplo, conseguir identificar tumores muitas vezes melhor que o Médico. Ainda que, este tipo de tecnologias, por questões éticas, não são usadas para substituir médicos mas sim para os ajudar.

Todos estes factos levaram a que o método Backpropagation não fosse questionado, e se tornasse um padrão a ser utilizado em RNA's, levando a AI a convergir para soluções com o mesmo modelo: RNA com backpropagation.

O artigo “A Deep Learning approach to Case Based Reasoning to the Evaluation and Diagnosis of Cervical Carnicoma”[7] propõe uma alternativa inovadora que apresenta vantagens em relação a RNA com backpropagation. As RNA acabam por afunilar o que é a IA e além disso, obrigam à utilização de uma função objetivo que não conhecemos, assim como à utilização de informação quantitativa, completa, não contraditória e necessidade de todo o conhecimento e que este conhecimento seja em forma quantitativa (ou traduzido para tal).

O método proposto por Neves. J. et al. [7] permite a solução destes problemas, conciliando deep learning com cased-based reasoning e permitindo que a informação possa ser desconhecida, incompleta, perdida, contraditória e qualitativa, medindo a qualidade da informação e o grau de confiança, testando-se todas as alternativas possíveis. Assim, este problema das RNA's com backpropation fica resolvido.

Enquanto uma RNA com backpropation, como explicado nas secções anteriores, não oferece uma função objetivo depois do treino, mas sim os pesos corretos para as funções ativação de cada neurónio, com este método passa a conhecer-se a função objetivo.

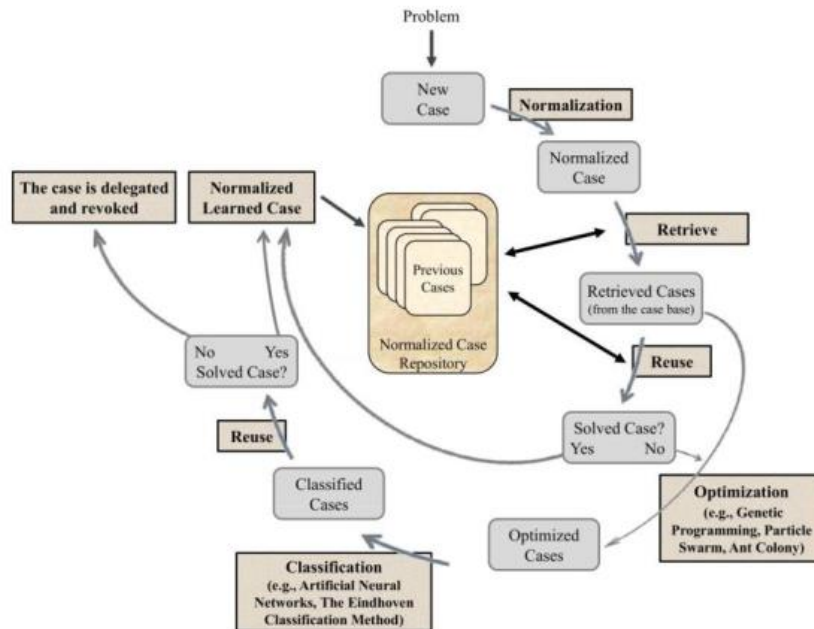


Fig. 13. Modelo proposto por Neves J. et al..

No modelo proposto, existem importantes variações no modelo original do Raciocínio Baseado em Casos (Fig. 5.), tornando o modelo mais flexível. A flexibilidade na forma de representação do conhecimento deste modelo é um ponto importante uma vez que permite que um caso não seja descartado por haver um conjunto de restrições como nas RNA's (informação necessariamente completa, não contraditória e quantitativa).

As diferenças para com o CBR tradicional começam na recuperação de um caso da base de casos onde vários casos são recuperados (os que têm maior similaridade com o caso proposto). De seguida, é possível obter a função objetivo para o problema em análise nos termos da extensão do predicado, como por exemplo:

$$diag_{CSCC_{NEW\ CASE}} \left(((0.49, 0.49)(1, 1)), \dots, ((0, 0.5)(1, 0.87)) \right) :: 1 :: 0.84$$

Fig. 14. Exemplo do predicado, proposto por Neves J. et al..

E comparar o novo caso com cada caso recuperado, usando a função de similaridade: média de cada módulo da diferença aritmética entre os argumentos de cada caso selecionado e os argumentos do novo caso.

Por fim, para cada caso é calculada a similaridade com o caso proposto a partir do cálculo de não similaridade e posterior diferença de $1 - DoC$ (cálculo de não similaridade).

$$dis_{NEW CASE \rightarrow 1}^{DoC} = \frac{|1 - 1| + \dots + |0.87 - 1|}{15} = 0.022$$

Fig. 15. Exemplo do cálculo de não similaridade, proposto por Neves J. et al..

Assim, conclui-se que nem AI nem DL são só RNA's com backpropagation.

7. Comparando os três modelos de aprendizagem

Todos estes métodos apresentam vantagens específicas aos seus modos de funcionamento e, para ser possível compará-los, é necessário estabelecer critérios de comparação. Os três principais critérios a considerar são a precisão dos resultados, performance a nível dos testes (tempo de execução) e interpretabilidade, ou seja, a facilidade com que é possível demonstrar os passos que levaram ao resultado.

Em termos de precisão, as redes neuronais foram muitas vezes demonstradas como tendo melhores resultados do que as árvores de decisão, como no artigo “*Trees vs Neurons*”[32], onde se compara uma versão de melhor desempenho das árvores (*Random Forests*) com RNAs (facto também observado na comparação da Fig. 9). Em “*A Comparison of Software Effort Estimation Techniques: Using Function Points with Neural Networks, Case-Based Reasoning and Regression Models*”[33] constata-se que as diferenças de precisão entre redes neuronais e case-based reasoning é estatisticamente insignificante em tarefas envolvendo estimativas de esforço para desenvolvimento de software.

Como já foi referido neste documento, uma das vantagens das árvores de decisão é descartar atributos que não considera úteis, enquanto os restantes métodos, a não ser que sejam feitas alterações de pré-processamento, irão usar todos os atributos. Esta característica, aliada aos tempos rápidos de procura numa árvore, significa que à partida devem possuir melhor performance em testes do que os restantes métodos. Testes de classificação efectuados tendem a suportar esta teoria [34 e 35].

Model	DT Model	NN Model
Model Topology	Tree structure	MLF three-layer structure with variable number of nodes on the middle layer
Estimation Method	Recursive partitioning	Backpropagation
Estimation Efficiency	Fast (approximately 1-2 sec)	Extremely slow (approximately 1-1.5 hr)
Prediction Performance	76.8%	78.2%
Interpretability	Explicit decision trees/ <i>if-then</i> rules	Implicit “black box”
Software Tool	See5/C5.0	MATLAB R11

Fig. 16. Comparação entre árvores de decisão (DT) e redes neuronais (NN) [35].

As comparações dos parâmetros anteriormente referidos estão sujeitas a variações e não se devem efectuar generalizações a partir de casos pontuais como os referenciados. Cada método de aprendizagem possuirá um conjunto de situações nas quais terá um melhor comportamento do que os restantes e, para além disso, existem diversos algoritmos para cada método, tornando comparações generalizadas ainda menos relevantes. Estas devem, portanto, ser tidas como estatísticas com base em diversos testes já efectuados.

O último critério de interpretabilidade é, no entanto, de fácil e concreta comparação. Perante este critério, as redes neuronais estão em clara desvantagem em relação aos restantes métodos, sendo conhecidas como “*Black Boxes*” (Fig.9), ou seja, os percursos que tomam para chegar ao resultado são muitas vezes ocultados e impossíveis de analisar, tornando-as pouco recomendadas para áreas onde uma explicação para o resultado é importante.

Combinando os métodos

Em “*Using Decision Trees to Improve Case-Based Learning*” [35] compara-se a performance em processamento de linguagens naturais de CBR com árvores de decisão e com um híbrido dos dois, sendo que o híbrido supera ambos os métodos.

Genetic programming oferece uma forma genérica de fundir classificadores como redes neuronais e árvores de decisão, de modo a gerar um método superior e já é usado em investigações [36].

8. Conclusões

Neste documento foram abordados três métodos de aprendizagem supervisionada. Cada método possui um modo de funcionamento único, traduzindo-se em diferentes formas de abordar problemas de classificação e resultados distintos.

Estas diferenças devem contribuir para a diversidade no ramo de *Machine Learning* e não para efectivamente seleccionar o “melhor” método de aprendizagem, conduzindo a um afunilamento na Inteligência Artificial. Até porque, como já demonstrado, a criação de “híbridos” tem levado a resultados que superam os métodos individuais que os compõem.

Agradecimentos. Para este trabalho, foram de grande utilidade as aulas lecionadas pelos professores César Analide e José Neves.

Referências

1. Richard S. Sutton and Andrew G. Barto.: Reinforcement Learning: An Introduction, The MIT Press, 2nd edition. (2012)
2. Aamodt A., Plaza E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, in AI Communications, Vol. 7, Nº 1, pages 39-59. (1994)

3. Haykin, S.: Neural Networks – A Comprehensive Foundation, Prentice-Hall, New Jersey, 2nd Edition. (1999)
4. David Goldberg.: Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley. (1989)
5. Quinlan, J. R.: Induction of Decision Trees, Machine Learning 1: 81-106, Kluwer Academic Publishers. (1986)
6. Nello Cristianini, John Shawe-Taylor.: An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press. (2000)
7. Neves J., Vicente H., Ferraz F., Leite A., Rodrigues A., Cruz M., Machado J., Neves. J.: A Deep Learning approach to Case Based Reasoning to the Evaluation and Diagnosis of Cervical Carnicoma. (2017)
8. Figueiredo M., Neves J., Vicente H.: A Soft Computing Approach to Quality Evaluation of General Chemistry Learning in Higher Education. (2011)
9. Neves J., Analide C.: Brief Introduction to AI, DL and KRR. Aprendizagem e Extração de Conhecimento. (2017)
10. Nahar K.: Artificial Neural Network. COMPUSOFT, An international journal of advanced computer technology, Volume-I, Issue-II. (2012)
11. Neves J., Cortez P.: Redes Neurais Artificiais. Departamento de Informática da Universidade do Minho. (2000)
12. Redes Neurais Artificiais, <http://conteudo.icmc.usp.br/pessoas/andre/research/neural/>. (Disponível a 17 de Outubro de 2017)
13. Kasparov G.: The computer hasn't prove anything yet. New York. (1997)
14. Aloraini.: Different machine learning algorithms for breast cancer diagnosis. International Journal of Artificial Intelligence. (2012)
15. Pires E.: Redes Neurais Artificiais. Departamento de Engenharias, Universidade de Trás-os-Montes e Alto Douro, Portugal. (2016)
16. Luiz C.: Redes Neurais Artificiais, <http://www.inf.ufg.br/~cedric/ia/grad/AI-Aula12-Redes%20Neurais%20Artificiais.pdf>. (Disponível a 18 de Outubro de 2017)
17. Gomes V., Fernandes A, Lopes P., Pereira L., Faia A.Pinto P.: Characterization of neural network generalization in the determination of pH and anthocyanin content of wine grape in new vintages and varieties. (2017)
18. Roy S, Mallik A., Gulati R., Obaidat M, Krishna P.: A Deep Learning Based Artificial Neural Network Approach for Intrusion Detection. (2017)
19. Dhungel N., Carneiro G., Bradley A.: A deep learning approach for the analysis of masses in mammograms with minimal user intervention. (2017)
20. Rumelhart, David E., Hinton, Geoffrey E., Williams, Ronald J.: "Learning representations by back-propagating errors". Nature. 323 (6088): 533–536. doi:10.1038/323533a0. (8 October 1986).
21. Leake, D. B.: CBR in Context: The Present and Future (AAAI Press/MIT Press). Consultado a 19 de Outubro em <http://www.cs.indiana.edu/pub/leake/p-96-01.pdf>. (1996)
22. Castoldi A., Santos M.O., Raciocínio Baseado em Casos. Consultado em 19 de Outubro em <http://www.inf.ufsc.br/~barreto/trabaluno/ia2002augmarc.pdf>. (2002)
23. Kolodner J.L.: Case-Based Learning. Consultado a 20 de Outubro em <http://www.google.pt/books?id=CC2-3VA8UfwC&printsec=frontcover&hl=pt-PT#v=onepage&q&f=false>. (1993)
24. Aamodt, A., Plaza. E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. Consultado a 20 de Outubro em: http://www.idi.ntnu.no/emner/tdt4173/papers/Aamodt_1994_Case.pdf. 1994)
25. Abel, M.: Um estudo sobre Raciocínio Baseado em Casos. Consultado a 20 de Outubro em: <http://www.inf.ufgrs.br/bdi/wp-content/uploads/CBR-TI60.pdf>. (1996)

26. Castoldi, A.: & Santos, M.O. Raciocínio Baseado em Casos. Consultado a 20 de Outubro em: <http://www.inf.ufsc.br/~barreto/trabaluno/ia20022augmarc.pdf>. (2002)
27. Olsson, T., Funk P.: Case-based reasoning combined with statistics for diagnostics and prognosis. Consultado a 20 de Outubro em <http://iopscience.iop.org/article/10.1088/1742-6596/364/1/012061/pdf>.
28. Mulyana S., Hartati S., Wardoyo R., Winarko E. Case-Based Reasoning for Selecting Study Program in Senior High School.: Consultado a 21 de Outubro em http://thesai.org/Downloads/Volume6No4/Paper_18-Case-Based Reasoning for Selecting Study Program in Senior High School.pdf.
29. Kerstin B., Klaus-Dieter A., Julian S., Julian K.: CookIIS Mobile: A Case-Based Reasoning Recipe Customizer for Android Phones. Consultado a 21 de Outubro em https://www.google.pt/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwjf7Jvgz9PQAUG1RoKHeZOBACQFggfMAA&url=https%3A%2F%2Fwww.dfki.de%2Fweb%2Fforschung%2Fkm%2Fpublikationen%2FrenameFileForDownload%3Ffilename%3D2012-cookiis-ukcbr.pdf%26file_id%3Duploads_1867&usg=AFQjCNE3k6sohKIDy1Q49mavr84OBNcpSA&sig2=GBclMbTkQw8VTb7VPaTk_A.
30. Pei-Chann, Chang, Jyun-Jie Lin, Wei-Yuan Dzan.: Forecasting of manufacturing cost in mobile phone products by case-based reasoning and artificial neural network models. Consultado a 21 de Outubro em <http://link.springer.com/article/10.1007/s10845-010-0390-7>.
31. Ferraz F., Neves J., Alves V., Henrique V.: Dyscalculia: A Behavioural Vision. (2017)
32. Muhammad Waseem Ahmad, Monjur Mourshed, Yacine Rezgui: Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption. Consultado a 24 de Outubro em <http://www.sciencedirect.com/science/article/pii/S0378778816313937>
33. G.R. Finnie, G.E. Wittig, J-M. Desharnais: A Comparison of Software Effort Estimation Techniques: Using Function Points with Neural Networks, Case-Based Reasoning and Regression Models. Consultado a 24 de Outubro em <http://www.sciencedirect.com/science/article/pii/S0164121297000551>
34. Daniela XHEMALI, Christopher J. HINDE, Roger G. STONE: Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages. Consultado a 24 de Outubro em <http://cogprints.org/6708/1/4-1-16-23.pdf>.
35. Chi Xie, Jinyang Lu, Emily Parkany: Work Travel Mode Choice Modeling Using Data Mining: Decision Trees And Neural Networks. Consultado a 24 de Outubro em <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.501.1028&rep=rep1&type=pdf>.
36. William B. Langdon, S. J. Barrett, F. Buxton: Combining Decision Trees and Neural Networks for Drug Discovery. Consultado a 24 de Outubro em <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.307.618&rep=rep1&type=pdf>.